acctbuf	acct.c	acct	acctbuf.ac_comm[i] = u.u_comm[i];
			acctbuf.ac_flag = u.u_acflag;
			acctbuf.ac_uid = u.u_ruid;
			acctbuf.ac_date = u.u_start;
			acctbuf.ac_etime = time - u.u_start;
			acctbuf.ac_utime = u.u_utime;
			acctbuf.ac_stime = u.u_stime;
			acctbuf.ac_dread = u.u_dread;
			acctbuf.ac_dwrit = u.u_dwrit;
			u.u_base = (caddr_t)&acctbuf;
			u.u_count = sizeof(acctbuf);

	systm.h	_	} acctbuf;
acctp	acct.c	acct	if ((ip=acctp)==NULL)
		sysacct	if (acctp) {
			plock(acctp);
			iput(acctp);
			acctp = NULL;
			if (acctp) {
			acctp = ip;
	systm.h	_	int *acctp; /* inode of accounting file */
badent	sysent.c	_	int badent[] {
	sysent.tu.c		int badent[] = {
	sysent.util.		int badent[] = {
	trap.c	int	struct sysent badent[1];
		trap	callp = badent;
			callp = badent; /* illegal */
bdevsw	alloc.c	iinit	(*bdevsw[rootdev.d_major].d_open)(rootdev, 1);
			(*bdevsw[swapdev.d_major].d_open)(swapdev, 1);
	bio.c	aabread	(*bdevsw[major(dev)].d_strategy)(bp);
		aagetblk	dp = bdevsw[major(dev)].d_tab;
		bdwrite	dp = bdevsw[major(bp->b_dev)].d_tab;
		binit	struct bdevsw *bdp;
			for (i=0, bdp=bdevsw; i<nblkdev; bdp++, i++) {
		breada	(*bdevsw[major(dev)].d_strategy)(bp);
			(*bdevsw[major(dev)].d_strategy)(rabp);
		bwrite	(*bdevsw[major(bp->b_dev)].d_strategy)(bp);
		incore	dp = bdevsw[major(dev)].d_tab;
		swap	(*bdevsw[swapdev>>8].d_strategy)(bp);
	conf.70.c	nodev	struct bdevsw bdevsw[] =
		sizeof	int nblkdev = sizeof(bdevsw)/sizeof(struct bdevsw);
	conf.h	_	struct bdevsw
	confx.h		struct bdevsw bdevsw[];
	errlog.c	logstart	register struct bdevsw *bdp;
			for(bdp = &bdevsw[nblkdev-1]; bdp >= bdevsw; bdp--)
	fio.c	closef	cfunc = bdevsw[major(dev)].d_close;
		openi	(*bdevsw[maj].d_open)(dev, rw);
	lfh.c	lfget	physio(*bdevsw[lfdev>>8].d_strategy,lfdev,rwflag,0);
		lfio	(*bdevsw[lfdev>>8].d_strategy)(bp);
	prf.c	restart	(*bdevsw[i].d_open)(NODEV, 1);
	sys3.c	getmdev	if(bdevsw[d.d_major].d_tab == NULL)
		smount	(*bdevsw[d.d_major].d_open)(d, u.u_arg[2]);
			(*bdevsw[d.d_major].d_close) (d, 0);

```
                    sumount     (*bdevsw[d.d_major].d_close)(d, 0);

blkacty     errlog.c    fmtberr     eup->e_bacty = blkacty;
                        logstray    eup->e_sbacty = blkacty;
            hps.c       hpintr      blkacty =& ~(1<<HP0);
                        hpstart     blkacty =| (1<<HP0);
            hs.c        hsintr      blkacty =& ~(1<<HS0);
                        hsstart     blkacty =| (1<<HS0);
            ht.c        htintr      blkacty =& ~(1<<HT0);
                        htstart     blkacty =| (1<<HT0);
            rf.c        rfintr      blkacty =& ~(1<<RF0);
                        rfstart     blkacty =| (1<<RF0);
            rk.c        rkintr      blkacty =& ~(1<<RK0);
                        rkstart     blkacty =| (1<<RK0);
            systm.h     aagetblk    int blkacty;                /* active block devices for error log */
            tm.c        tmstart     blkacty =| (1<<TM0);
                                    blkacty =& ~(1<<TM0);

buf         bio.c       _           struct  buf     swbuf1;
                                    struct  buf     swbuf2;
                                    struct buf *
                                    struct buf *
                                    struct buf *
                                    struct buf *
                        aabread     register struct buf *bp;
                        aagetblk    register struct buf *bp;
                                    register struct buf *abp;
                        bawrite     register struct buf *bp;
                        bdwrite     register struct buf *bp;
                        bflush      register struct buf *bp;
                        binit       register struct buf *bp;
                                    register struct buf *dp;
                                    for (i=0, bp=buf; i<NBUF+NXBUF; i++, bp++) {
                        breada      register struct buf *bp, *rabp;
                        brelse      register struct buf *bp;
                                    register struct buf **backp;
                        bwrite      register struct buf *bp;
                        getablk     register struct buf *bp;
                                    register struct buf *dp;
                        getbfh      register struct buf *bp;
                        geterror    register struct buf *bp;
                        hinit       register struct buf *bp;
                        hrelse      register struct buf *bp;
                        incore      register struct buf *bp;
                        iodone      register struct buf *bp;
                        iowait      register struct buf *bp;
                        notavail    register struct buf *bp;
                        physio      register struct buf *bp;
                        swap        register struct buf *bp;
            buf.h       _           struct buf
                                    struct      buf *b_forw;        /* headed by devtab of b_dev */
                                    struct      buf *b_back;        /*  "   */
                                    struct      buf *av_forw;       /* position on free list, */
                                    struct      buf *av_back;       /*     if not BUSY*/
            bufx.h                  struct      buf     buf[NBUF+NXBUF];
                                    struct      buf     bfreelist;
```

```
                                struct      buf     heads[NHEAD];
                                struct      buf     *hfreelist;
clock.c         bisrch          bisrch(x,buf,siz)
                                struct NHIT *buf;
                                if (x < buf[mid].nloc)
                                else if( x > buf[mid].nloc)
devstart.c      devstart        register struct buf *bp;
dmc11.c         -               struct      buf *b_forw;            /* buffers held by DMC11 */
                                struct      buf *b_back;            /* buffers held by DMC11 */
                                struct      buf *dm_actq;           /* active buffers */
                                struct      buf *dm_watq;           /* waiting buffers */
                                struct      buf *dm_cmd;            /* delayed command (at most one) */
                dmcboot         register struct buf *bp,*abp;
                dmcbufq         register struct buf *bp;
                dmcclr          register struct buf     *bp,*abp;
                dmccmd          register struct buf *bp;
                dmciint         register struct buf     *bp;
                dmcioctl        struct buf *bp;
                dmcoint         register struct buf *bp,*abp;
                dmcstart        register struct buf *bp,*obp;
                                struct      buf *abp;
                dmcstrat        register struct buf *bp;
                                register struct buf     *abp;
errlog.c        fmtberr         register struct buf *bp;
                logberr         register struct buf *dp;
hps.c           hpintr          register struct buf *bp;
                hpstart         register struct buf *bp;
                hpstrate        struct buf *abp;
                                register struct buf *bp;
                hpustart        register struct buf *bp;
hs.c            hsintr          register struct buf *bp;
                hsstart         register struct buf *bp;
                hsstrate        struct buf *abp;
                                register struct buf *bp;
                                register struct buf *p1, *p2;
ht.c            hcommand        register struct buf *bp;
                htclose         register struct buf *bp;
                htintr          register struct buf *bp;
                htstart         register struct buf *bp;
                htstrate        struct buf *abp;
                                register struct buf *bp;
                tabinit         struct      buf     chtbuf;
iget.c          iupdat          struct buf *bp;
iobuf.h         NDEVREG         struct      buf     *b_forw;
                                struct      buf     *b_back;
                -               struct      buf *b_forw;            /* first buffer for this dev */
                                struct      buf *b_back;            /* last buffer for this dev */
                                struct      buf *b_actf;            /* head of I/O queue */
                                struct      buf *b_actl;            /* tail of I/O queue */
lfh.c           getlfd          register struct buf *bp;
                lfalloc         register struct buf *bp;
                lfcopy          register struct buf *bp;
                lffree          register struct buf *bp;
                lfio            register struct buf *bp;
                lfspace         struct buf *bp;
                putlfd          register struct buf *bp;
```

```
                cinit       ccp = cfree;
                            cp <= &cfree[NCLIST-1]; cp++) {
        ttyx.h  _           struct cblock cfree[];

chans   mx1.c   _           struct    chan    chans[NCHANS];
                xcp         cp = (isport)? schans+i: chans+i;
        mx2.c   _           struct    chan    chans[NCHANS];

chracty systm.h aagetblk    int chracty;            /* active character devices */

chtbuf  ht.c    hcommand    bp = &chtbuf;
                htstart     if (bp == &chtbuf && bp->b_resid == NOP) {
                            if (bp == &chtbuf) {
                tabinit     struct    buf    chtbuf;

clk_fn  clock.c clock       int (*clk_fn)()     &clock;

cmask   mx2.c   _           short     cmask[16]     ={
                nextcp      while ((gp->g_datq & cmask[gp->g_rot]) == 0) {
                            lgp->g_datq &= ~cmask[lgp->g_rot];
                            ngp->g_datq &= ~cmask[lgp->g_index];
                sdata       ngp->g_datq |= cmask[gp->g_index];

colpres ds40.c  _           extern int rowpres,colpres;
                ds401ca     colmoves = col - ((colpres > 79) ? 79: colpres);
                ds40outp    rowpres = colpres = 0;
        tty.c   _           char      colsave,colpres;
                ttyctl      colpres = *colp;

colsave tty.c   _           char      colsave,colpres;
                hqrelse     ttyctl(LCA, tp, colsave, rowsave);
                ttwrite     colsave = tp->t_col;
                            colsave = tp->t_col;

coremap main.c  main        mfree(coremap, 1, 1);
                            if ((bufbase = malloc(coremap, btoc(BSIZE)*NXBUF)) == NULL)
                                  if((cpool = malloc(coremap,NPOOL)) == NULL)
                            if (mauscore = malloc(coremap, ssize)) {
                            if((xclistbase = malloc(coremap, i)) == 0)
                            ssize = malloc(coremap, 0);
        malloc.c mfree      if ((bp = mp)==coremap && runin) {
        messag.c msginit    core = malloc(coremap, MSGMEM);
        slp.c   expand      mfree(coremap, n+newsize, a1+newsize);
                            a2 = malloc(coremap, newsize);
                            mfree(coremap, n, a1);
                newproc     a2 = malloc(coremap, n);
                swapin      if ((a = malloc(coremap, p->p_size)) == NULL)
                            mfree(coremap,p->p_size,a);
        sys1.c  exit        mfree(coremap, p->p_size, p->p_addr);
        systm.h _           struct map coremap[CMAPSIZ];        /* space for core allocation */
        text.c  xccdec      mfree(coremap, xp->x_size, xp->x_caddr);
                xexpand     if ((xp->x_caddr = malloc(coremap, xp->x_size)) != NULL) {
                xfree       mfree(coremap, xp->x_size, xp->x_caddr);
                xswap       mfree(coremap, os, xp->p_addr);
                xswapin     if ((x = malloc(coremap, xp->x_size)) == NULL) {
```

cputype	errlog.c	logstart	eup->e_cpu = cputype;
			if(cputype == 70 || cputype == 45) {
			if(cputype == 70) {
	hps.c	tabinit	#define rh70 (cputype == 70)
	hs.c		hstab.io_nreg = (cputype == 70)?NDEVREG:NDEVREG-2;
	ht.c		#define rh70 (cputype = 70)
	main.c	checkur	if(cputype == 40)
		main	if(cputype == 70)
		sureg	if (cputype==40)
	maus.c	maus	if(cputype != 40) {
			if(cputype != 40) {
	prf.c	int	if (cputype == 40)
			if (cputype != 40) {
			if (cputype == 70)
			if (cputype != 40) {
			if (cputype == 70)
	rh.c	mapalloc	if(cputype != 70 || bp->b_bcount == 0)
		rhstart	if(cputype == 70)
	systm.h	_	int cputype; /* type of cpu =40, 45, or 70 */
	trap.c	trap	if(cputype == 70) {
curpri	slp.c	setrun	if(p->p_pri < curpri)
		swtch	curpri = n;
	systm.h	_	char curpri; /* more scheduling */
	trap.c	trap	curpri = setpri(u.u_procp);
			if(curpri > nxtpri)
dh11	conf.70.c	_	&dhioctl, &dmcntrl, &dh11,
		nodev	extern dhopen(), dhclose(), dhread(), dhwrite(), dhioctl(), dmcntrl(), dh11;
	dh.c	_	struct tty dh11[NDH11];
		dhclose	tp = &dh11[dev.d_minor];
		dhcntrl	tp = &dh11[dev.d_minor];
		dhioctl	tp = &dh11[dev.d_minor];
		dhopen	pwr_init(dh11, NDH11, 0);
			tp = &dh11[line];
		dhparam	tp = &dh11[dev.d_minor];
		dhread	tp = &dh11[dev.d_minor];
		dhrint	tp = &dh11[((c)>>8)&017) + (dev*16)];
			if(tp >= &dh11[NDH11])
		dhrstrt	lineno = tp - dh11;
		dhwrite	tp = &dh11[dev.d_minor];
		dhxint	tp = &dh11[dev.d_minor*16];
	dhdm.c	_	struct tty dh11[];
		dmint	tp = &dh11[(dmader->dmcsr&017) + dev.d_minor*16];
			if(tp < &dh11[NDH11])
dh_dma	dh.c	_	struct tty_dma dh_dma[NDH11];
		dhopen	dh_dma[line].dma_blk = &dh_dma[line].dma_char;
		dhstart	dp = &dh_dma[tp->t_dev.d_minor];
dhopenf	dh.c	_	int dhopenf;
		dhopen	if (dhopenf & (1<<line))
			dhopenf |= (1 << (dev.d_minor>>4));
		dhrint	if(dhopenf & (1<<dev))
dhsar	dh.c	_	int dhsar[(NDH11+15)>>4];

```
                    dhstart     dhsar[tp->t_dev.d_minor>>4] =| lineno;
                    dhxint      bar =^ bar[dhsar[dev.d_minor];
                                dhsar[dev.d_minor] =& ~ttybit;
                    dhxoff      dhsar[tp->t_dev.d_minor>>4] |= linebit;
                                dhsar[tp->t_dev.d_minor>>4] &= ~linebit;

dk_acyl     hps.c   hpustart    dk_acyl[search)>>3]++;       /* absolute cylinder */
            systm.h aagetblk    long        dk_acyl[103];

dk_busy     clock.c clock       a = dk_busy&(~(-1<<NIOSTAT));
            hps.c   hpintr      dk_busy =& ~(1<<unit);
                    hpstart     dk_busy =| (1<<unit);
                    hpustart    dk_busy =| unitbit;
            hs.c    hsintr      dk_busy =& ~(1<<DK_N);
                    hsstart     dk_busy =| 1<<DK_N;
            rf.c    rfintr      dk_busy =& ~(1<<DK_N);
                    rfstart     dk_busy =| 1<<DK_N;
            rk.c    rkintr      dk_busy =& ~(1<<DK_N);
                    rkstart     dk_busy =| 1<<DK_N;
            rp.c    rpintr      dk_busy =& ~(1<<DK_N);
                    rpstart     dk_busy =| 1<<DK_N;
            systm.h aagetblk    int dk_busy;

dk_numb     hps.c   hpstart     dk_numb[unit] =+ 1;
            hs.c    hsstart     dk_numb[DK_N] =+ 1;
            rf.c    rfstart     dk_numb[DK_N] =+ 1;
            rk.c    rkstart     dk_numb[DK_N] =+ 1;
            rp.c    rpstart     dk_numb[DK_N] =+ 1;
            systm.h aagetblk    long        dk_numb[NIOSTAT];

dk_scyl     hps.c   hpustart    dk_scyl[(search+7)>>3]++;    /* seek distance */
            systm.h aagetblk    long        dk_scyl[103];

dk_time     clock.c clock       dk_time[a] =+ 1;
            systm.h aagetblk    long        dk_time[(4<<NIOSTAT)];

dk_unit     hps.c   hpustart    if (dp->b_active==1 && unit==dk_unit) {
            systm.h aagetblk    int dk_unit;

dk_wds      hps.c   hpstart     dk_wds[unit] =+ (bp->b_bcount>>6) & 03777;
            hs.c    hsstart     dk_wds[DK_N] =+ (bp->b_bcount>>6) & 03777;
            rf.c    rfstart     dk_wds[DK_N] =+ (bp->b_bcount>>6) & 03777;
            rk.c    rkstart     dk_wds[DK_N] =+ (bp->b_bcount>>6) & 03777;
            rp.c    rpstart     dk_wds[DK_N] =+ (bp->b_bcount>>6) & 03777;
            systm.h aagetblk    long        dk_wds[NIOSTAT];

dqlen       clock.c clock       if(dqlen) {
                                meas.m_dqlen =+ dqlen;
            hps.c   hpintr      dqlen--;
                    hpstrate    dqlen++;
                    hpustart    dqlen--;
            systm.h aagetblk    int dqlen;                  /* number of requests on disk queues */

dstflag     sys4.c  _           short       dstflag = DSTFLAG;
                    ftime       t.dstflag = dstflag;
            timeb.h _           short       dstflag;
```

dz11	conf.70.c	_	&dzioctl, &fmcntrl, &dz11,
		nodev	extern dzopen(), dzclose(), dzread(), dzwrite(), dzioctl(), dz11;
	dz.c	_	struct tty dz11[NDZ11]; /* tty structers */
		dzclose	tp = &dz11[dev.d_minor];
		dzcntrl	tp = &dz11[dev.d_minor];
		dzioctl	tp = &dz11[dev.d_minor];
		dzopen	pwr_init(dz11, NDZ11, 0);
			tp = &dz11[line];
		dzparam	tp = &dz11[dev.d_minor];
		dzread	tp = &dz11[dev.d_minor];
		dzrint	tp = &dz11[((c >> 8) & 007) + (dev << 3)];
			if(tp >= &dz11[NDZ11])
		dzrstrt	lineno = tp - dz11;
		dzscan	tp = &dz11;
		dzstart	lineno = tp - dz11;
		dzwrite	tp = &dz11[dev.d_minor];
		dzxint	btp = &dz11[dev.d_minor << 3];
dzopenf	dz.c	_	int dzopenf; /* open flag */
		dzopen	if (dzopenf & (1<<line)) {
			dzopenf =| (1 << (dev.d_minor>>3));
		dzrint	if(dzopenf&(1<<dev))
		dzscan	if ((dzopenf&(1<<i)) && (((dzadr->dzcar&n)==n)
dzspeed	dz.c	_	char dzspeed[]
		dzparam	lpr = ((dzspeed[tp->t_speeds.lobyte]<<8) | ((dev.d_minor&07) |
err	err.h	_	struct err{
			extern struct err err;
	errlog.c		struct err err = {
		errinit	if(err.e_nslot)
			mfree(err.e_map,err.e_nslot,1);
			err.e_org = err.e_ptrs;
			err.e_nxt = err.e_ptrs;
		freeslot	mfree(err.e_map,ns,(((struct errslot *)eup)-err.e_slot)+1);
		geterec	while(*err.e_org == NULL)
			sleep(&err.e_org,PZERO+1);
			eup = *err.e_org;
			*err.e_org++ = NULL;
			if(err.e_org >= &err.e_ptrs[err.e_nslot])
			err.e_org = err.e_ptrs;
		geteslot	n = malloc(err.e_map,ns);
			eup = (err_t *)(&err.e_slot[--n]);
		logberr	logberr(dp,err)
			if(err)
		puterec	*err.e_nxt++ = eup;
			if(err.e_nxt >= &err.e_ptrs[err.e_nslot])
			err.e_nxt = err.e_ptrs;
			wakeup(&err.e_org);
	ht.c	htintr	int err;
			err = HTADDR->hter;
			if (HTADDR->htcs2.hibyte>0 || err&HARD)
			err =& ~FCE;
			err =& ~(CSICOR);
			else if(state != SERR && err == 0)

	main.c	checkur	goto err;
			goto err;
			goto err;
			goto err;
			goto err;
			goto err;
			goto err;
			err:
	rx.c	NRXDEV	err:
			goto err;
	tty.c	lsgtty	goto err;
			goto err;
			goto err;
			err:
file	file.h	_	struct file
	filex.h		struct file file[NFILE];
	fio.c		struct file *
			struct file *
		closef	register struct file *fp;
			for(fp=file; fp < &file[NFILE]; fp++)
		falloc	register struct file *fp;
			for (fp = &file[0]; fp < &file[NFILE]; fp++)
		getf	register struct file *fp;
	inode.h	_	struct file *g_file;
	ioctl.c	OTHERBIT	register struct file *fp;
	mx.h	_	struct file *c_fy;
			struct file *c_fy;
	mx1.c	mpxchan	struct file *fp, *chfp, *gfp;
		mxfalloc	register struct file *fp;
	mx2.c	mxclose	register struct file *fp;
			for(fp=file; fp<&file[NFILE]; fp++)
			for(fp=file; fp < &file[NFILE]; fp++) {
		mxioctl	struct file *fp;
		mxopen	register struct file *fp;
		xcp	#define FP ((struct file *)cp)
	nmpipe.c	_	struct file *p_rfp;
			struct file *p_wfp;
		npclose	register struct file *rfp, *wfp;
		npopen	extern struct file *getf();
	pipe.c	pipe	register struct file *rf, *wf;
		readp	register struct file *fp;
		writep	register struct file *fp;
	sys2.c	open1	register struct file *fp;
		seek	register struct file *fp;
		tell	register struct file *fp;
	sys3.c	fcntl	register struct file *fp;
		stat1	struct file *fp;
	tty.c	sgtty	register struct file *fp;
groups	mx1.c	_	struct group *groups[NGROUPS];
		gpalloc	if (groups[i]==NULL) {
			groups[i]++;
		mpxchan	groups[i] = NULL;
			groups[i] = gp;
	mx2.c	_	struct group *groups[NGROUPS];

		mxclose	groups[minor(dev)] = NULL;
		xcp	if (d >= NGROUPS || groups[d] == NULL) {
			return(groups[d]);
h_blkno	ht.c	htintr	h_blkno[unit]++;
			h_blkno[unit]++;
			h_blkno[unit] = bp->b_blkno;
		htphys	h_blkno[unit] = a;
		htstart	blkno = h_blkno[unit];
		tabinit	daddr_t h_blkno[NHT], h_nxrec[NHT];
			h_blkno[unit] = 0;
h_den	ht.c	htstart	if((HTADDR->httc&03777)!=h_den[unit])
			HTADDR->httc = h_den[unit];
		tabinit	int h_den[NHT];
			h_den[unit] = (dev&010 ? P1600 : P800)|unit;
h_flags	bio.c	getbfh	h_flags |= B_WANTED;
		hrelse	if(h_flags&B_WANTED) {
			h_flags &= ~B_WANTED;
	bufx.h	—	int h_flags;
h_nxrec	ht.c	htphys	h_nxrec[unit] = ++a;
		htstrate	p = &h_nxrec[bp->b_dev&03];
		tabinit	daddr_t h_blkno[NHT], h_nxrec[NHT];
			h_nxrec[unit] = -1;
h_openf	ht.c	htclose	h_openf[unit] = 0;
		htintr	h_openf[unit] = -1;
			h_openf[unit] = (state == SBACK) ? 1 : -2;
		htstart	if (h_openf[unit] < 0) {
			if(h_openf[unit] == -2) {
			if(h_openf[unit] == -2)
		tabinit	char h_openf[NHT];
			if (h_openf[unit]) {
			h_openf[unit] = 1;
			if (h_openf[unit]) {
			h_openf[unit]++;
			h_openf[unit] = 0;
			h_openf[unit] = 0;
heads	bio.c	hinit	hfreelist = bp = heads;
			for(; bp<&heads[NHEAD-1]; bp++)
	bufx.h	—	struct buf heads[NHEAD];
hp45str	hp45.c	—	char hp45str[] {' ', ESC, 'R', 010, 0};
		hp45outp	for(cp=hp45str; *cp; cp++)
hpopenf	hps.c	hp_pwrfa	if (hpopenf&unitbit && HPADDR->hper3&(ACL|UWR)) {
		hpustart	hpopenf =| unitbit; /* This drive is open */
			hpopenf =& ~unitbit;
		tabinit	char hpopenf; /* Power fail drive open flag */
			hp_pwron = hpopenf;
			if (hpopenf & unitbit)
hpqtab	hps.c	tabinit	qhsh_t hpqtab[QHSHSZ] = {

```
                                0, &hpqtab[0], &hpqtab[0],
                                0, &hpqtab[1], &hpqtab[1],
                                0, &hpqtab[2], &hpqtab[2],
                                0, &hpqtab[3], &hpqtab[3],
                                0, &hpqtab[4], &hpqtab[4],
                                0, &hpqtab[5], &hpqtab[5],
                                0, &hpqtab[6], &hpqtab[6],
                                0, &hpqtab[7], &hpqtab[7]
                                (qhsh_t *)hptab.b_forw = hpqtab;

hpsioc      hps.c       hpstrate    hpsioc[unit]++;
                        tabinit     int hpsioc[NHP];

hpsios      hps.c       hpustart    hpsios[unit]++;
                        tabinit     int hpsios[NHP];

hpstat      hps.c       _           struct      iostat  hpstat[NHP];
                        hpintr      misc = &hpstat[unit].io_misc;
                                    hpstat[unit].io_ops++;
                                    if ((hpstat[unit].io_ops&017) == 0) {
                        hpstart     hpstat[unit].io_ops++;
                        hpustart    misc = &(hpstat[unit].io_misc);
                        tabinit     tabinit(HP0,&hpstat[0]), tabinit(HP0,&hpstat[1]),
                                    tabinit(HP0,&hpstat[2]), tabinit(HP0,&hpstat[3]),
                                    tabinit(HP0,&hpstat[4]), tabinit(HP0,&hpstat[5]),
                                    tabinit(HP0,&hpstat[6]), tabinit(HP0,&hpstat[7])

hptab       conf.70.c   nodev       extern struct iobuf hptab;
                                    &hpopen,    &nulldev,       &hpstrategy,    &hptab,         /*hp*/
            hps.c       hpintr      if (hptab.b_active) {       /* data transfer underway */
                                    dp = hptab.b_actf;
                                    hptab.b_active = 0;
                                    hptab.b_actf = 0;
                                    if (++hptab.b_errcnt > 12 ||
                                    if (recal==0 || hptab.b_errcnt==4) {
                                    hptab.b_errcnt--;
                                    hptab.b_active = 0;
                                    if (hptab.b_active) {
                                    hptab.b_active = 0;
                                    hptab.b_errcnt = 0;
                                    hptab.b_actf = dp->b_forw;
                        hpstart     if (hptab.b_active || (dp = hptab.b_actf) == 0)
                                    hptab.b_active++;
                        hpustart    if (hptab.b_actf == 0)
                                    hptab.b_actf = dp; else
                                    hptab.b_actl->b_forw = dp;
                                    hptab.b_actl = dp;
                        tabinit     struct      iobuf   hptab   tabinit(HP0,0);
                                    hptab.b_active = 0;
                                    hptab.b_actf = 0;
                                    (qhsh_t *)hptab.b_forw = hpqtab;
                                    hptab.b_flags = B_QHASH;

hputab      hps.c       _           struct      iobuf   hputab[8] {
                        hpstrate    dp = &hputab[unit];
                        hpustart    dp = &hputab[unit];
```

```
                    tabinit     dp = &hputab[sunit(dev)];

htstat      ht.c    _           struct      iostat  htstat[NHT];
                    htintr      httab.io_stp = &htstat[unit];
                                htstat[unit].io_misc++;
                                htstat[unit].io_misc++;
                                httab.io_stp = &htstat[unit];
                                htstat[unit].io_misc++;
                                htstat[unit].io_misc++;
                    htstart     htstat[unit].io_misc++;
                                htstat[unit].io_cps++;
                                htstat[unit].io_misc++;
                    tabinit     struct      iobuf   httab    tabinit(HT0,&htstat);

httab       conf.70.c nodev     extern struct iobuf httab;
                                &htopen,    &htclose,       &htstrategy,    &httab,         /*ht*/
                ht.c    htclose for(bp = httab.b_forw; bp != &httab; bp = bp->b_forw)
                        htintr  if ((bp = httab.b_actf)==0)
                                state = httab.b_active;
                                httab.b_active = 0;
                                httab.io_stp = &htstat[unit];
                                fmtberr(&httab,0);
                                if (state==SIO && ++httab.b_errcnt < 10) {
                                httab.b_active = SRETRY;
                                httab.io_stp = &htstat[unit];
                                fmtberr(&httab,0);
                                if (httab.io_erec)
                                logberr(&httab, bp->b_flags&B_ERROR);
                                httab.b_errcnt = 0;
                                httab.b_actf = bp->av_forw;
                                httab.b_active = SSEEK;
                    htstart     if ((bp = httab.b_actf) == 0)
                                httab.b_active = SCOM;
                                httab.b_active = SIO;
                                httab.b_active = SSEEK;
                                httab.b_active = SRACK;
                                httab.b_actf = bp->av_forw;
                    htstrate    if (httab.b_actf==0)
                                httab.b_actf = bp;
                                httab.b_actl->av_forw = bp;
                                httab.b_actl = bp;
                                if (httab.b_active==0)
                    tabinit     struct      iobuf   httab    tabinit(HT0,&htstat);
                                httab.b_active = 0;
                                httab.b_flags |= B_TAPE;
                                httab.io_addr = HTADDR;
                                httab.io_nreg = sh70?NDEVREG:NDEVREG-2;

icode       main.c  _           int icode[]
                    main        copyout(icode, 0, sizeof icode);

inode       acct.c  acct        register struct inode *ip;
                    sysacct     register struct inode *ip;
            alloc.c ialloc      if(dev==inode[k].i_dev && ino==inode[k].i_number)
                    update      register struct inode *ip;
                                for(ip = &inode[0]; ip < &inode[NINODE]; ip++)
```

```
            file.h      _           struct inode *f_inode;      /* pointer to inode structure */
            fio.c       access      register struct inode *ip;
                        closef      register struct inode *ip;
                        openi       register struct inode *ip;
                        owner       register struct inode *ip;
            iget.c      _           struct inode *
                        iget        register struct inode *p;
                                    for(p = &inode[0]; p < &inode[NINODE]; p++) {
                        iput        register struct inode *ip;
            ino.h       _           struct      inode
            inode.h                 struct      inode   *g_inode;
                                    struct      inode
            inodex.h                struct      inode inode[NINODE];
                                    struct      inode   *mpxip;         /* mpx virtual inode */
            ioctl.c     OTHERBIT    register struct inode *ip;
            mx1.c       addch       struct inode *ip;
                        mpxchan     struct      inode   *ip, *gip;
            mx2.c       mxclose     register struct inode *ip;
            nami.c      _           struct inode *
                        namei       register struct inode *dp;
            pipe.c      pipe        register struct inode *ip;
                        plock       register struct inode *ip;
                        prele       register struct inode *ip;
                        readp       register struct inode *ip;
                        writep      register struct inode *ip;
            rdwri.c     readi       register struct inode *ip;
                        writei      register struct inode *ip;
            subr.c      bmap        struct inode *ip;
            sys1.c      NCABLK      struct inode *ip;
                        exece       struct inode *ip;
                        gethead     register struct inode *ip;
                        getxfile    register struct inode *ip;
            sys3.c      stat1       register struct inode *ip;
                        sumount     register struct inode *ip;
                                    for(ip = &inode[0]; ip < &inode[NINODE]; ip++)
            sys4.c      utime       register struct inode *ip;
            text.c      xflsh       register struct inode *ip;
                        xfree       register struct inode *ip;
            tty.c       sgtty       register struct inode *ip;

ipc         sig.c       _           } ipc;
                        procxmt     if (ipc.ip_lock != u.u_procp->p_pid)
                                    i = ipc.ip_req;
                                    ipc.ip_req = 0;
                                    wakeup(&ipc);
                                    if (fuibyte(ipc.ip_addr) == -1)
                                    ipc.ip_data = fuiword(ipc.ip_addr);
                                    if (fubyte(ipc.ip_addr) == -1)
                                    ipc.ip_data = fuword(ipc.ip_addr);
                                    i = ipc.ip_addr;
                                    ipc.ip_data = u.inta[i>>1];
                                    i = suiword(ipc.ip_addr, 0);
                                    suiword(ipc.ip_addr, ipc.ip_data);
                                    if (suword(ipc.ip_addr, 0) < 0)
                                    suword(ipc.ip_addr, ipc.ip_data);
                                    p = &u.inta[ipc.ip_addr>>1];
```

```
                                    ipc.ip_data =| 0170000;      /* assure user space */
                                    ipc.ip_data =& ~0340;        /* priority 0 */
                                    *p = ipc.ip_data;
                                    psignal(u.u_procp, ipc.ip_data);
                                    ipc.ip_req = -1;
                        ptrace      while (ipc.ip_lock)
                                    sleep(&ipc, IPCPRI);
                                    ipc.ip_lock = p->p_pid;
                                    ipc.ip_data = u.u_ar0[R0];
                                    ipc.ip_addr = u.u_arg[1] & ~01;
                                    ipc.ip_req = u.u_arg[2];
                                    while (ipc.ip_req > 0)
                                    sleep(&ipc, IPCPRI);
                                    u.u_ar0[R0] = ipc.ip_data;
                                    if (ipc.ip_req < 0)
                                    ipc.ip_lock = 0;
                                    wakeup(&ipc);

ka6     main.c      main        i = *ka6 + USIZE;
                                UISA->r[7] = ka6[1]; /* io segment */
                                proc[0].p_addr = *ka6;
        prf.c       int         pwr_ka6 = *ka6;              /* save current u. address */
        seg.h       _           int *ka6;           /* 11/40 KISA6; 11/45 KDSA6 */
        trap.c      trap        printf("ka6 = %o\n", *ka6);

kl11    conf.70.c   _           &klioctl,   &fmcntrl,       &kl11,
                    nodev       extern klopen(), klclose(), klread(), klwrite(), klioctl(), kl11;
        kl.c        _           struct      tty kl11[NKL11+NDL11];
                    klclose     tp = &kl11[dev.d_minor];
                    klcntrl     tp = &kl11[dev.d_minor];
                    klioctl     tp = &kl11[dev.d_minor];
                    klopen      pwr_init(kl11, NKL11+NDL11, klxint);
                                tp = &kl11[dev.d_minor];
                    klread      tp = &kl11[dev.d_minor];
                    klrint      tp = &kl11[dev.d_minor];
                    klwrite     tp = &kl11[dev.d_minor];
                    klxint      tp = &kl11[dev.d_minor];

lbolt   clock.c     clock       if(++lbolt >= HZ) {
                                lbolt -= HZ;
                                wakeup(&lbolt);
        ht.c        tabinit     sleep(&lbolt,-1);
        sys4.c      ftime       ms = lbolt;
        systm.h     _           int lbolt;                   /* time of day in 60th not in time */
        tm.c        tabinit     sleep(&lbolt, -1);
        trans.c     trswrite    extern lbolt;
                                sleep((caddr_t)&lbolt, -1);

linesw  conf.70.c   sizeof      int nldisc = (sizeof(linesw)/sizeof(struct linesw))-1;
                    ttread      struct      linesw  linesw[] =
        conf.h      _           struct      linesw  linesw[];
        confx.h                 (*linesw[tp->t_ltype].l_close)(dev, tp);
        dh.c        dhclose     (*linesw[tp->t_ltype].l_open)(tp);
                    dhopen      (*linesw[tp->t_ltype].l_read)(tp);
                    dhread      (*linesw[tp->t_ltype].l_rcvd)(c, tp);
                    dhrint
```

```
                dhstart         if((cnt = (*linesw[tp->t_ltype].l_xdma)(tp,dp)) < 0) {
                dhwrite         (*linesw[tp->t_ltype].l_write) (tp);
dhdm.c          dmint           (*linesw[tp->t_ltype].l_dst) (tp, dmadr->dmcsr,
dj.c            djclose         (*linesw[tp->t_ltype].l_close)(dev, tp);
                djopen          (*linesw[tp->t_ltype].l_open)(tp);
                djread          (*linesw[tp->t_ltype].l_read)(tp);
                djrint          (*linesw[tp->t_ltype].l_rcvd)(c, tp);
                djwrite         (*linesw[tp->t_ltype].l_write)(tp);
                djxint          c = (*linesw[tp->t_ltype].l_xmtd)(tp, 0);
                                (*linesw[tp->t_ltype].l_xmtd)(tp, 1);
dz.c            dzclose         (*linesw[tp->t_ltype].l_close)(dev, tp);
                dzopen          (*linesw[tp->t_ltype].l_open)(tp);
                dzread          (*linesw[tp->t_ltype].l_read) (tp);
                dzrint          (*linesw[tp->t_ltype].l_rcvd) (c,tp);
                dzscan          (*linesw[tp->t_ltype].l_dst)
                dzwrite         (*linesw[tp->t_ltype].l_write) (tp);
                dzxint          c = (*linesw[tp->t_ltype].l_xmtd) (tp, 0);
                                (*linesw[tp->t_ltype].l_xmtd) (tp, 1);
ioctl.c         ttioccom        if (t >= nldisc || *linesw[t].l_ioctl == &nodev) {
                                (*linesw[tp->t_line].l_ioctl)(com, tp, addr, LUNSET);
                                (*linesw[tp->t_line=t].l_ioctl)(com, tp, addr, LSET);
                                (*linesw[tp->t_line].l_ioctl)(com, tp, addr, LRESET);
kl.c            klclose         (*linesw[tp->t_ltype].l_close)(dev, tp);
                klopen          (*linesw[tp->t_ltype].l_open)(tp);
                klread          (*linesw[tp->t_ltype].l_read)(tp);
                klrint          (*linesw[tp->t_ltype].l_rcvd)(c, tp);
                klwrite         (*linesw[tp->t_ltype].l_write)(tp);
mx1.c           mpxchan         for (i=0; linesw[i].l_open; i++) {
                                if (linesw[i].l_read==mcread) {
mx2.c           mxread          more = (*linesw[cp->c_line].l_read)(cp->c_ttyp);
                mxwrite         waddr = (caddr_t)(*linesw[line].l_write)(tp);
prf.c           pwr_init        (*linesw[tp->t_ltype].l_dst)(tp,
ttdma.c         ttxdma          while((c = (*linesw[tp->t_ltype].l_xmtd)(tp, 0))){
                                (*linesw[tp->t_ltype].l_xmtd)(tp, 1);
tty.c           lsgtty          (*linesw[tp->t_ltype].l_ioctl)(OLDSGTTY, tp, v);
                                if ((unsigned)i >= nldisc || *linesw[i].l_ioctl == &nodev)
                                (*linesw[i].l_ioctl)(OLDSGTTY, tp, 0, LRESET);
                                (*linesw[tp->t_ltype].l_ioctl)(OLDSGTTY, tp, 0, LUNSET);
                                (*linesw[tp->t_ltype=i].l_ioctl)(OLDSGTTY, tp, 0, LSET);
                ttstart         c = (*linesw[tp->t_ltype].l_xmtd)(tp, 0);
                                (*linesw[tp->t_ltype].l_xmtd)(tp, 1);

lks             clock.c         clock           *lks = 0115;
                main.c          main            lks = CLOCK1;
                                                if(fuiword(lks) == -1) {
                                                lks = CLOCK2;
                                                if(fuiword(lks) == -1)
                                                *lks = 0115;
                prf.c           restart         *lks = 0115;
                systm.h         _               int *lks;                       /* pointer to clock device */

lpuser          vtll.c          _               int lpuser;
                                vtopen          if(lpuser == 0)
                vtlp.c          _               int     lpuser;                         /* lp opened for ?? -- see vtuser */
                                vtlpclos        if(lpuser&VTUSER) {
                                                lpuser =& ~(VTUSER|VTSYS);
```

		vtlpopen	lpuser = lpstate = 0;
			if(lpuser&VTUSER) {
			if((lpuser&(VTSYS|VTUSER)) == 0)
			lpuser = VTSYS;
			lpuser = VTUSER;

m_eot	mx2.c	— mxread	char m_eot[] ={ M_EOT, 0, 0, 0}; iomove(m_eot, sizeof m_eot, B_READ);

maphp45	hp45.c	—
		hp45inpu
		hp45outp
			char maphp45[] {
			for (cp=maphp45; *cp++;)
			for(cp=maphp45; *cp++;)

maptab	tty.c	— canon	char maptab[] mc = maptab[c];

mapwant	rh.c	—
		mapalloc
		mapfree
			int mapwant;
			mapwant++;
			if(mapwant) {
			mapwant = 0;

mausend	confx.h main.c	— main	int mausend; /* Last core address of MAUS regions */ mausend = mauscore + ssize;
mausmap	main.c	main	for(i=0; mausmap[i].boffset != -1; i++) { if((maussize=mausmap[i].boffset+mausmap[i].bsize)>ssize)

	maus.c	—
		maus
			struct mausmap mausmap[];
			u.u_msav[j].ms_uisd = i | ABS | (mausmap[min].bsize-1)<<8;
			u.u_msav[j].ms_uisa = mausmap[min].boffset + mauscore;

	maus.h	—	struct mausmap { struct mausmap mausmap[];

	mem.c	mmoff	struct mausmap mausmap[] {
			bn = (mausmap[dev].bsize<<6) - u.u_offset;
			return(((long)mauscore+mausmap[dev].boffset)<<6)+u.u_offset);

maxmem	main.c	checkur
		main
			if(nt+nd+ns+USIZE > ((maxmem)>(32*32) && sep==0)? (32*32): maxmem))
			for(;maxmem<memsiz;) {
			maxmem++;
			maxmem =- MPOOL;
			maxmem =- ssize;
			maxmem -= msginit();
			maxmem =- 1;
			ctok(ssize), ctoik(ssize), ctok(maxmem), ctoik(maxmem));
			maxmem = min(maxmem, MAXMEM);

	systm.h	—	int maxmem; /* actual max memory per process */

maxsize	slp.c	—
		sched
			int maxsize;
			maxsize = -1;
			if(maxsize < rp->p_size) {
			maxsize = rp->p_size;
			} else if(maxsize < 0) {
			if(p2!=0 && (maxsize)>=0 || ns!=1 ||

meas	bio.c	aabread
		aagetblk
			meas.m_dread++;
			meas.m_gbcnt++;
			meas.m_agbcnt++;

```
                                meas.m_fndblk++;
                                meas.m_noblk++;
                                meas.m_anoblk++;
                    breada      meas.m_dread++;
                                meas.m_dread++;
                    bwrite      meas.m_dwrit++;
                    getablk     meas.m_gbcnt++;
                                meas.m_agbcnt++;
                                meas.m_noblk++;
                                meas.m_anoblk++;
                    getbfh      meas.m_physio++;
                                meas.m_nphys++;
                    physio      meas.m_dread++;
                                meas.m_dwrit++;
                    swap        meas.m_swap++;
                                meas.m_dwrit++;
                                meas.m_dread++;
          clock.c   clock       meas.m_dqlen =+ dqlen;
                                meas.m_dqocc++;
                                meas.m_rqlen =+ rqlen;
                                meas.m_rqocc++;
                                meas.m_sqlen =+ sqlen;
                                meas.m_sqocc++;
          fio.c     falloc      meas.m_fovfl++;
          iget.c    iget        meas.m_iovfl++;
          slp.c     newproc     meas.m_fork++;
                    swtch       meas.m_swtch++;
          sys1.c    exece       meas.m_exec++;
                    fork        meas.m_povfl++;
          sysmesx.h _           struct M_SYSMEAS meas;
          text.c    xalloc      meas.m_tovfl++;
          tty.c     ttyinput    meas.m_temin++;
                    ttyoutl     meas.m_termop++;

memsiz    main.c    _           int memsiz  [MAXCORE];
                    main        memsiz =- i;
                                for(;maxmem<memsiz;) {
          systm.h   _           int memsiz;                 /* max core allowed to be used */

mount     alloc.c   getfs       register struct mount *p;
                                for(p = &mount[0]; p < &mount[NMOUNT]; p++)
                    iinit       mount[0].m_bufp = cp;
                                mount[0].m_dev = rootdev;
                                mount[0].m_flags = M_INCOR;
                    putfs       register struct mount *mp;
                                for(mp = &mount[0]; mp < &mount[NMOUNT]; mp++)
                    rofs        register struct mount *p;
                                for(p = &mount[0]; p < &mount[NMOUNT]; p++)
                    update      register struct mount *mp;
                                for(mp = &mount[0]; mp < &mount[NMOUNT]; mp++)
          iget.c    iget        register struct mount *ip;
                                for(ip = &mount[0]; ip < &mount[NMOUNT]; ip++)
          nami.c    namei       if(mount[i].m_bufp != NULL)
                                if(mount[i].m_dev == d) {
                                dp = mount[i].m_inodp;
          sys3.c    smount      register struct mount *mp, *smp;
```

```
                                for(mp = &mount[0]; mp < &mount[NMOUNT]; mp++) {
                sumount         register struct mount *mp;
                                for(mp = &mount[0]; mp < &mount[NMOUNT]; mp++)
                                } mount[NMOUNT];
        systm.h     -           struct      mount
                    int

mpid    slp.c       newproc     mpid++;
                                if(mpid < 0) {
                                mpid = 0;
                                if ((rpp->p_pid==mpid)||(rpp->p_pgrp == mpid))
                                rpp->p_pid = mpid;
        systm.h     -           int mpid;               /* generic for unique process id's */

mpxdev  mx1.c       mpxchan     if (mpxdev < 0) {
                                mpxdev = (dev_t)(i<<8);
                                if (mpxdev < 0) {
                                ip->i_un.i_rdev = (daddr_t)(mpxdev+i);
                    xcp         dev_t       mpxdev  = -1;

mpxip   inodex.h    -           struct      inode   *mpxip;         /* mpx virtual inode */
        mx2.c       mcread      prele(mpxip);
                    mxclose     if (ip == mpxip) {
                                mpxip = NULL;
                    mxioctl     if (mpxip == NULL) {
                                mpxip = gp->g_inode;
                    mxopen      if (gp->g_inode == mpxip) {
                                plock(mpxip);
                                prele(mpxip);
                    namei       if (mpxip!=NULL && c=='!')
                                if (c == '!' && mpxip != NULL) {
                                plock(mpxip);
                                mpxip->i_count++;
                                return(mpxip);

mpxline mx1.c       -           int mpxline;
                    mpxchan     mpxline = i;
                                cp->c_line = cp->c_oline = mpxline;
                    mx2.c       -           int mpxline;
                    mxopen      cp->c_line = cp->c_oline = mpxline;

msgbase messag.c    -           paddr_t     msgbase;                /* pointer to space for messages */
                    messag      sleep(&msgbase, PMSG);
                    msgfree     wakeup(&msgbase);
                    msghmov     paddr += msgbase;
                    msginit     msgbase = core;
                                msgbase--;
                                msgbase <<= 6;
                    msgmove     paddr += msgbase + sizeof(struct msghdr);

msgbuf  prf.c       -           char        *msgbufp msgbuf;        /* Next saved printf character */
                    putchar     if (msgbufp >= &msgbuf[MSGBUFS])
                                msgbufp = msgbuf;
        systm.h     -           char        msgbuf[MSGBUFS];        /* saved "printf" characters */

msgbufp prf.c       -           char        *msgbufp msgbuf;        /* Next saved printf character */
                    putchar     *msgbufp++ = rc;
```

```
                                    if (msgbufp >= &msgbuf[MSGBUFS])
                                    msgbufp = msgbuf;

msgmap      messag.c    _           struct      map     msgmap[MMAPSIZ];        /* space for message allocation */
                        messag      if((hp = malloc(msgmap, (n+movrhead)>>6)) == NULL) {
                        msgfree     mfree(msgmap, (size+movrhead)>>6, hp);
                        msginit     mfree(msgmap, MSGMEM, 1);

msgqhdr     ipcomm.h    _           struct msgqhdr
            messag.c                struct msgqhdr msgqhdr[NMQHDR];
                        messag      register struct msgqhdr *rqp;
                                    for(rqp = &msgqhdr[0];rqp < &msgqhdr[NMQHDR];rqp++)
                        mqsrch      register struct msgqhdr *rqp;
                                    for(rqp = &msgqhdr[0]; rqp < &msgqhdr[NMQHDR]; rqp++)
                        msgflush    register struct msgqhdr *rqp1, *rqp2;
                        msgrecv     struct msgqhdr *qp;
                        msgremov    struct msgqhdr *qp;
                        msgsend     register struct msgqhdr *qp;

mxnmbuf     mx2.c       _           char        mxnmbuf[NMSIZE];
                        mcread      np = mxnmbuf;
                        mpxname     np = mxnmbuf;
                                    while (np < &mxnmbuf[NMSIZE]) {
                                    nmsize = np - mxnmbuf;

mxnmcp      mx2.c       _           struct      chan *mxnmcp;

nblkdev     bio.c       aagetblk    if(major(dev) >= nblkdev)
                        binit       for (i=0, bdp=bdevsw; i<nblkdev; bdp++, i++) {
            conf.70.c   sizeof      int nblkdev = sizeof(bdevsw)/sizeof(struct bdevsw);
            confx.h     _           int nblkdev;
            errlog.c    logstart    extern int nblkdev;
                                    for(bdp = &bdevsw[nblkdev-1]; bdp >= bdevsw; bdp--)
            fio.c       openi       if(maj >= nblkdev)
            prf.c       restart     for (i=0; i<nblkdev; i++)
            sys3.c      getmdev     if(d.d_major >= nblkdev)

nchrdev     conf.70.c   sizeof      int nchrdev = sizeof(cdevsw)/sizeof(struct cdevsw);
            confx.h     _           int nchrdev;
            fio.c       openi       if(maj >= nchrdev)
            prf.c       restart     for (i=0; i<nchrdev; i++)

ndh11       dh.c        _           int ndh11 NDH11;
            dhdm.c                  int ndh11;
                        dmint       if (tp < &dh11[ndh11])

ndn11       dn.c        _           int ndn11   =       NDN11;
                        dnopen      if(physdn >= ndn11 || dnadr->dn_reg & PWI)

ndz11       dz.c        _           int ndz11 NDZ11;    /* number of dz11 lines */

nkl11       kl.c        _           int nkl11   NKL11+NDL11;

nldisc      conf.70.c   sizeof      int nldisc = (sizeof(linesw)/sizeof(struct linesw))-1;
            confx.h     _           int nldisc;
            ioctl.c     ttioccom    if (t >= nldisc || *linesw[t].l_ioctl == &nodev) {
```

```
            tty.c       lsgtty      if ((unsigned)i >= nldisc || *linesw[i].l_ioctl == &nodev)

nmpipe      nmpipe.c    —           struct nmpipe {
                                    } nmpipe[NNAMPIPE];
                        npclose     register struct nmpipe *npp;
                                    npp = &nmpipe[dev.d_minor];
                        npioctl     register struct nmpipe *npp;
                                    npp = &nmpipe[dev.d_minor];
                        npopen      register struct nmpipe *npp;
                                    npp = &nmpipe[dev.d_minor];
                        npread      register struct nmpipe *npp;
                                    npp = &nmpipe[dev.d_minor];
                        npwrite     register struct nmpipe *npp;
                                    npp = &nmpipe[dev.d_minor];

nmsize      mx2.c       —           int nmsize;
                        mcread      while (nmsize--)
                        mpxname     nmsize = np - mxnmbuf;
                        mxread      count += nmsize;

ns          errlog.c    freeslot    register ns, sps;
                                    ns = (eup->e_hdr.e_len+sizeof(struct errslot)-1)/sizeof(struct errslot);
                                    mfree(err.e_map,ns,(((struct errslot *)eup)-err.e_slot)+1);
                        geteslot    register ns, *p;
                                    ns = (size+sizeof(struct errslot)-1)/sizeof(struct errslot);
                                    n = malloc(err.e_map,ns);
                                    ns *= sizeof(struct errslot)/sizeof(int);
                                    } while(--ns);
            main.c      checkur     checkur(nt, nd, ns, sep)
                                    if(ctos(nd) > first || 8-ctos(ns) <= last)
                                    if(ctos(nd) + ctos(ns) > 8)
                                    if(ctos(nt) + ctos(nd) > first || 8-ctos(ns) <= last)
                                    if(ctos(nt) + ctos(nd) + ctos(ns) > 8)
                                    if(nt+nd+ns+USIZE > ((maxmem>(32*32) && sep==0)? (32*32): maxmem))
                        estabur     estabur(nt, nd, ns, sep, xrw)
                                    if(checkur(nt, nd, ns, sep))
                                    a =+ ns;
                                    while(ns >= 128) {
                                    ns =- 128;
                                    if(ns) {
                                    *--dp = ((128-ns)<<8) | RW | ED;
            slp.c       —           int ns;
                        sched       ns = 1;
                                    if(rp->p_stat==SSLEEP && rp->p_ctime > ns) {
                                    ns = rp->p_ctime;
                                    } else if(ns == 1 &&
                                    if(p2!=0 && (nmsize)=0 || ns!=1 ||

nswap       conf.70.c   sizeof      int nswap   8359;
            main.c      main        mfree(swapmap, nswap, swplo);
            systm.h     —           int nswap;                  /* size of swap space */

nttype      conf.70.c   sizeof      int nttype = sizeof(termsw)/sizeof(struct termsw);
            confx.h     —           int nttype;
            tty.c       lsgtty      if ((unsigned)i >= nttype)
                        ttyioctl    if ((unsigned)termblk.st_termt >= nttype) {
```

```
nxtpri      slp.c       setrun      else if(p->p_pri < nxtpri)
                                    nxtpri = p->p_pri;
                        swtch       nxtpri = n;
            systm.h     _           char        nxtpri;             /* more scheduling */
            trap.c      trap        if(curpri > nxtpri)

ooswap      sys1.c      exece       panic(ooswap);
            sys3.c      smount      panic(ooswap);
            systm.h     _           char        *ooswap;            /* out of swap space printf message */
            text.c                  char *ooswap "out of swap space";
                        xalloc      panic(ooswap);
                        xswap       panic(ooswap);

p1          clock.c     int         register struct callo *p1, *p2;
                                    p1 = &callout[0];
                                    while(p1->c_time <= 0 && p1->c_func) {
                                    (*p1->c_func) (p1->c_arg);
                                    p2 = p1;
                                    while(p2->c_func = (++p1)->c_func) {
                                    p2->c_time = p1->c_time;
                                    p2->c_arg = p1->c_arg;
                        kiltout     register struct callo *p1;
                                    p1 = &callout[0];
                                    while(p1->c_func != fun || p1->c_arg != arg) {
                                    if(p1->c_func == 0) {
                                    p1++;
                                    (p1+1)->c_time =+ p1->c_time;
                                    while((p1+1)->c_func != 0) {
                                    p1->c_time = (p1+1)->c_time;
                                    p1->c_func = (p1+1)->c_func;
                                    p1->c_arg  = (p1+1)->c_arg;
                                    p1++;
                                    p1->c_func = 0;
                        timeout     register struct callo *p1, *p2;
                                    p1 = &callout[0];
                                    while(p1->c_func != 0 && p1->c_time <= t) {
                                    t =- p1->c_time;
                                    p1++;
                                    p1->c_time =- t;
                                    p2 = p1;
                                    while(p2 >= p1) {
                                    p1->c_time = t;
                                    p1->c_func = fun;
                                    p1->c_arg = arg;
            hps.c       hpstrate    register char *p1, *p2;
                                    p1 = &hp_sizes[part(bp->b_dev)];
                                    if (dp->status || bp->b_blkno >= p1->nblocks) {
                                    bp->cylin = bp->b_blkno/(NSECT*NTRACK) + p1->cyloff;
                                    if ((p1 = dp->b_actf)==0) {
                                    for (; p2 = p1->av_forw; p1 = p2) {
                                    if (p1->cylin <= bp->cylin
                                     || p1->cylin >= bp->cylin
                                    p1->av_forw = bp;
            hs.c        hsstrate    register struct buf *p1, *p2;
                                    if ((p1 = hstab.b_actf) == 0) {
```

```
                                for (; p2 = p1->av_forw; p1 = p2)
                                p1->av_forw = bp;
        rf.c        rfstrate    register struct buf *p1, *p2;
                                if ((p1 = rftab.b_actf) == 0) {
                                for (; p2 = p1->av_forw; p1 = p2)
                                p1->av_forw = bp;
        rk.c        rkstrate    register struct buf *p1, *p2;
                                if ((p1 = rktab.b_actf) == 0) {
                                for (; p2 = p1->av_forw; p1 = p2)
                                p1->av_forw = bp;
        rp.c        rpstrate    register char *p1, *p2;
                                p1 = &rp_sizes[bp->b_dev.d_minor&07];
                                if(bp->b_blkno >= p1->nblocks) {
                                bp->cylin = p1->cyloff;
                                p1 = bp->b_blkno;
                                p2 = lrem(p1, 10);
                                p1 = ldiv(p1, 10);
                                bp->trksec = (p1%20)<<8 | p2;
                                bp->cylin =+ p1/20;
                                if ((p1 = rptab.b_actf)==0) {
                                for (; p2 = p1->av_forw; p1 = p2) {
                                if (p1->cylin <= bp->cylin
                                 || p1->cylin >= bp->cylin
                                p1->av_forw = bp;
        slp.c       _          struct proc *p1, *p2;
                    sched      p1 = rp;
                               rp = p1;
                                   a!=p1->p_nice || (p1->p_time)>=3 && n)>=2))) {
        sys1.c      fork       register struct proc *p1, *p2;
                               p1 = u.u_procp;
                               u.u_ar0[R0] = p1->p_pid;

p2      clock.c     clock      register struct callo *p2;
                               p2 = &callout[0];
                               while(p2->c_time<=0 && p2->c_func!=0)
                               p2++;
                               p2->c_time--;
                    int        register struct callo *p1, *p2;
                               p2 = p1;
                               while(p2->c_func = (++p1)->c_func) {
                               p2->c_time = p1->c_time;
                               p2->c_arg = p1->c_arg;
                               p2++;
                    timeout    register struct callo *p1, *p2;
                               p2 = p1;
                               while(p2->c_func != 0)
                               p2++;
                               if (p2 >= &callout[NCALL-2])        /* kiltout assumes last entry is 0 */
                               while(p2 >= p1) {
                               (p2+1)->c_time = p2->c_time;
                               (p2+1)->c_func = p2->c_func;
                               (p2+1)->c_arg = p2->c_arg;
                               p2--;
        hps.c       hpstrate   register char *p1, *p2;
                               for (; p2 = p1->av_forw; p1 = p2) {
                               if (p2->b_pri < bp->b_pri)
```

```
                        if (p2->b_pri > bp->b_pri)
                         && bp->cylin <  p2->cylin
                         && bp->cylin >  p2->cylin)
                        bp->av_forw = p2;
                        while (p2) {
                        if (p2->b_pri != bp->b_pri)
                        p2->b_pri--;
                        p2 = p2->av_forw;
hs.c        hsstrate    register struct buf *p1, *p2;
                        for (; p2 = p1->av_forw; p1 = p2)
                        if (p2->b_pri > bp->b_pri)
                        bp->av_forw = p2;
                        while (p2) {
                        if (p2->b_pri > bp->b_pri)
                        p2->b_pri--;
                        p2 = p2->av_forw;
rf.c        rfstrate    register struct buf *p1, *p2;
                        for (; p2 = p1->av_forw; p1 = p2)
                        if (p2->b_pri > bp->b_pri)
                        bp->av_forw = p2;
                        while (p2) {
                        if (p2->b_pri > bp->b_pri)
                        p2->b_pri--;
                        p2 = p2->av_forw;
rk.c        rkstrate    register struct buf *p1, *p2;
                        for (; p2 = p1->av_forw; p1 = p2)
                        if (p2->b_pri > bp->b_pri)
                        bp->av_forw = p2;
                        while (p2) {
                        if (p2->b_pri > bp->b_pri)
                        p2->b_pri--;
                        p2 = p2->av_forw;
rp.c        rpstrate    register char *p1, *p2;
                        p2 = lrem(p1, 10);
                        bp->trksec = (p1%20)<<8 | p2;
                        for (; p2 = p1->av_forw; p1 = p2) {
                        if (p2->b_pri < bp->b_pri)
                        if (p2->b_pri > bp->b_pri)
                         && bp->cylin <  p2->cylin
                         && bp->cylin >  p2->cylin)
                        bp->av_forw = p2;
                        while (p2) {
                        if (p2->b_pri != bp->b_pri)
                        p2->b_pri--;
                        p2 = p2->av_forw;
slp.c       _
            sched       struct proc *p1, *p2;
                        p2 = 0;
                        p2 = rp;
                        p2 = rp;
                        p2 = rp;
                        if(p2!=0 && (maxsize>=0 || ns!=1 ||
                        rp = p2;
sys1.c      fork        register struct proc *p1, *p2;
                        for(p2 = &proc[0]; p2 < &proc[NPROC]; p2++)
                        if(p2->p_stat == NULL)
                        u.u_ar0[R0] = p2->p_pid;
```

```
partab      hf.c        hfxint      c =| CPRES | partab[c]&0200;
            partab.c    _           char partab[] {
            tty.c       ttxint       (((tp->t_flags&(ODDP|EVENP))==ODDP) ? ~partab[c] : partab[c])&0200;
                        ttyoutl     ctype = partab[c&0177];
            ttyx.h      _           char partab[];

pir_fn      clock.c     int         int (*pir_fn)()     &pir;
            prf.c                   extern *pirr, (*pir_fn)();
                                    pwr_save.pf_addr = pir_fn;
                                    pir_fn = pwrfail;
                        restart     pir_fn = pwr_save.pf_addr;

pirr        clock.c     clock       extern *pirr;
                                    *pirr =| 1 << (1+8);        /* PIRQ 1 */
                        int         extern *pirr;
                                    pirr->hibyte =& ~(1 << ((*pirr & 016) >> 1));
            prf.c                   extern *pirr, (*pir_fn)();
                                    pwr_save.pf_pirr = *pirr;
                                    *pirr = 1 << (1+8); /* Priority 1 */
                        restart     *pirr = pwr_save.pf_pirr;

power       prf.c       int         extern char power, pwr_flg;
                                    power--;
                                    power = 0;

proc        clock.c     clock       register struct proc *pp;
                                    for(pp = &proc[0]; pp < procend; pp++)
            main.c      main        proc[0].p_addr = *ka6;
                                    proc[0].p_size = USIZE;
                                    proc[0].p_stat = SRUN;
                                    proc[0].p_flag =| SLOAD|SSYS;
                                    u.u_procp = &proc[0];
                                    VTNEWPROC(p=proc,PR_NAME,"UNIX Scheduler",15);
            messag.c    mqsrch      {   register struct proc *rpp;
            prf.c       int         register struct proc *pp;
                                    for (pp = &proc[0]; pp<procend; pp++)
            proc.h      _           struct      proc {
            procx.h                 struct      proc proc[NPROC];
                                    struct proc *procMAX;       /* highwater mark of proc table */
            sig.c       fsig        struct proc *p;
                        issig       register struct proc *p, *q;
                                    for(q = &proc[0];q < &proc[NPROC];q++)
                        psignal     register struct proc *p;
                        ptrace      register struct proc *p;
                                    for (p=proc; p < &proc[NPROC]; p++)
                        signal      register struct proc *p;
                                    for(p = &proc[0]; p < &proc[NPROC]; p++)
                        stop        register struct proc *pp, *cp;
                                    for (pp = &proc[0]; pp < &proc[NPROC]; pp++)
            slp.c       HASH        struct proc *slpque[SQSIZE];
                                    register struct proc *rp;
                        _           struct proc *procend {&proc[1]};
                                    struct proc *p1, *p2;
                        newproc     struct proc *p, *up;
                                    register struct proc *rpp;
```

```
                                    struct proc *pend;
                                    for(rpp = &proc[0]; rpp < &proc[NPROC]; rpp++) {
                    sched           register struct proc *rp;
                                    proc[0].p_nice = rp->p_nice;
                                    proc[0].p_nice = 0;
                                    for(rp = &proc[0]; rp < procend; rp++) {
                    setrq           struct proc *p;
                                    register struct proc *q;
                    setrun          register struct proc *p;
                    shuffle         struct proc *pp;
                                    for(pp= &proc[1]; pp< procend; pp++)
                                    for(pp= &proc[1]; pp< procend;pp++)
                    swapin          struct proc *pp;
                                    register struct proc *p;
                    swtch           register struct proc *p, *q;
                                    static struct proc *pp, *pq;
                                    retu(proc[0].p_addr);
                    wakeup          register struct proc *p, *q;
          sys1.c    exit            register struct proc *p, *q;
                                    for(q = &proc[1]; q < procend; q++) {
                                    psignal(&proc[1], SIGCLD);
                    fork            register struct proc *p1, *p2;
                                    for(p2 = &proc[0]; p2 < &proc[NPROC]; p2++)
                    freeproc        register struct proc *p;
                    wait            register struct proc *p;
                                    for(p = &proc[0]; p < procend; p++)
          sys4.c    kill            register struct proc *p;
                                    p = &proc[0];
                                    p = &proc[2];
                    ssig            register struct proc *p;
                                    for(p = &proc[0]; p < &proc[NPROC]; p++)
          tty.c     ttyopen         register struct proc *pp;
          vs.c      vsopen          register struct proc *pp;
          vtmon.c   vtprinit        vtprocent(proc+(pcp-pr_frms),PR_CLEAR);
                    vtprocen        vtprocent(procp,field,value,valu2)  struct proc *procp,*value;  {
                                    register struct proc    *prp;
                                    pcp = &pr_frms[procn=(prp-proc)];
                                    vtccopy(pr_frms[value-proc].pr_name, pcp->pr_name, 15);

procend   clock.c   clock           for(pp = &proc[0]; pp < procend; pp++)
          prf.c     int             for (pp = &proc[0]; pp<procend; pp++)
          procx.h   --              struct proc *procend;       /* highwater mark of proc table */
          slp.c                     struct proc *procend {&proc[1]};
                    newproc         procend = pend;
                    sched           for(rp = &proc[0]; rp < procend; rp++) {
                    shuffle         for(pp= &proc[1]; pp< procend; pp++)
                                    for(pp= &proc[1]; pp< procend;pp++)
          sys1.c    exit            for(q = &proc[1]; q < procend; q++) {
                    wait            for(p = &proc[0]; p < procend; p++)
          sys4.c    kill            for(; p < procend; p++) {

pwr_flg   prf.c     int             extern char power, pwr_flg;
                                    pwr_flg = 1;
                                    pwr_flg = 0;

pwr_ka6   prf.c     int             extern pwr_ka6;
```

```
                                        pwr_ka6 = *ka6;             /* save current u. address */

rablock     rdwri.c     readi           rablock = bn+1;
                                        bp = breada(dev, bn, rablock);
            subr.c      bmap            rablock = 0;
                                        rablock = ip->i_un.i_addr[bn+1];
                                        rablock = xget(paddr(bp) + 2*(i+1));
            systm.h     _               int rablock;                /* block to be read ahead */

rebstr      syscb.c     _               char        rebstr[14];     /* file name to reboot terminated by newline */
                        syscb           p = rebstr;
                                        if(p == &rebstr[sizeof(rebstr)])

rebunit     syscb.c     _               int rebunit;        /* device and unit spec for DEC YC ROM */
                        syscb           rebunit = u.u_ar0[R0];

regloc      sig.c       procxmt         if (p == &u.u_ar0[regloc[i]])
            sys1.c      setregs         for(cp = &regloc[0]; cp < &regloc[6];)
            systm.h     _               char        regloc[];               /* locs. of saved user registers (trap.c) */
            trap.c      int             char        regloc[9]

rootdev     alloc.c     iinit           (*bdevsw[rootdev.d_major].d_open)(rootdev, 1);
                                        bp = bread(rootdev, 1);
                                        mount[0].m_dev = rootdev;
            conf.70.c   sizeof          dev_t       rootdev = makedev(5, 15);
            main.c      main            rootdir = iget(rootdev, ROOTINO);
            mx1.c       mpxchan         if ((ip=ialloc(rootdev))==NULL)
            pipe.c      pipe            ip = ialloc(rootdev);
            systm.h     _               int rootdev;                /* dev of root see conf.c */

rootdir     main.c      main            rootdir = iget(rootdev, ROOTINO);
                                        rootdir->i_flag =& ~ILOCK;
                                        u.u_cdir = u.u_rdir = rootdir;
                                        rootdir->i_count =+ 2;
            nami.c      namei           dp = rootdir;
            sys4.c      chroot          u.u_rdir = rootdir;
            systm.h     _               int *rootdir;               /* pointer to inode of root directory */

rowpres     ds40.c      _               extern int rowpres,colpres;
                        ds401ca         rowmoves = row - rowpres%(atp->t_lrow+1);
                        ds40outp        rowpres = colpres = 0;
            tty.c       _               char        rowsave,rowpres;
                        ttyctl          rowpres = tp->t_row;

rowsave     tty.c       _               char        rowsave,rowpres;
                        hqrelse         ttyctl(LCA, tp, colsave, rowsave);
                        ttwrite         rowsave = tp->t_row;
                                        rowsave = tp->t_row;

runin       bio.c       physio          if(runin) {
                                        runin = 0;
                                        wakeup(&runin);
            clock.c     clock           if(runin!=0) {
                                        runin = 0;
                                        wakeup(&runin);
            malloc.c    mfree           if ((bp = mp)==coremap && runin) {
```

```
                                        runin = 0;
                                        wakeup(&runin);             /* Wake scheduler when freeing core */
            slp.c       HASH            if(runin != 0) {
                                        runin = 0;
                                        wakeup((caddr_t)&runin);
                        sched           runin++;
                                        sleep(&runin, PSWP);
            systm.h     _               char        runin;                  /* scheduling flag */

runlock     slp.c       expand          runlock++;
                                        runlock++;
                        sched           if(runlock)
                                        runlock = 0;
                        shuffle         wakeup(&runlock);
            sysch.c     lock            sleep(&runlock,-1);
                        proclock        runlock++;
                        punlock         runlock++;
                        textlock        runlock++;
                        tunlock         runlock++;
            systm.h     _               char        runlock;                /* sched flag used in text&data locking */

runout      slp.c       sched           runout++;
                                        sleep(&runout, PSWP);
                        setrun          if(runout != 0 && (p->p_flag&SLOAD) == 0) {
                                        runout = 0;
                                        wakeup((caddr_t)&runout);
            sys1.c      exit            if(runout !=0)
                                        runout=0;
                                        wakeup(&runout);
            sysch.c     lock            if(runout)
                                        runout = 0;
                                        wakeup(&runout);
            systm.h     _               char        runout;                 /* scheduling flag */
            text.c      xswap           if(runout) {
                                        runout = 0;
                                        wakeup(&runout);

runq        slp.c       sched           for(rp = runq; rp != NULL; rp = rp->p_link)
                        setrq           p->p_link = runq;
                                        runq = p;
                        swtch           for(p=runq; p!=NULL; p=p->p_link) {
                                        runq = p->p_link; else
            systm.h     _               int *runq;                  /* head of linked list of running processes */

runrun      clock.c     clock           runrun++;
            slp.c       setrun          runrun++;
                        swtch           runrun = 0;
            systm.h     _               char        runrun;                 /* scheduling flag */

rxreg       rx.c        NRXDEV          #define NRXDEV      (2*sizeof(rxreg)/sizeof(rxreg[0]))
                                        if(!(rxreg[minor(bp->b_dev)/2]->integ&RX2_RX02))
                        RX2_BC2         int *rxreg[] =
                        rx02wait        rx2cs = rxreg[minor(rxtab.b_actf->b_dev)/2];
                        rxdone          rx2cs = rxreg[minor(bp->b_dev)/2];
                        rxintr          rx2cs               = rxreg[minor(bp->b_dev)/2];
                        rxstart         rx2cs               = rxreg[minor(bp->b_dev)/2];
```

rxtab	conf.70.c	nodev	extern struct iobuf rxtab; &rxopen, &nulldev, &rxstrategy, &rxtab, /*rx*/

	rx.c	NRXDEV	struct iobuf rxtab;
			if(rxtab.b_actf == 0)
			rxtab.b_actf = bp;
			rxtab.b_actl->av_forw = bp;
			rxtab.b_actl = bp;
			if(rxtab.b_active == NO)

		rx02wait	rx2cs = rxreg[minor(rxtab.b_actf->b_dev)/2];

		rxdone	bp = rxtab.b_actf;
			rxtab.b_active = NO;
			rxtab.b_errcnt = MAXERRS+1;
			else if((dbaRX2_ERDEN) == 0 && rxtab.b_errcnt == 0)
			rxtab.b_errcnt = MAXERRS+1;
			if(rxtab.b_errcnt)
			if(rxtab.b_errcnt < MAXERRS)
			if(rxtab.b_errcnt)
			rxtab.b_errcnt = 0;
			rxtab.b_actf = bp->av_forw;

		rxintr	if(rxtab.b_active == NO)
			bp = rxtab.b_actf;
			rxtab.b_errcnt++;
			rxtab.b_errcnt = 0;

		rxstart	if((bp=rxtab.b_actf) == 0)
			rxtab.b_active = NO;
			rxtab.b_active = YES;

sabuf	bio.c	—	char sabuf[NBUF][BSIZE+BSLOP]; /* Addressable buffers */
		aagetblk	if (flag && bp->b_paddr >= (paddr_t)(unsigned)&sabuf[NBUF][0]){ if (bp->b_paddr < (paddr_t)(unsigned)&sabuf[NBUF][0])
		binit	bp->b_paddr = (paddr_t)sabuf[i];
		brelse	if (bp->b_paddr < (paddr_t)(unsigned)&sabuf[NBUF][0])
		getablk	if (bp->b_paddr < (paddr_t)(unsigned)&sabuf[NBUF][0])
		notavail	if (bp->b_paddr < (paddr_t)(unsigned)&sabuf[NBUF][0])
schans	mx1.c	—	struct schan schans[NPORTS];
		xcp	cp = (isport)? schans+i: chans+i;
sdzbrk	dz.c	—	char sdzbrk[(NDZ11+7)/8]; /* software copy of brk register */
		dzrstrt	sdzbrk[lineno>>3] =& ~(1<<(lineno&07));
			dzadr->dzbrk = sdzbrk[lineno>>3];
		dzxint	sdzbrk[dev.d_minor] =| (1<<lineno);
			dzadr->dzbrk = sdzbrk[dev.d_minor];
semi	sys5.c	—	int semi[NEVT] ;
		event	if (semi[s] < 0) {
			if (semi[s] > 0) {
			if(semi[s] != -u.u_procp->p_pid) {
			u.u_ar0[R0] = semi[s];
			if (++semi[s] < 0)
			semi[s] = 1;
			wakeup(&semi[s]);
			while (semi[s]==0) sleep(&semi[s],PEVT);
			u.u_ar0[R0] = semi[s];
			semi[s]--;

```
                                u.u_semav = semi[s];
                                sleep(&semi[s],PEVT);
                                } while(u.u_semav == semi[s]);
                                u.u_ar0[R0] = semi[s];
                                if (semi[s] != 0) semi[s]--;
                                sav = semi[s];
                                semi[s] = u.u_ar0[R0];
                                wakeup(&semi[s]);
                                while((u.u_ar0[R0] = semi[s]) != 0)
                                sleep(&semi[s], PEVT);
                                semi[s] = -u.u_procp->p_pid;
                                if((u.u_ar0[R0] = semi[s]) != 0)
                                u.u_ar0[R0] = semi[s];
                                semi[s] = 0;
                                wakeup(&semi[s]);
                                u.u_ar0[R0] = semi[s];
                semafree        if (semi[i] == pid) {
                                semi[i] = 0;
                                wakeup(&semi[i]);

slpque      slp.c       HASH    struct proc *slpque[SQSIZE];
                                rp->p_link = slpque[h];
                                slpque[h] = rp;
                                slpque[h] = rp->p_link;
                        wakeup  p = slpque[i];
                                slpque[i] = p->p_link;
                                p = slpque[i];

spyontp     tty.c       _       struct      tp      *spyontp, *spytp;
                        ttxint  if(tp == spyontp) {
                        ttyioctl if(spyontp) {
                                spyontp = tp;
                                spyontp = 0;

spytp       tty.c       _       struct      tp      *spyontp, *spytp;
                        ttxint  if(spytp->t_outq.c_cc >= TTYHOG)
                                flushtty(spytp);
                                putc(c, &spytp->t_outq);
                                ttstart(spytp);
                        ttyioctl spytp = u.u_ttyp;

suincnt     alloc.c     getfs   if (suincnt >= NBUF-NFBUF)
                                suincnt++;
                        iinit   suincnt++;
                        update  suincnt--;
                sys3.c  smount  suincnt++;
                                if (suincnt >= NBUF-NFBUF)
                        sumount suincnt--;
                systm.h _       char        suincnt;                /* number of superblocks currently in core */

swapdev     alloc.c     getfs   p->m_bufp = abread(swapdev, p->m_bufp);
                        iinit   (*bdevsw[swapdev.d_major].d_open)(swapdev, 1);
                                cp = agetblk(swapdev, cp);
                bio.c   swap    bp->b_dev = swapdev;
                                (*bdevsw[swapdev>>8].d_strategy)(bp);
                conf.70.c sizeof dev_t      swapdev = makedev(5, 39);
```

	sys1.c	exece	bp = getblk(swapdev, bno+(nc)>>BSHIFT));
			bp = bread(swapdev, bno+(nc)>>BSHIFT));
			bp = getblk(swapdev, bno+nc);
		exit	q = getblk(swapdev, a);
		freeproc	bp = abread(swapdev, f=p->p_addr);
	sys3.c	smount	smp->m_bufp = agetblk(swapdev, smp->m_bufp);
	systm.h	–	int swapdev; /* dev of swap see conf.c */
swapmap	alloc.c	iinit	cp = malloc(swapmap, 1);
	main.c	main	mfree(swapmap, nswap, swplo);
	slp.c	swapin	mfree(swapmap, ctod(p->p_size), p->p_addr);
	sys1.c	exece	if ((bno = malloc(swapmap,NCABLK)) == 0)
			mfree(swapmap, NCABLK, bno);
			mfree(swapmap, NCABLK, bno);
		exit	a = malloc(swapmap, 8);
		fork	if ((a = malloc(swapmap, ctod(MAXMEM))) == 0) {
			mfree(swapmap, ctod(MAXMEM), a);
		freeproc	mfree(swapmap, 8, f);
	sys3.c	smount	if((smp->m_bufp = malloc(swapmap, 1)) == NULL)
		sumount	mfree(swapmap, 1, mp->m_bufp);
	systm.h	–	struct map swapmap[SMAPSIZ]; /* space for swap allocation */
	text.c	xalloc	if((xp->x_daddr = malloc(swapmap, ctod(ts))) == NULL)
		xflsh	mfree(swapmap, ctod(xp->x_size), xp->x_daddr);
		xfree	mfree(swapmap, ctod(xp->x_size), xp->x_daddr);
		xswap	a = malloc(swapmap, ctod(rp->p_size));
swbuf1	bio.c	–	struct buf swbuf1;
		swap	bp = &swbuf1;
swbuf2	bio.c	–	struct buf swbuf2;
		swap	if((swbuf2.b_flags&B_WANTED) == 0)
			bp = &swbuf2;
swplo	conf.70.c	sizeof	daddr_t swplo 1;
	main.c	main	mfree(swapmap, nswap, swplo);
	systm.h	–	int swplo; /* block number of swap space */
sysent	sysent.c	nullsys	int sysent[]
	sysent.tu.c		int sysent[] =
	sysent.util.		int sysent[] =
	trap.c	–	struct sysent {
		int	} sysent[NSYSENT];
			struct sysent badent[1];
		trap	register struct sysent *callp;
			callp = &sysent[i];
			if (u.u_indflg = (callp == sysent)) { /* indirect */
			callp = &sysent[i];
sysprof	clock.c	clock	if (sysprof.pid)
		sincupc	register struct sysprof *tp;
			tp = &sysprof;
	sprof.h	–	struct sysprof {
	sprofx.h		extern struct sysprof sysprof;
	sys1.c	exit	if(u.u_procp->p_pid == sysprof.pid)
	syscb.c	–	struct sysprof sysprof = {NULL, NULL, 0,0,0,0,0};
		sprofil	register struct sysprof *tp;

			tp = &sysprof;
		sprofoff	sysprof.pid = 0;
tempq	tty.c	_	struct clist tempq;
		hqrelse	while((c = getc(&tempq)) >= 0)
		ttyoutpu	putc(c, &tempq);
termsw	conf.70.c	hp45inpu	struct termsw termsw[] =
		sizeof	int nttype = sizeof(termsw)/sizeof(struct termsw);
	conf.h	int	struct termsw {
	confx.h	_	struct termsw termsw[];
	tty.c	lsgtty	(*termsw[i].t_ioctl)(tp, LSET, 0);
			(*termsw[tp->t_term].t_ioctl)(tp, LRESET, i);
		ttyctl	(*termsw[tp->t_term].t_output)(c, tp);
		ttyinput	c = (*termsw[tp->t_term].t_input) (c, tp);
		ttyioctl	(*termsw[tp->t_term].t_ioctl)
text	sig.c	procxmt	register struct text *xp;
		ptrace	register struct text *xp;
	slp.c	shuffle	struct text *xp;
	syscb.c	textlock	struct text *xp;
		tunlock	struct text *xp;
	text.c	xalloc	register struct text *xp;
			for (xp = &text[0]; xp < &text[NTEXT]; xp++) {
		xccdec	struct text *xp;
			register struct text *xp;
		xexpand	struct text *xp;
			register struct text *xp;
		xflsh	register struct text *xp;
			for(xp=text; xp<&text[NTEXT]; xp++) {
		xfree	register struct text *xp;
		xlock	register struct text *xp;
		xswapin	struct text *xp;
			register struct text *xp;
		xunlock	register struct text *xp;
	text.h	_	struct text
	textx.h		struct text text[NTEXT];
time	acct.c	acct	acctbuf.ac_etime = time - u.u_start;
	alloc.c	iinit	time = cp->s_time;
		update	ip->s_time = time;
			iupdat(ip, &time);
	clock.c	clock	++time;
			if((time.loword&03) == 0)
	errlog.c	logovfl	if (type != lasttype || (time > (lasttime + 60))) {
			lasttime = time;
		logprdev	if (dev != lastdev || type != lasttype || (time > (lasttime + 60))) {
			lasttime = time;
		puterec	eup->e_hdr.e_time = time;
	iget.c	iput	iupdat(ip, &time);
		iupdat	xputl(addr, time);
	sys1.c	fork	u.u_start = time;
	sys3.c	stat1	iupdat(ip,&time);
	sys4.c	ftime	t.time = time;
			t.time++;
		gtime	u.u_ar0[R0] = time.hiword;

```
                                    u.u_ar0[R1] = time.loword;
                stime               time = ntime;
        systm.h —                   long        time;           /* time in sec from 1970 */
        timeb.h                     time_t      time;

tout    systm.h —                   long        tout;           /* time of day of next sleep */

ubmap   main.c  main                mfree(ubmap, 25, 6);
        rh.c    mapalloc            while((regno = malloc(ubmap, j)) == 0) {
                                    sleep(ubmap, PSWP);
                mapfree             mfree(ubmap, (bp->b_bcount-1)/8192+1, regno);
                                    wakeup(ubmap);
        systm.h —                   struct map ubmap[13];       /* Unibus Map allocation map */

updlock alloc.c update              if(updlock)
                                    updlock++;
                                    updlock = 0;
        systm.h —                   char        updlock;        /* lock for sync */

utsname conf.70.c sizeof            struct      utsname utsname = {
        utsname.h —                 struct utsname {
                                    extern struct utsname utsname;
        utssys.c utssys             if (copyout(&utsname, u.u_ar0[R0], sizeof(utsname)))

vp_addr vp.c    —                   struct device *vp_addr[NVP] = { 0177500 };
                vpclose             vpaddr = &(vp_addr[dev]->plcsr);
                vpioctl             vpaddr = &(vp_addr[dev]->plcsr);
                vpopen              vpaddr = &(vp_addr[dev]->plcsr);
                vpready             vpaddr = (vp->vp_state&PLOT)?&(vp_addr[dev]->plcsr):
                                    &(vp_addr[dev]->prcsr);
                vpwrite             vpaddr = vp_addr[dev];

vp_vp   vp.c    —                   struct vp vp_vp[NVP];
                vpclose             vp = &vp_vp[dev];
                vpintr              wakeup(&vp_vp[dev]);
                vpioctl             vp = &vp_vp[dev];
                vpopen              if (dev>=NVP || (vp = &vp_vp[dev])->vp_buf) {
                vpready             vp = &vp_vp[dev];
                vpwrite             vp = &vp_vp[dev];

vt11    jy.c    —                   } vt11;
                jyclose             vt11.vtjyfr.vtfradr = &vt11.vtlpfr.vtjump;
                jyopen              jyframe[JYLASTA] = &vt11.vtlpfr.vtjump;
                                    vt11.vtjyfr.vtfradr = jyframe;
        vt11.c  —                   struct      vtcycl vt11;            /* VT11 master frame loop */
                vtcopy              *to   = &vt11.vtfrm[frame+1];
                vtopen              vt11.vtstop = VTSTOP;
                                    vt11.vtsync = VTSTATSA | VTSYNC;
                                    vtfp = &vt11.vtjyfr;
                                    (--vtfp)->vtfradr = &vt11;
                                    vt11.vtstop = VTSTOP;
                                    VTADDR->vtdpc = &vt11;
                vtreplac            vtfrp = &vt11.vtfrm[frame];
                vtwrite             *stradr   = &vt11.vtfrm[frame+1];
                                    vt11.vtfrm[frame].vtfradr = vt11f.vtfrad;
        vtlp.c  —                   extern struct vtcycl vt11;
```

```
                    vtlpclos    vtll.vtlpfr.vtfradr = vtll.vtfrm;
                    vtlpopen    lpframe[LPLASTA] = vtll.vtfrm;
                                vtll.vtlpfr.vtfradr = lpframe;

vtllf       vtll.c  _           struct      vtfrpt vtllf;                   /* VTll frame replacement values */
                    vtwrite     frame  = vtllf.vtfrmn;
                                stradr = vtllf.vtfrad + vtllf.vtfrcn;
                                vtll.vtfrm[frame].vtfradr = vtllf.vtfrad;
            vtdbg.c _           extern struct vtfrpt vtllf;
                    vtdbgin     vtllf.vtfrmn = framen;
                                vtllf.vtfrad = sp;
                                vtllf.vtfrcn = (ep-sp) - 4;
            vtmon.c _           extern struct vtfrpt vtllf;
                    vtinit      vtllf.vtfrmn = vframen++;
                                vtllf.vtfrad = sp;
                                vtllf.vtfrcn = (ep-sp) - 4;
                                vtllf.vtfrmn = vframen++;
                                        vtllf.vtfrad = sp;
                                        vtllf.vtfrcn = (ep-sp) - 4;
                                vtllf.vtfrmn = vframen++;
                                vtllf.vtfrad = sp;
                                vtllf.vtfrcn = (ep-sp) - 4;

vtllh       vtll.c  _           struct      vtfrhd vtllh[VTNFRM];           /* VTll frame headers */
                    vtclose     vhp = vtllh;
                    vtnxpc      if(vfp=vtllh[frame].vfd_frm) {
                                vtllh[frame].vfd_frm = 0;
                    vtreplac    vtfrmfree(vtllh[frame].vfd_frm);
                                vtllh[frame].vfd_frm = frp;

vtinuse     vtll.c  _           int vtinuse;                        /* # of system bufs in use by VTll */
                    vtfree      vtinuse--;
                    vtnxpc      if(vtinuse++ == VTMXBFS) {
                                vtinuse--;

vttab       vtll.c  _           struct      iobuf vttab;                    /* VTll header */
                    vtclose     if(vttab.v_flags&VTUSER) {
                                vttab.v_flags =& ~(VTUSER|VTSYS);
                    vtfrmfre    vp = &vttab;
                    vtint       vp = &vttab;
                    vtnxpc      if(vttab.v_flags&VTBFREE) {
                                while(vttab.v_flags&VTBFREE)
                                sleep(&vttab.v_frmpc,VTBPRI);
                    vtopen      vp = &vttab;
                    vtread      sleep(&vttab,VTSPRI);
                    vtwrite     switch(vttab.v_flags&(VTUSER|VTSYS)) {

vtwait      vtll.c  _           int vtwait;                         /* # of cycles before user awake */
                    vtint       if(vtwait || (vp->v_flags&VTSTOP))
                    vtread      vtwait = u.u_count&077777;
                                while(vtwait--)
```

aabread	bio.c	aabread	aabread(dev, blkno, flag)
aagetblk	bio.c	aabread	bp = aagetblk(dev, blkno, flag);
		aagetblk	aagetblk(dev, blkno, flag)
access	fio.c	access	access(ip, mode)
	maus.c	maus	access(ip, IREAD);
			access(ip, IWRITE);
	nami.c	namei	access(dp, IEXEC);
			if(access(dp, IWRITE))
			if(access(dp, IWRITE))
	sig.c	core	if(!access(ip, IWRITE) &&
	sys1.c	gethead	if (access(ip, IEXEC) ||
	sys2.c	openi	access(rip, IREAD);
			access(rip, IWRITE);
		saccess	access(ip, IREAD);
			access(ip, IWRITE);
			access(ip, IEXEC);
	sys4.c	chdir	if(access(ip, IEXEC))
acct	acct.c	_	acct() { }
		acct	acct()
	sys1.c	exit	acct();
addch	mx1.c	addch	addch(ip, isport)
		mpxchan	if ((cp=addch(gip, 1))==NULL) {
			if((gfp->f_flag&FMP)==FMP)cp = addch(gip, 0);
	mx2.c	mxopen	if ((cp=addch(gp->g_inode,0)) == NULL) {
		xcp	struct chan *xcp(),*addch(),*nextcp();
alarm	alarm.c	_	struct alarm {
	sys4.c	alarm	alarm()
	sysent.c	nullsys	extern setuid(), getuid(), stime(), ptrace(), alarm(), fstat(), pause();
			0, &alarm, /* 27 = alarm */
	sysent.tu.c		extern setuid(), getuid(), stime(), ptrace(), alarm(), fstat(), pause();
			0, alarm, /* 27 = alarm */
	sysent.util.		extern setuid(), getuid(), stime(), ptrace(), alarm(), fstat(), pause();
			0, alarm, /* 27 = alarm */
alloc	alloc.c	alloc	alloc(dev)
	subr.c	bmap	if (rwflg==B_READ || (bp = alloc(d))==NULL)
			if (rwflg==B_READ || (bp = alloc(d))==NULL)
			if (rwflg==B_READ || (bp = alloc(d)) == NULL)
			if (rwflg==B_READ || (nbp = alloc(d)) == NULL) {
almclose	alarm.c	almclose	almclose(dev,flag)
almioctl	alarm.c	almioctl	almioctl(dev, cmd, addr, flg)
almopen	alarm.c	almopen	almopen(dev,flag)
aretu	mch.70.s	_aretu	_aretu:
	slp.c	HASH	aretu(u.u_qsav);
		swtch	aretu(u.u_qsav);

argadj	sys2.c	seek	vflag = argadj(UV_CBR2,1);
	sys4.c	chown	vflag = argadj(UV_CBR2,1);
		ulimit	if(argadj(UV_CBR2,2))
		utime	vflag = argadj(UV_CBR2,1);
	sys5.c	argadj	argadj(version,narg)
backup	mch.70.s	_backup	_backup:
	trap.c	trap	if(backup(u.u_ar0) == 0)
badblock	alloc.c	alloc	} while (badblock(fp, bno, dev));
		badblock	badblock(afp, abn, dev)
		free	if (badblock(fp, bno, dev))
bawrite	bio.c	bawrite	bawrite(bp)
		bdwrite	bawrite(bp);
	lfh.c	lfalloc	bawrite(bp);
		lfcopy	bawrite(bp);
		lffree	bawrite(bp);
		lfspace	bawrite(bp);
			bawrite(bp);
		putlfd	bawrite(bp);
	rdwri.c	writei	bawrite(bp);
bcopy	prf.c	int	bcopy(UISA, pwr_save.pf_uisa, cnt);
			bcopy(UISD, pwr_save.pf_uisd, cnt);
			bcopy(SDSA, pwr_save.pf_sdsa, sizeof(pwr_save.pf_sdsa));
			bcopy(SDSD, pwr_save.pf_sdsd, sizeof(pwr_save.pf_sdsd));
			bcopy(UBMAP, pwr_save.pf_ubmap, 124);
			bcopy(pwr_save.pf_uisa, UISA, cnt);
			bcopy(pwr_save.pf_uisd, UISD, cnt);
			bcopy(pwr_save.pf_sdsa, SDSA, sizeof(pwr_save.pf_sdsa));
			bcopy(pwr_save.pf_sdsd, SDSD, sizeof(pwr_save.pf_sdsd));
			bcopy(pwr_save.pf_ubmap, UBMAP, 124);
	subr.c	bcopy	bcopy(from, to, count)
bdwrite	alloc.c	update	bdwrite(mp->m_bufp);
	bio.c	bdwrite	bdwrite(bp)
	iget.c	iupdat	bdwrite(bp);
	rdwri.c	writei	bdwrite(bp);
	subr.c	bmap	bdwrite(bp);
			bdwrite(bp);
			bdwrite(nbp);
			bdwrite(bp);
	sys1.c	exece	bdwrite(bp);
			bdwrite(bp);
bflush	alloc.c	update	bflush(NODEV);
	bio.c	bflush	bflush(dev)
binit	bio.c	binit	binit()
	main.c	main	binit();
bisrch	clock.c	bisrch	bisrch(x,buf,siz)
		sincupc	t = bisrch((pc)>>1)&077777,tp->base->u_ct.ropt,tp->numcnts);
bmap	nami.c	namei	bmap(dp, (daddr_t)(u.u_offset>>BSHIFT), B_WRITE);

			bn = bmap(dp, (daddr_t)(u.u_offset>>BSHIFT), B_READ);
	rdwri.c	readi	bn = bmap(ip, lbn, B_READ);
		writei	if ((bn = bmap(ip, bn, B_WRITE)) == (daddr_t)-1
	subr.c	bmap	bmap(ip, bn, rwflg)
breada	bio.c	breada	breada(dev, blkno, rablkno)
	rdwri.c	readi	bp = breada(dev, bn, rablock);
brelse	alloc.c	alloc	brelse(bp);
		ialloc	brelse(bp);
		iinit	brelse(bp);
	bio.c	aagetblk	brelse(abp);
		bdwrite	brelse(bp);
		binit	brelse(bp);
		breada	brelse(rabp);
		brelse	brelse(bp)
		bwrite	brelse(bp);
		iodone	brelse(bp);
	iget.c	iget	brelse(ip);
			brelse(ip);
		itrunc	brelse(bp);
	lfh.c	getlfd	brelse(bp);
		lfalloc	brelse(bp);
	nami.c	namei	brelse(bp);
			brelse(bp);
			brelse(bp);
			brelse(bp);
	rdwri.c	readi	brelse(bp);
		writei	brelse(bp);
	subr.c	bmap	brelse(bp);
			brelse(bp);
	sys1.c	exece	brelse(bp);
			brelse(bp);
			brelse(bp);
			brelse(bp);
		freeproc	brelse(bp);
	sys3.c	smount	brelse(mp);
			brelse(mp);
		statl	brelse(bp);
		sumount	brelse(mp->m_bufp);
	vp.c	vpclose	brelse(vp->vp_buf);
	vtll.c	vtfree	brelse(vfp);
bwrite	alloc.c	free	bwrite(bp);
		update	bwrite(bp);
	bio.c	aagetblk	bwrite(bp);
		bawrite	bwrite(bp);
		bflush	bwrite(bp);
		bwrite	bwrite(bp)
		getablk	bwrite(bp);
	sys1.c	exit	bwrite(q);
canon	hf.c	hfread	while(tp->t_canq.c_cc || (tp->t_state&CARR_ON && canon(tp))) {
	tty.c	canon	canon(tp)
		ttread	if (tp->t_canq.c_cc || (tp->t_state&CARR_ON && canon(tp)))

```
cdelay      tty.c       cdelay      cdelay(colp)
                        ttyoutl     c = cdelay(colp);
                                    c = cdelay(colp);

challoc     mx1.c       addch       if ((cp=challoc(i, isport)) != NULL) {
                        xcp         challoc(index, isport)

chdir       sys4.c      chdir       chdir()
            sysent.c    nullsys     extern creat(), link(), unlink(), exec(), chdir(), gtime(), mknod(), chmod();
                                    1, &chdir,                      /* 12 = chdir */
            sysent.tu.c             extern creat(), link(), unlink(), exec(), chdir(), gtime(), mknod(), chmod();
                                    1, chdir,                       /* 12 = chdir */
            sysent.util.            extern creat(), link(), unlink(), exec(), chdir(), gtime(), mknod(), chmod();
                                    1, chdir,                       /* 12 = chdir */

chdrain     mx1.c       detach      chdrain(cp);
                                    chdrain(cp);
            mx2.c       chdrain     chdrain(cp)
                        mxclose     chdrain(cp);
                        mxopen      chdrain(cp);

checkur     main.c      checkur     checkur(nt, nd, ns, sep)
                        estabur     if(checkur(nt, nd, ns, sep))
            sys1.c      gethead     checkur(ts,ds,SSIZE+btoc(NCARGS-1),sep);

chfree      mx1.c       detach      chfree(cp);
                                    chfree(cp);
            mx2.c       chfree      chfree(cp)
                        mxclose     chfree(cp);
                        mxopen      chfree(cp);

chgrp       sys4.c      chgrp       chgrp()
                        chroot      chgrp();            /* CBR2 chgrp sys entry */

chmod       sys4.c      chmod       chmod()
            sysent.c    nullsys     extern creat(), link(), unlink(), exec(), chdir(), gtime(), mknod(), chmod();
                                    2, &chmod,                      /* 15 = chmod */
            sysent.tu.c             extern creat(), link(), unlink(), exec(), chdir(), gtime(), mknod(), chmod();
                                    2, chmod,                       /* 15 = chmod */
            sysent.util.            extern creat(), link(), unlink(), exec(), chdir(), gtime(), mknod(), chmod();
                                    2, chmod,                       /* 15 = chmod */

chown       sys4.c      chown       chown()
            sysent.c    nullsys     extern chown(), sbreak(), stat(), seek(), getpid(), smount(), sumount();
                                    3, &chown,                      /* 16 = chown */
            sysent.tu.c             extern chown(), sbreak(), stat(), seek(), getpid(), smount(), sumount();
                                    3, chown,                       /* 16 = chown */
            sysent.util.            extern chown(), sbreak(), stat(), seek(), getpid(), smount(), sumount();
                                    3, chown,                       /* 16 = chown */

chroot      sys4.c      chroot      chroot()
            sysent.c    nullsys     extern syscb(), chroot(), umask(), event();
                                    2, &chroot,                     /* 61 = chroot */
            sysent.tu.c             extern syscb(), chroot(), umask(), event();
                                    2, chroot,                      /* 61 = chroot */
            sysent.util.            extern syscb(), chroot(), umask(), event();
```

```
                                2, chroot,                      /* 61 = chroot */

chwake          mx1.c           detach          chwake(cp);
                mx2.c           chdrain         chwake(cp);
                                chwake          chwake(cp)

cinit           main.c          main            cinit();
                tty.c           cinit           cinit()

clear           alloc.c         alloc           clear(paddr(bp), BSIZE);
                mch.70.s        _clear          _clear:
                rdwri.c         readi           clear(paddr(bp), BSIZE);

clearse         mch.70.s        _clearse        _clearseg:

clearseg        main.c          main            clearseg(i);
                sig.c           grow            clearseg(--a);
                sys1.c          getxfile        clearseg(u.u_procp->p_addr+i);
                                sbreak          clearseg(--a);

clock           clock.c         clock           extern      int     clock();
                                                int (*clk_fn)()     &clock;
                                                clock(dev, sp, r1, nps, r0, pc, ps)

close           sys2.c          close           close()
                sysent.c        nullsys         extern rexit(), fork(), read(), write(), open(), close(), wait();
                                                0, &close,                      /*  6 = close */
                sysent.tu.c                     extern rexit(), fork(), read(), write(), open(), close(), wait();
                                                0, close,                       /*  6 = close */
                sysent.util.                    extern rexit(), fork(), read(), write(), open(), close(), wait();
                                                0, close,                       /*  6 = close */

closef          fio.c           closef          closef(fp)
                mx1.c           detach          closef(sub->g_file);
                                                closef(cp->c_fy);
                nmpipe.c        npclose         closef(rfp);
                                                closef(wfp);
                sys1.c          exit            closef(*a);
                                setregs         closef(u.u_ofile[i]);
                sys2.c          close           closef(fp);

copyiin         mch.70.s        _copyiin        _copyiin:

copyin          ioctl.c         ttioccom        if (copyin(addr, (caddr_t)&t, sizeof(t))) {
                                                if (copyin(addr, (caddr_t)&iocb, sizeof(iocb))) {
                                                if (copyin(addr, (caddr_t)&iooth, sizeof(iooth))) {
                lfh.c           lfioctl         if (copyin(addr, (caddr_t)&lfcb, sizeof(lfcb))) {
                mch.70.s        _copyin         _copyin:
                mx1.c           mpxchan         copyin(uap->argvec, &vec, sizeof vec);
                nmpipe.c        npioctl         if (copyin(addr, (caddr_t)&pipcb, sizeof(pipcb))) {
                sys4.c          utime           if(copyin((caddr_t)(u.u_arg[1]+4), (caddr_t) tbuf, 4)) {
                trans.c         trsioctl        if (copyin(addr, (caddr_t)&transcb, sizeof(transcb))) {
                tty.c           ttyioctl        if (copyin(addr, (caddr_t)&termblk, sizeof(termblk))) {
                vs.c            vsioctl         if (copyin(addr, (caddr_t)&vscb, sizeof(vscb))) {
                vt11.c          vtcopy          if(copyin(u.u_base,to,count) < 0)
```

```
copyio      alloc.c     alloc       copyio(paddr(bp)+2, fp->s_free, 200, U_RKD);
                        free        copyio(paddr(bp)+2, fp->s_free, 200, U_WKD);
                        iinit       copyio(paddr(bp), (caddr_t)cp->b_paddr, BSIZE, U_RKD);
                        update      copyio(paddr(bp), ip, BSIZE, U_WKD);
            bio.c       aagetblk    copyio(bp->b_paddr,(caddr_t)&bp->b_paddr,BSIZE,U_RKD);
            iget.c      iget        copyio((ip->b_paddr + 32*lron(ino+31, 16)), &p->i_mode,
                        iupdat      copyio(addr, &xp->i_mode, size, U_WKD);
            mch.70.s    _copyio     _copyio:
            mem.c       mmread      if (copyio(offset, u.u_base, n, U_RUD)) {
                        mmwrite     if (copyio(offset, u.u_base, n, U_WUD)) {
            messag.c    msghmov     copyio(paddr, mhd, sizeof(*mhd), mode);
                        msgmove     if (copyio(paddr,u.u_arg[1],len,mode))
            nami.c      namei       copyio(paddr(bp)+((unsigned)u.u_offset&BMASK), (caddr_t)&u.u_dent,
            rdwri.c     pimove      if (copyio(cp, u.u_base, n, (u.u_segflg<<1)|flag))
            rx.c        rxintr      copyio(bp->b_paddr+OFFSET, mx?_buf, bc, U_WKD);
                        rxstart     copyio(bp->b_paddr+OFFSET, mx?_buf, bc, U_RKD);
                                    copyio(bp->b_paddr+OFFSET, mx?_buf, bc, U_WKD);
            subr.c      bmap        copyio(paddr(bp), ip->i_un.i_addr, 16, U_WKD);
            sys1.c      exit        copyio(q->b_paddr, (caddr_t)&u, BSIZE, U_WKD);
            sys3.c      smount      copyio(mp->b_paddr, (caddr_t)mp->m_bufp->b_paddr, BSIZE, U_RKD);
                        stat1       copyio(paddr(bp) + 32*(int) (((long)ip->i_number + 31L) * 16) + 24,

copyiou     mch.70.s    _copyiou    _copyiout:

copyout     err.c       errread     if (copyout(eup, u.u_base, n))
            ioctl.c     ttioccom    if (copyout((caddr_t)&t, addr, sizeof(t)))
                                    if (copyout((caddr_t)&iocb, addr, sizeof(iocb)))
                                    if (copyout((caddr_t)&iooth, addr, sizeof(iooth)))
            jy.c        jyread      if (copyout(stake,ubase,6) == -1)
            lfh.c       lfioctl     if(copyout((caddr_t)&lfstat, addr, sizeof(lfstat)))
                        lfret       if(copyout((caddr_t)lfcb1,addr,sizeof(struct lfcb)))
            main.c      main        copyout(icode, 0, sizeof icode);
            mch.70.s    _copyout    _copyout:
            messag.c    messag      if (copyout((caddr_t)&mstat, u.u_arg[1],
            mx2.c       mxread      copyout(&h, base, sizeof h);
                        mxwrite     copyout(&h, hbase, sizeof h);
            nmpipe.c    npioctl     if (copyout((caddr_t)&pipcb, addr, sizeof(pipcb)))
            sys3.c      stat1                   if(copyout((caddr_t) &ds, ub, sizeof(ds))<0)
                                    if (copyout((caddr_t)&ods, ub, sizeof(ods))<0)
            sys4.c      ftime       if (copyout((caddr_t)&t, (caddr_t)uap->tp, sizeof(t)) < 0)
            syscb.c     syscb       else if(copyout(&u.lobyte+offset,u.u_arg[0],cnt) == -1)
            trans.c     trsioctl    if (copyout((caddr_t)&transcb, addr, sizeof(transcb)))
            tty.c       ttyioctl    if (copyout((caddr_t)&termblk, addr, sizeof(termblk)))
            utssys.c    utssys      if (copyout(&utsname, u.u_arg[0], sizeof(utsname)))
            vtlp.c      vtlpread    if(copyout(stake,ubase,6) == -1) {

copyseg     mch.70.s    _copyseg    _copyseg:
            sig.c       grow        copyseg(a-si, a);
            slp.c       expand      copyseg(a1+i, a2++);
                        newproc     copyseg(a1++, a2++);
            sys1.c      sbreak      copyseg(a-d, a);
                                    copyseg(a-d, a);

core        messag.c    msginit     register unsigned core;
                                    core = malloc(coremap, MSGMEM);
                                    if(core) {
```

```
                                        msgbase = core;
          sig.c         core            core()
                        psig            if(core())

cpass     dn.c          dnwrite         if((dnadr->dn_reg&PND) == 0 || u.u_count == 0 || (c=cpass()) < 0)
          lp.c          lpwrite         while ((c=cpass())>=0)
          lpm.c                         while ((c=cpass())>=0)
          subr.c        cpass           cpass()
          trans.c       trswrite        while((c = cpass()) >= 0) {
          tty.c         ttwrite         while((c = cpass()) >= 0) {
                                        switch(c = cpass()) {
                                        col = cpass();
                                        row = cpass();
          vs.c          vswrite         while((c = cpass()) >= 0) {

cpx       mx1.c         mpxchan         u.u_ar0[R0] = cpx(ngp);
                                        u.u_ar0[R0] = cpx(cp);
                                        u.u_ar0[R0] = cpx(cp);
                                        u.u_ar0[R0] = cpx(cp);
          mx2.c         cpx             cpx(cp)
                        mxread          h.index = cpx(cp);

creat     sys2.c        creat           creat()
          sysent.c      nullsys         extern creat(), link(), unlink(), exec(), chdir(), gtime(), mknod(), chmod();
                                        2, &creat,                      /*  8 = creat */
          sysent.tu.c                   extern creat(), link(), unlink(), exec(), chdir(), gtime(), mknod(), chmod();
                                        2, creat,                       /*  8 = creat */
          sysent.util.                  extern creat(), link(), unlink(), exec(), chdir(), gtime(), mknod(), chmod();
                                        2, creat,                       /*  8 = creat */

cvtdec    vt100.c       cvtdec          cvtdec(value, aqp)
                        vttrans         cvtdec(atp->t_row, qp);
                                        cvtdec(atp->t_lrow, qp);
                                        cvtdec(atp->t_vrow, qp);
                                        cvtdec(atp->t_col, qp);

delchar   trans.c       delchar         delchar(atp, ac)
                        trsinput        if(delchar(tp, c)) {
                        trsread         if(delchar(tp, c)) {

detach    mx1.c         detach          detach(cp)
                        mpxchan         detach(cp);
          mx2.c         mxclose         detach(cp);

devstart  devstart.c    devstart        devstart(bp, devloc, devblk, hbcom)
          rf.c          rfstart         devstart(bp, &RFADDR->rfda, bp->b_blkno<<8, 0);
          rk.c          rkstart         devstart(bp, &RKADDR->rkda, a, 0);
          rp.c          rpstart         devstart(bp, &RPADDR->rpca, bp->cylin, bp->b_dev.d_minor>>3);

dhclose   conf.70.c     _               &dhopen,    &dhclose,       &dhread,        &dhwrite,       /*dh*/
                        nodev           extern dhopen(), dhclose(), dhread(), dhwrite(), dhioctl(), dmcntrl(), dhil;
          dh.c          dhclose         dhclose(dev) {

dhcntrl   dh.c          dhcntrl         dhcntrl(dev, action) {
                        dhopen          dhcntrl(dev, 1);
```

```
dhioctl     conf.70.c   _           &dhioctl,   &dmcntrl,       &dhll,
                        nodev       extern dhopen(), dhclose(), dhread(), dhwrite(), dhioctl(), dmcntrl(), dhll;
            dh.c        dhioctl     dhioctl(dev, cmd, addr, flag)

dhopen      conf.70.c   _           &dhopen,    &dhclose,       &dhread,        &dhwrite,       /*dh*/
                        nodev       extern dhopen(), dhclose(), dhread(), dhwrite(), dhioctl(), dmcntrl(), dhll;
            dh.c        dhopen      dhopen(dev, flag) {

dhparam     dh.c        dhioctl     dhparam(dev);
                        dhopen      dhparam(dev);
                        dhparam     dhparam(dev) {

dhread      conf.70.c   _           &dhopen,    &dhclose,       &dhread,        &dhwrite,       /*dh*/
                        nodev       extern dhopen(), dhclose(), dhread(), dhwrite(), dhioctl(), dmcntrl(), dhll;
            dh.c        dhread      dhread(dev) {

dhrint      dh.c        dhrint      dhrint() {
                        dhscan      dhrint();

dhrstrt     dh.c        dhrstrt     dhrstrt(atp)  struct tty *atp;  {
                        dhstart     extern dhrstrt();
                                          timeout(dhrstrt,tp,(cnt&0177)+chrdelay[tp->t_speeds.hibyte&0171]);

dhscan      dh.c        dhcntrl     extern dhstart(),dhscan();
                                    timeout(&dhscan,0,DHSCNPAT);
                        dhscan      dhscan() {
                                    timeout(&dhscan, 0, DHSCNPAT);

dhstart     dh.c        dhcntrl     extern dhstart(),dhscan();
                                    tp->t_addr  = dhstart;
                        dhstart     dhstart(atp)  struct tty *atp;  {
                        dhxint      dhstart(tp);
                        dhxoff      dhstart(tp);

dhwrite     conf.70.c   _           &dhopen,    &dhclose,       &dhread,        &dhwrite,       /*dh*/
                        nodev       extern dhopen(), dhclose(), dhread(), dhwrite(), dhioctl(), dmcntrl(), dhll;
            dh.c        dhwrite     dhwrite(dev) {

dhxint      dh.c        dhopen      dhxint(line);
                        dhxint      dhxint(dev) {

dhxoff      dh.c        dhrint      dhxoff(tp, c&0177);
                        dhxoff      dhxoff(tp, c)

display     clock.c     clock       display();
            mch.70.s    _display    _display:

djclose     dj.c        djclose     djclose(dev)

djcntrl     dj.c        djcntrl     djcntrl(dev, action)
                        djopen      djcntrl(dev, 1);

djioctl     dj.c        djioctl     djioctl(dev, cmd, addr, flag)

djopen      dj.c        djopen      djopen(dev, flag)
```

```
djread      dj.c        djread      djread(dev)

djrint      dj.c        djrint      djrint()
                        djscan      djrint();

djscan      dj.c        djcntrl     extern djscan();
                                    timeout(djscan, 0, 4);
                        djscan      djscan()
                                    timeout(djscan, 0, 4);

djstart     dj.c        djcntrl     extern djstart();
                                    tp->t_addr = djstart;
                        djstart     djstart(atp)

djwrite     dj.c        djwrite     djwrite(dev)

djxint      dj.c        djxint      djxint(dev)

dmcboot     dmc11.c     dmcboot     dmcboot(dev)
                        dmcioctl    if(dmcboot(dev) == FALSE) {

dmcbufq     dmc11.c     dmcboot     bp  = dmcbufq(dev,dmcmopd[dev.d_minor],sizeof(dmcmopd[0]),B_READ);
                                    abp = dmcbufq(dev,dmcmop,sizeof(dmcmop),B_WRITE);
                        dmcbufq     dmcbufq(dev,bufa,nbytes,flag)
                        dmcioctl    bp = dmcbufq(dev,dmcmopd[dev.d_minor],1,B_WRITE);

dmcclose    dmc11.c     dmcclose    dmcclose(dev)

dmcclr      dmc11.c     dmcclose    dmcclr(dev);
                        dmcclr      dmcclr(dev)
                        dmcinit     dmcclr(dev);

dmccmd      dmc11.c     dmcboot     dmccmd(CNTLI,bp,&dmc11[dev.d_minor]);
                        dmccmd      dmccmd(type,bp,dmp)
                        dmciint     if(dmccmd(type,bp,dmp) == FALSE)
                        dmcstart    if(dmccmd(BACCI,bp,dmp) == FALSE)

dmciint     dmc11.c     dmciint     dmciint(dev)

dmcinit     dmc11.c     dmciint     dmcinit(dev);
                        dmcinit     dmcinit(dev)
                        dmcioctl    int dmcinit();
                                    dmcinit(dev);
                                    dmcinit(dev);
                                    dmcinit(dev);
                                    timeout(dmcinit,dev,DELAY);
                        dmcoint     dmcinit(dev,dmp);
                        dmcopen     dmcinit(dev);

dmcioctl    dmc11.c     dmcioctl    dmcioctl(dev, cmd, addr, flag)

dmcload     dmc11.c     dmccmd      if(dmcload(dmp,cmd,base,count) == TRUE) {
                        dmcinit     if(dmcload(dmp,BASEI,dmcarea[dev.d_minor],0) == FALSE)
                                    if(dmcload(dmp,CNTLI,0,0) == FALSE)
                        dmcload     dmcload(dmp,type,w0,w1)
```

```
dmcntrl     conf.70.c   _           &dhioctl,   &dmcntrl,       &dhll,
                        nodev       extern dhopen(), dhclose(), dhread(), dhwrite(), dhioctl(), dmcntrl(), dhll;
            dhdm.c      dmcntrl     dmcntrl(tp, flag, bits)

dmcoint     dmc11.c     dmcoint     dmcoint(dev)

dmcopen     dmc11.c     dmcopen     dmcopen(dev, flag)

dmcread     dmc11.c     dmcread     dmcread(dev)

dmcstart    dmc11.c     dmcbufq     dmcstart(dmp);
                        dmciint     dmcstart(dmp);
                        dmcoint     dmcstart(dmp);
                        dmcstart    dmcstart(dmp)
                        dmcstrategy dmcstart(dmp);

dmcstrategy dmc11.c     dmcboot     physio(&dmcstrategy,dev,B_WRITE,0);
                        dmcread     physio(dmcstrategy,dev,B_READ,0);
                        dmcstrategy dmcstrategy(bp)
                        dmcwrite    physio(dmcstrategy,dev,B_WRITE,0);

dmcwrite    dmc11.c     dmcwrite    dmcwrite(dev)

dmint       dhdm.c      dmint       dmint(dev) {

dnclose     conf.70.c   _           &dnopen,    &dnclose,       &nodev,         &dnwrite,       /*dn*/
                        nodev       extern dnopen(), dnclose(), dnwrite(), dnint();
            dn.c        dnclose     dnclose(dev)

dnint       conf.70.c   nodev       extern dnopen(), dnclose(), dnwrite(), dnint();
            dn.c        dnint       dnint(dev)

dnopen      conf.70.c   _           &dnopen,    &dnclose,       &nodev,         &dnwrite,       /*dn*/
                        nodev       extern dnopen(), dnclose(), dnwrite(), dnint();
            dn.c        dnopen      dnopen(dev, flag)

dnwrite     conf.70.c   _           &dnopen,    &dnclose,       &nodev,         &dnwrite,       /*dn*/
                        nodev       extern dnopen(), dnclose(), dnwrite(), dnint();
            dn.c        dnwrite     dnwrite(dev)

dpcmp       mch.70.s    _dpcmp      _dpcmp:

ds40input   ds40.c      ds40input   ds40input(ac, atp)

ds40ioctl   ds40.c      ds40ioctl   ds40ioctl(tp, flag, nvrow)

ds40lca     ds40.c      ds40lca     ds40lca(row, col, atp)
                        ds40output  ds40lca(atp->t_row,atp->t_col,atp);

ds40output  ds40.c      ds40lca     ds40output(HOME,atp);
                                    while(rowmoves--) ds40output(rowdir,atp);
                                    while(aseq--)     ds40output(ASEC,atp);
                                    ds40output(CRTN,atp);
                                    while(colmoves--) ds40output(coldir,atp);
                        ds40output  ds40output(ac, atp)
                                    ds40output(DC,atp);
```

```
                                ds40output(HOME,atp);

dup         sys3.c      dup     dup()
            sysent.c    nullsys extern ulimit(), setpgrp(), tell(), dup(), pipe(), times(), profil();
                                0, &dup,                    /* 41 = dup */
            sysent.tu.c         extern ulimit(), setpgrp(), tell(), dup(), pipe(), times(), profil();
                                0, dup,                     /* 41 = dup */
            sysent.util.        extern ulimit(), setpgrp(), tell(), dup(), pipe(), times(), profil();
                                0, dup,                     /* 41 = dup */

dzclose     conf.70.c   -       &dzopen,    &dzclose,       &dzread,        &dzwrite,       /*dz*/
                        nodev   extern dzopen(), dzclose(), dzread(), dzwrite(), dzioctl(), dz11;
            dz.c        dzclose dzclose(dev)

dzcntrl     dz.c        dzcntrl dzcntrl(dev,action)
                        dzopen  dzcntrl(dev,1);

dzioctl     conf.70.c   -       &dzioctl,   &fmcntrl,       &dz11,
                        nodev   extern dzopen(), dzclose(), dzread(), dzwrite(), dzioctl(), dz11;
            dz.c        dzioctl dzioctl(dev, cmd, addr, flag)

dzmcntrl    dz.c        dzclose dzmcntrl(dev,'c',CDLEAD);
                        dzcntrl dzmcntrl(dev,'s',CDLEAD);
                                dzmcntrl(dev,'c',CDLEAD);
                                if(dzmcntrl(dev, 't', CARRIER))
                        dzmcntrl dzmcntrl(dev,flag,bits)

dzopen      conf.70.c   -       &dzopen,    &dzclose,       &dzread,        &dzwrite,       /*dz*/
                        nodev   extern dzopen(), dzclose(), dzread(), dzwrite(), dzioctl(), dz11;
            dz.c        dzopen  dzopen(dev,flag)

dzparam     dz.c        dzioctl dzparam(dev);
                        dzopen  dzparam(dev);
                        dzparam dzparam(dev)

dzread      conf.70.c   -       &dzopen,    &dzclose,       &dzread,        &dzwrite,       /*dz*/
                        nodev   extern dzopen(), dzclose(), dzread(), dzwrite(), dzioctl(), dz11;
            dz.c        dzread  dzread(dev)

dzrint      dz.c        dzrint  dzrint()
                        dzscan  dzrint();

dzrstrt     dz.c        dzrstrt dzrstrt(atp)
                        dzxint  extern dzrstrt();
                                timeout(dzrstrt,tp,(c & 0177)+chrdelay[tp->t_speeds

dzstart     dz.c        dzcntrl extern dzstart(),dzscan();
                                tp->t_addr = dzstart;
                        dzstart dzstart(atp)

dzwrite     conf.70.c   -       &dzopen,    &dzclose,       &dzread,        &dzwrite,       /*dz*/
                        nodev   extern dzopen(), dzclose(), dzread(), dzwrite(), dzioctl(), dz11;
            dz.c        dzwrite dzwrite(dev)

dzxint      dz.c        dzopen  dzxint(line);
                        dzxint  dzxint(dev)
```

dzxoff	dz.c	dzrint	dzxoff(tp, c&0177);
		dzxoff	dzxoff(tp, c)
errclose	conf.70.c	—	/*20*/ &erropen, &errclose, &errread, &nodev, /*er*/
		nodev	extern erropen(), errclose(), errread();
	err.c	errclose	errclose(dev,flg)
errinit	errlog.c	errinit	errinit()
	errlogf.c	—	errinit() {}
	main.c	main	errinit();
erropen	conf.70.c	—	/*20*/ &erropen, &errclose, &errread, &nodev, /*er*/
		nodev	extern erropen(), errclose(), errread();
	err.c	erropen	erropen(dev,flg)
errread	conf.70.c	—	/*20*/ &erropen, &errclose, &errread, &nodev, /*er*/
		nodev	extern erropen(), errclose(), errread();
	err.c	errread	errread(dev)
estabur	main.c	estabur	estabur(nt, nd, ns, sep, xrw)
		main	estabur(0, 1, 0, 0, RO);
	sig.c	core	estabur((unsigned)0, s, (unsigned)0, 0, RO);
		grow	if(estabur(u.u_tsize, u.u_dsize, u.u_ssize+si, u.u_sep, RO))
		procxmt	estabur(u.u_tsize, u.u_dsize, u.u_ssize, u.u_sep, RW);
			estabur(u.u_tsize, u.u_dsize, u.u_ssize, u.u_sep, RO);
	sys1.c	getxfile	estabur((unsigned)0, u.u_dsize, (unsigned)0, 0, RO);
			estabur(u.u_tsize, u.u_dsize, u.u_ssize,u.u_sep,RO);
		sbreak	if(estabur(u.u_tsize, u.u_dsize+d, u.u_ssize, u.u_sep, RO))
	text.c	xalloc	estabur(ts, 0, 0, 0, RW);
event	mx2.c	scontrol	scontrol(cp,event,value)
			short event,value;
			putw(event,q);
	sys5.c	event	event()
	sysent.c	nullsys	extern syscb(), chroot(), umask(), event();
			2, &event /* 63 = event */
	sysent.tu.c		extern syscb(), chroot(), umask(), event();
			2, event /* 63 = event */
	sysent.util.		extern syscb(), chroot(), umask(), event();
			2, event /* 63 = event */
exec	sys1.c	NCABLK	exec(ip)
	sysent.c	nullsys	extern creat(), link(), unlink(), exec(), chdir(), gtime(), mknod(), chmod();
			2, &exec, /* 11 = exec */
	sysent.tu.c		extern creat(), link(), unlink(), exec(), chdir(), gtime(), mknod(), chmod();
			2, exec, /* 11 = exec */
	sysent.util.		extern creat(), link(), unlink(), exec(), chdir(), gtime(), mknod(), chmod();
			2, exec, /* 11 = exec */
exece	sys1.c	NCABLK	exece(ip);
		exece	exece()
	sysent.c	nullsys	extern exece(); /* 59 = exece */
			3, &exece,
	sysent.tu.c		extern exece(); /* 59 = exece */
			3, exece,

```
            sysent.util.            extern exece();
                                    3, exece,                   /* 59 = exece */

exit        sig.c       procxmt     exit();
                        psig        exit();
                        stop        exit();
            sys1.c      exit        exit()
                        rexit       exit();

expand      main.c      main        expand(USIZE+1);
            sig.c       grow        expand(u.u_procp->p_size+si);
            slp.c       expand      expand(newsize)
            sys1.c      getxfile    expand(i);
                        sbreak      expand(i);
                                    expand(n);

falloc      fio.c       falloc      falloc()
            mx1.c       mpxchan     if ((fp=falloc()) == NULL) {
                                    if ((fp=falloc()) == NULL)
            pipe.c      pipe        rf = falloc();
                                    wf = falloc();
            sys2.c      open1       if ((fp = falloc()) == NULL)

fcntl       sys3.c      fcntl       fcntl()
            sysent.c    nullsys     extern fcntl();
                                    2, &fcntl,                  /* 62 = fcntl   */
            sysent.tu.c             extern fcntl();
                                    2, fcntl,                   /* 62 = fcntl   */
            sysent.util.            extern fcntl();
                                    2, fcntl,                   /* 62 = fcntl   */

flush       mx2.c       chdrain     flush(&cp->cx.datq);
                                    flush(&cp->cy.datq);
                        chwake      flush(&cp->c_ctlx);
                        flush       flush(q)
                        wflush      flush(q);

flushtty    hf.c        hfclose     flushtty(tp);
                        hfioctl     flushtty(tp);
            trans.c     trsinput    flushtty(tp);
                        trsioctl    flushtty(tp);
                                    flushtty(tp);
                                    flushtty(tp);
                                    flushtty(tp);
            tty.c       flushtty    flushtty(tp)
                        ttxint      flushtty(spytp);
                        ttydst      flushtty(tp);
                        ttyinput    flushtty(tp);
                                    flushtty(tp);
                        ttyioctl    flushtty(tp);
                        wflushtt    flushtty(tp);
            vs.c        vsdst       flushtty(tp);
                        vsinput     flushtty(tp);
                        vsioctl     flushtty(tp);

fmcntrl     conf.70.c   -           &klioctl,   &fmcntrl,       &kl11,
```

```
                                        &dzioctl,   &fmcntrl,       &dz11,
                    nodev               extern fmcntrl();   /* fake modem control routine */
          dhfdm.c   fmcntrl             fmcntrl(tp, action)

fmtberr   errlog.c  fmtberr             fmtberr(dp,cyl)
          errlogf.c _                   fmtberr() {}
          hps.c     hpintr              fmtberr(dp,hp_sizes[spart(bp->b_dev)].cyloff);
                    hpustart            fmtberr(dp,(bp==NULL)?0:hp_sizes[spart(bp->b_dev)].cyloff);
          hs.c      hsintr              fmtberr(&hstab,0);
          ht.c      htintr              fmtberr(&httab,0);
                                        fmtberr(&httab,0);
          rf.c      rfintr              fmtberr(&rftab,0);
          rk.c      rkintr              fmtberr(&rktab,0);
          rp.c      rpintr              fmtberr(&rptab,rp_sizes[minor(bp->b_dev)&07].cyloff);
          tm.c      tmstart             fmtberr(&tmtab,0);
                                        fmtberr(&tmtab,0);

fork      sys1.c    fork                fork()
          sysent.c  nullsys             extern rexit(), fork(), read(), write(), open(), close(), wait();
                                        0, &fork,                   /*  2 = fork */
          sysent.tu.c                   extern rexit(), fork(), read(), write(), open(), close(), wait();
                                        0, fork,                    /*  2 = fork */
          sysent.util.                  extern rexit(), fork(), read(), write(), open(), close(), wait();
                                        0, fork,                    /*  2 = fork */

free      alloc.c   free                free(dev, bno)
          iget.c    itrunc              free(rp->i_dev, *cp);
                                        free(rp->i_dev, *ip);

freeproc  sig.c     issig               freeproc(q, 0);
          sys1.c    freeproc            freeproc(p, flg)
                    wait                freeproc(p, 1);

freeslot  err.c     errread             freeslot(eup);
          errlog.c  freeslot            freeslot(eup)

fsig      sig.c     fsig                fsig(p)
                    issig               n = fsig(p);
                    psig                n = fsig(rp);
          sys1.c    wait                u.u_ar0[R1] = (fsig(p)<<8) | 0177;
          vtmon.c   vtprocen            vtvconv(fsig(pmp),pcp->pr_sig,2,10);
                                        vtvconv(fsig(pmp),pcp->pr_sig,2,10);

fstat     sys3.c    fstat               fstat()
          sysent.c  nullsys             extern setuid(), getuid(), stime(), ptrace(), alarm(), fstat(), pause();
                                        1, &fstat,                   /* 28 = fstat */
          sysent.tu.c                   extern setuid(), getuid(), stime(), ptrace(), alarm(), fstat(), pause();
                                        1, fstat,                    /* 28 = fstat */
          sysent.util.                  extern setuid(), getuid(), stime(), ptrace(), alarm(), fstat(), pause();
                                        1, fstat,                    /* 28 = fstat */

ftime     sys4.c    ftime               ftime()
          sysent.c  nullsys             extern ftime();
                                        1, &ftime,                   /* 35 = ftime; formerly sleep */
          sysent.tu.c                   extern ftime();
                                        1, ftime,                    /* 35 = ftime; formerly sleep */
```

	sysent.util.		extern ftime(); /* 35 = ftime; formerly sleep */ 1, ftime,
fubyte	mch.70.s	_fubyte	_fubyte:
	nami.c	uchar	c = fubyte(u.u_dirp++);
	sig.c	procxmt	if (fubyte(ipc.ip_addr) == -1)
	subr.c	cpass	if((c = id==0?fubyte(u.u_base):fuibyte(u.u_base)) < 0) {
	sysl.c	exece	if ((c = fubyte((caddr_t)ap++)) < 0)
fuibyte	mch.70.s	_fuibyte	_fuibyte:
	sig.c	procxmt	if (fuibyte(ipc.ip_addr) == -1)
	subr.c	cpass	if((c = id==0?fubyte(u.u_base):fuibyte(u.u_base)) < 0) {
fuiword	main.c	main	if(fuiword(lks) == -1) {
			if(fuiword(lks) == -1)
	mch.70.s	_fuiword	_fuiword:
	sig.c	procxmt	ipc.ip_data = fuiword(ipc.ip_addr);
	trap.c	trap	if(fuiword(pc-2) == SETD && u.u_signal[SIGINS] == 0)
			if((i=fuiword(pc-2)&0377) < NSYSENT)
			a = fuiword(pc);
			u.u_arg[i] = fuiword(pc);
fuword	mch.70.s	_fuword	_fuword:
	rx.c	rxioctl	if((den=fuword(addr)) != 1 && den != 2)
	sig.c	procxmt	ipc.ip_data = fuword(ipc.ip_addr);
	sysl.c	exece	ap = fuword((caddr_t)uap->argp);
			if ((ap = fuword((caddr_t)uap->envp)) == NULL)
	trap.c	trap	i = fuword(a);
			u.u_arg[i] = fuword(a += 2);
	tty.c	gtty	v[0] = fuword(up);
			v[1] = fuword(up+1);
			v[2] = fuword(up+2);
		stty	u.u_arg[0] = fuword(up);
			u.u_arg[1] = fuword(++up);
			u.u_arg[2] = fuword(++up);
	vp.c	vpioctl	m = fuword(addr);
getablk	bio.c	aagetblk	abp = getablk(1);
		getablk	getablk(flag)
	rdwri.c	readi	bp = getablk(0);
	vp.c	vpopen	vp->vp_buf = getablk(1);
	vtll.c	vtnxpc	vfp = getablk();
getbfh	bio.c	getbfh	getbfh()
		physio	bp = getbfh();
	dmcll.c	dmcboot	bp = getbfh();
		dmcbufq	bp = getbfh();
	lfh.c	lfio	bp = getbfh();
	rx.c	rxioctl	bp = getbfh();
		rxopen	bp = getbfh();
getc	hf.c	hfread	if((c1=getc(&tp->t_canq)) == QESC) break;
			if((c1=getc(&tp->t_canq)) == QESC)
		hfxint	c = getc(&tp->t_outq);
			switch(c = getc(&tp->t_outq)) {
	lp.c	lpstart	while (LPADDR->lpsr&DONE && (c = getc(&lp11)) >= 0)

```
            lpm.c                   while (lprtab[lpr].addr->lpsr&DONE && (c = getc(lp)) >= 0)
            mch.70.s    _getc       _getc:
            mx2.c       flush       getc(q);
                        q_to_b      if ((c = getc(q)) == -1)
            subr.c      getw        s = getc(p);
                                    return(s | (getc(p)<<8));
            trans.c     trsread     while((c=getc(&tp->ts_rawq)) != -1) {
                        trsxint     if((c = getc(&tp->ts_outq)) < 0)
            tty.c       canon       while ((c=getc(&tp->t_rawq)) >= 0) {
                        flushtty    while (getc(&tp->t_canq) >= 0);
                                    while (getc(&tp->t_outq) >= 0);
                                    while (getc(&tp->t_rawq) >= 0);
                        hqrelse     while((c = getc(&tempq)) >= 0)
                        ttread      while (tp->t_canq.c_cc && passc(getc(&tp->t_canq))>=0);
                        ttxint      c = getc(&tp->t_outq);
                                    switch(c = getc(&tp->t_outq)) {
            vs.c        vsread      while((c=getc(&tp->t_rawq)) != -1) {
                        vsxint      if((c = getc(&tp->t_outq)) < 0)

geterec     err.c       errread     err_t       *geterec();
                                    eup = geterec();
            errlog.c    geterec     geterec()

geterror    bio.c       bwrite      geterror(bp);
                        geterror    geterror(bp)
                        iowait      geterror(bp);
                        physio      geterror(bp);
            lfh.c       lfio        geterror(bp);

geteslot    errlog.c    fmtberr     if((eup = geteslot(sizeof(struct eblock)+(dp->io_nreg*sizeof(int)))) == NULL) {
                        geteslot    geteslot(size)
                        logovfl     if ((eup = geteslot(sizeof(struct eovfl))) == NULL)
                        logparity   if((eup = geteslot(sizeof(struct eparity))) != NULL) {
                        logpower    if ((eup = geteslot(sizeof(struct epower))) != NULL)
                        logprdev    if ((eup = geteslot(sizeof(struct eprdev))) == NULL)
                        logstart    if ((eup = geteslot(sizeof(struct estart))) == NULL)
                        logstray    if((eup = geteslot(sizeof(struct estray))) != NULL) {
                        logtchg     if((eup = geteslot(sizeof(struct etimchg))) != NULL) {

getf        fio.c       getf        getf(f)
            ioctl.c     OTHERBIT    if ((fp = getf(uap->fdes)) == NULL)
            mx1.c       mpxchan     gfp = getf(vec.m_arg[1]);
                                    if ((fp=getf(vec.m_arg[0]))==NULL)
                                    if ((fp=getf(vec.m_arg[0]))==NULL)
                                    if ((chfp=getf(vec.m_arg[1]))==NULL)
            mx2.c       mxioctl     if ((gp=getmpx(dev))==NULL || (fp=getf(u.u_arg[0]))==NULL) {
                        mxread      if ((FP=getf(u.u_ar0[R0])) == NULL) {
                        mxwrite     if ((FP=getf(u.u_ar0[R0])) == NULL) {
            nmpipe.c    npopen      extern struct file *getf();
                                    npp->p_rfp  = getf(u.u_ar0[R0]);
                                    (npp->p_wfp = getf(u.u_ar0[R1]))->f_flag =| FNPIPE;
                                    getf(u.u_ar0[R0])->f_flag =| FNPIPE;
            sys2.c      close       fp = getf(u.u_ar0[R0]);
                        rdwr        fp = getf(u.u_ar0[R0]);
                        seek        fp = getf(u.u_ar0[R0]);
                        tell        if (fp = getf(u.u_ar0[R0])) {
```

```
                sys3.c          dup             fp = getf(u.u_ar0[R0].lobyte);
                                fcntl           fp = getf(u.u_ar0[R0]);
                                fstat           fp = getf(u.u_ar0[R0]);
                tty.c           sgtty           if ((fp = getf(u.u_ar0[R0])) == NULL)

getfs           alloc.c         alloc           fp = getfs(dev);
                                free            fp = getfs(dev);
                                getfs           getfs(dev)
                                ialloc          fp = getfs(dev);
                                ifree           fp = getfs(dev);

getgid          sys4.c          getgid          getgid()
                sysent.c        nullsys         extern setgid(), getgid(), ssig(), messag(), sysacct();
                                                0, &getgid,                     /* 47 = getgid */
                sysent.tu.c                     extern setgid(), getgid(), ssig(), messag(), sysacct();
                                                0, getgid,                      /* 47 = getgid */
                sysent.util.                    extern setgid(), getgid(), ssig(), messag(), sysacct();
                                                0, getgid,                      /* 47 = getgid */

gethead         sys1.c          exece           if((ip = gethead()) == NULL)
                                gethead         gethead()

getmdev         sys3.c          getmdev         getmdev()
                                smount          d = getmdev();
                                sumount         d = getmdev();

getmpx          mx2.c           mxclose         if ((gp=getmpx(dev)) == NULL)
                                mxioctl         if ((gp=getmpx(dev))==NULL || (fp=getf(u.u_arg[0]))==NULL) {
                                mxopen          if ((gp=getmpx(dev)) == NULL) {
                                mxread          if ((gp=getmpx(dev)) == NULL) {
                                mxwrite         if ((gp=getmpx(dev)) == NULL) {
                                xcp             getmpx(dev)

getpid          sys4.c          getpid          getpid()            /* get own and parent process id */
                sysent.c        nullsys         extern chown(), sbreak(), stat(), seek(), getpid(), smount(), sumount();
                                                0, &getpid,                     /* 20 = getpid */
                sysent.tu.c                     extern chown(), sbreak(), stat(), seek(), getpid(), smount(), sumount();
                                                0, getpid,                      /* 20 = getpid */
                sysent.util.                    extern chown(), sbreak(), stat(), seek(), getpid(), smount(), sumount();
                                                0, getpid,                      /* 20 = getpid */

getuid          sys4.c          getuid          getuid()
                sysent.c        nullsys         extern setuid(), getuid(), stime(), ptrace(), alarm(), fstat(), pause();
                                                0, &getuid,                     /* 24 = getuid */
                sysent.tu.c                     extern setuid(), getuid(), stime(), ptrace(), alarm(), fstat(), pause();
                                                0, getuid,                      /* 24 = getuid */
                sysent.util.                    extern setuid(), getuid(), stime(), ptrace(), alarm(), fstat(), pause();
                                                0, getuid,                      /* 24 = getuid */

getw            subr.c          getw            getw(p)

getxfile        sys1.c          exece           getxfile(ip, nc+2*na);
                                getxfile        getxfile(ip, nargc)

gpalloc         mx1.c           gpalloc         gpalloc()
                                mpxchan         if ((i=gpalloc()) < 0) {
```

```
grow        sig.c       grow        grow(sp)
                        psig        grow(n);
            trap.c      trap        if(grow(a))

gtime       sys4.c      gtime       gtime()
            sysent.c    nullsys     extern creat(), link(), unlink(), exec(), chdir(), gtime(), mknod(), chmod();
                                    0, &gtime,                  /* 13 = time */
            sysent.tu.c             extern creat(), link(), unlink(), exec(), chdir(), gtime(), mknod(), chmod();
                                    0, gtime,                   /* 13 = time */
            sysent.util.            extern creat(), link(), unlink(), exec(), chdir(), gtime(), mknod(), chmod();
                                    0, gtime,                   /* 13 = time */

gtty        sysent.c    nullsys     extern utime(), stty(), gtty(), saccess(), nice(), sync(), kill();
                                    1, &gtty,                   /* 32 = gtty */
            sysent.tu.c             extern utime(), stty(), gtty(), saccess(), nice(), sync(), kill();
                                    1, gtty,                    /* 32 = gtty */
            sysent.util.            extern utime(), stty(), gtty(), saccess(), nice(), sync(), kill();
                                    1, gtty,                    /* 32 = gtty */
            tty.c       gtty        gtty()

hcommand    ht.c        hcommand    hcommand(unit, com)
                        htclose     hcommand(unit, WEOF);
                                    hcommand(unit, WEOF);
                                    hcommand(unit, REW);
                                    hcommand(unit, SREV);
                                    hcommand(unit, NOP);
                        tabinit     ds = hcommand(unit, NOP);

hfclose     hf.c        hfclose     hfclose(dev,tp)

hfdst       hf.c        hfdst       hfdst(atp,acsr,alsr)  struct tty *atp;  {

hfioctl     hf.c        hfioctl     hfioctl(dev, cmd, addr, flag)

hfread      hf.c        hfread      hfread(atp)  struct tty *atp;  {

hfrint      hf.c        hfrint      hfrint(achar,atp)  struct tty *atp;  {

hftrnrd     hf.c        hftrnrd     hftrnrd(tp)
                        hfxint      extern hftrnrd(),ttrstrt();
                                    timeout(hftrnrd,tp,PAUSE);

hfwrite     hf.c        hfwrite     hfwrite(atp)  struct tty *atp;  {

hfxint      hf.c        hfxint      hfxint(atp, flag)  struct tty *atp;  {

hinit       bio.c       binit       hinit();
                        hinit       hinit()

hp45input   conf.70.c   hp45input   extern hp45input(), hp45output(), hp45ioctl();
                                    &hp45input, &hp45output,    &hp45ioctl,     /*HP 45*/
            hp45.c                  hp45input(ac, atp)

hp45ioctl   conf.70.c   hp45input   extern hp45input(), hp45output(), hp45ioctl();
                                    &hp45input, &hp45output,    &hp45ioctl,     /*HP 45*/
```

```
                hp45.c          hp45ioctl       hp45ioctl(tp, flag, nvrow)

hp45output      conf.70.c       hp45input       extern hp45input(), hp45output(), hp45ioctl();
                                                &hp45input, &hp45output,    &hp45ioctl,     /*HP 45*/
                hp45.c          hp45output      hp45output(ac, atp)

hpintr          hps.c           hpintr          hpintr()
                low.70.s        hpio            hpio:       jsr     r0,call; jmp _hpintr

                                                &hpopen,    &nulldev,       &hpread,        &hpwrite,       /*hp*/
hpopen          conf.70.c       -               extern hpopen(), hpread(), hpwrite(), hpstrategy();
                                nodev           &hpopen,    &nulldev,       &hpstrategy,    &hptab,         /*hp*/
                hps.c           tabinit         hpopen(dev, flag)

                                                &hpopen,    &nulldev,       &hpread,        &hpwrite,       /*hp*/
hpread          conf.70.c       -               extern hpopen(), hpread(), hpwrite(), hpstrategy();
                                nodev
                hps.c           hpread          hpread(dev)

hpstart         hps.c           hpintr          hpstart();
                                hpstart         hpstart()
                                hpustart        hpstart();

                                                extern hpopen(), hpread(), hpwrite(), hpstrategy();
hpstrategy      conf.70.c       nodev           &hpopen,    &nulldev,       &hpstrategy,    &hptab,         /*hp*/
                hps.c           hpread          physio(hpstrategy, dev, B_READ, nblks);
                                hpstrategy      hpstrategy(abp)
                                hpwrite         physio(hpstrategy, dev, B_WRITE, nblks);

hpustart        hps.c           hpintr          hpustart(unit);
                                                hpustart(unit);
                                                hpustart(cnt);
                                hpstrategy      hpustart(unit);
                                hpustart        hpustart(unit)
                                tabinit         hpustart(unit);

                                                &hpopen,    &nulldev,       &hpread,        &hpwrite,       /*hp*/
hpwrite         conf.70.c       -               extern hpopen(), hpread(), hpwrite(), hpstrategy();
                                nodev
                hps.c           hpwrite         hpwrite(dev)

hqrelse         tty.c           hqrelse         hqrelse(tp)
                                ttwrite         hqrelse(tp);
                                                hqrelse(tp);

hrelse          bio.c           hrelse          hrelse(bp)
                                physio          hrelse(bp);
                dmc11.c         dmcboot         hrelse(bp);
                                                hrelse(bp);
                                                hrelse(bp);
                                                hrelse(abp);
                                dmcioctl        hrelse(bp);
                lfh.c           lfio            hrelse(bp);
                rx.c            rxioctl         hrelse(bp);
                                rxopen          hrelse(bp);

hsintr          hs.c            hsintr          hsintr()
```

```
hsopen      hs.c        tabinit     hsopen(dev, flag)

hsread      hs.c        hsread      hsread(dev)

hsstart     hs.c        hsintr      hsstart();
                        hsstart();
                        hsstart();
                        hsstart     hsstart()
                        hsstrategy  hsstart();
                        tabinit     hsstart();

hsstrategy  hs.c        hsread      physio(hsstrategy, dev, B_READ,
                        hsstrategy  hsstrategy(abp)
                        hswrite     physio(hsstrategy, dev, B_WRITE,

hswrite     hs.c        hswrite     hswrite(dev)

htclose     conf.70.c   _           &htopen,    &htclose,       &htread,        &htwrite,       /*ht*/
                        nodev       extern htopen(), htclose(), htread(), htwrite(), htstrategy();
                                    &htopen,    &htclose,       &htstrategy,    &httab,         /*ht*/
            ht.c        htclose     htclose(dev, flag)

htintr      ht.c        htintr      htintr()
            low.70.s    htio        htio:       jsr     r0,call; jmp _htintr

htopen      conf.70.c   _           &htopen,    &htclose,       &htread,        &htwrite,       /*ht*/
                        nodev       extern htopen(), htclose(), htread(), htwrite(), htstrategy();
                                    &htopen,    &htclose,       &htstrategy,    &httab,         /*ht*/
            ht.c        tabinit     htopen(dev, flag)

htphys      ht.c        htphys      htphys(dev)
                        htread      htphys(dev);
                        htwrite     htphys(dev);

htread      conf.70.c   _           &htopen,    &htclose,       &htread,        &htwrite,       /*ht*/
                        nodev       extern htopen(), htclose(), htread(), htwrite(), htstrategy();
            ht.c        htread      htread(dev)

htstart     ht.c        htintr      htstart();
                        htstart     htstart()
                        htstrategy  htstart();
                        tabinit     htstart();

htstrategy  conf.70.c   nodev       extern htopen(), htclose(), htread(), htwrite(), htstrategy();
                                    &htopen,    &htclose,       &htstrategy,    &httab,         /*ht*/
            ht.c        hcommand    htstrategy(bp);
                        htread      physio(htstrategy, dev, B_READ, 0);
                        htstrategy  htstrategy(abp)
                        htwrite     physio(htstrategy, dev, B_WRITE, 0);

htwrite     conf.70.c   _           &htopen,    &htclose,       &htread,        &htwrite,       /*ht*/
                        nodev       extern htopen(), htclose(), htread(), htwrite(), htstrategy();
            ht.c        htwrite     htwrite(dev)

ialloc      alloc.c     ialloc      ialloc(dev)
            iget.c      maknode     ip = ialloc(u.u_pdir->i_dev);
```

```
            mx1.c       mpxchan     if ((ip=ialloc(rootdev))==NULL)
            pipe.c      pipe        ip = ialloc(rootdev);

idle        clock.c     _           extern char idle;
            mch.70.s    _idle       _idle:
            prf.c       panic       idle();
            slp.c       swtch       idle();

ifree       alloc.c     ifree       ifree(dev, ino)
            iget.c      iput        ifree(ip->i_dev, ip->i_number);

iget        alloc.c     ialloc      ip = iget(dev, ino);
            iget.c      iget        iget(dev, ino)
            main.c      main        rootdir = iget(rootdev, ROOTINO);
            nami.c      namei       iget(dp->i_dev, dp->i_number);
                                    dp = iget(d, u.u_dent.u_ino);
            sys4.c      unlink      ip = iget(pp->i_dev, u.u_dent.u_ino);

iinit       alloc.c     iinit       iinit()
            main.c      main        iinit();

incore      bio.c       breada      if (!incore(dev, blkno)) {
                                    if (rablkno && bfreelist.b_bcount>rathresh && !incore(dev, rablkno)) {
                        incore      incore(dev, blkno)

incupc      clock.c     clock       incupc(pc, u.u_prof);
            mch.70.s    _incupc     _incupc:

ioctl       ioctl.c     OTHERBIT    ioctl()
            sysent.c    nullsys     extern ioctl();
                                    3, &ioctl,                  /* 54 = ioctl */
            sysent.tu.c             extern ioctl();
                                    3, ioctl,                   /* 54 = ioctl */
            sysent.util.            extern ioctl();
                                    3, ioctl,                   /* 54 = ioctl */
            tty.c       gtty        ioctl();
                        stty        ioctl();

iodone      bio.c       iodone      iodone(bp)
            dmc11.c     dmcclr      iodone(bp);
                                    iodone(bp);
                        dmccmd      iodone(bp);
                        dmcoint     iodone(bp);
                        dmcstrategy iodone(bp);
            hps.c       hpintr      iodone(bp);
                        hpstrategy  iodone(bp);
                        hpustart    iodone(bp);
            hs.c        hsintr      iodone(bp);
                        hsstrategy  iodone(bp);
            ht.c        htintr      iodone(bp);
                        htstart     iodone(bp);
                        htstrategy  iodone(bp);
            rf.c        rfintr      iodone(bp);
                        rfstrate    iodone(bp);
            rk.c        rkintr      iodone(bp);
                        rkstrate    iodone(bp);
```

```
            rp.c        rpintr      iodone(bp);
                        rpstrate    iodone(bp);
            rx.c        NRXDEV      iodone(bp);
                        rxdone      iodone(bp);
            tm.c        tmstart     iodone(bp);
                                    iodone(bp);
                                    iodone(bp);
                        tmstrate    iodone(bp);

iomove      mx2.c       iomove      iomove(adr, cnt, flg)
                        mxmove      iomove(buf, cc, dir);
                        mxread      iomove(m_eot, sizeof m_eot, B_READ);
                        mxwcontr    iomove(cmd, sizeof cmd, B_WRITE);
                        mxwrite     iomove(&h, sizeof h, B_WRITE);

iowait      bio.c       aabread     iowait(bp);
                        breada      iowait(bp);
                        bwrite      iowait(bp);
                        iowait      iowait(bp)
            ht.c        hcommand    iowait(bp);
            rx.c        rxioctl     iowait(bp);
                        rxopen      iowait(bp);
            tm.c        tcommand    iowait(bp);

iput        acct.c      sysacct     iput(acctp);
                                    iput(ip);
            alloc.c     ialloc      iput(ip);
                        update      iput(ip);
            fio.c       closef      iput(ip);
                        owner       iput(ip);
            iget.c      iget        iput(p);
                        iput        iput(ip)
                        maknode     iput(u.u_pdir);
                        wdir        iput(u.u_pdir);
            maus.c      maus        iput(ip);
            mx1.c       mpxchan     iput(ip);
                                    iput(ip);
                                    iput(ip);
            mx2.c       mxclose     iput(ip);
            nami.c      namei       iput(dp);
                                    iput(dp);
                                    iput(dp);
                                    iput(dp);
            pipe.c      pipe        iput(ip);
                                    iput(ip);
            sig.c       core        iput(ip);
            sys1.c      exece       iput(ip);
                                    iput(ip);
                        exit        iput(u.u_cdir);
                                    iput(u.u_rdir);
                        gethead     iput(ip);
            sys2.c      link        iput(ip);
                                    iput(u.u_pdir);
                                    iput(u.u_pdir);
                                    iput(ip);
                        mknod       iput(ip);
```

		open1	iput(rip);
		saccess	iput(ip);
	sys3.c	getmdev	iput(ip);
		smount	iput(ip);
		stat	iput(ip);
		sumount	iput(ip);
	sys4.c	chdir	iput(ip); iput(u.u_cdir);
		chgrp	iput(ip);
		chmod	iput(ip);
		chown	iput(ip);
		chroot	iput(ip); iput(oldroot);
		unlink	iput(ip); iput(pp);
		utime	iput(ip);
	text.c	xflsh	iput(ip);
		xfree	iput(ip);
issig	clock.c	clock	if(issig())
	sig.c	issig	issig()
	slp.c	HASH	if(issig()) { if(pri > PZERO && issig())
	trans.c	trswrite	if (issig()) {
	trap.c	trap	if(issig())
	tty.c	ttwrite	if (hqflag==0 && issig()) {
itrunc	iget.c	iput	itrunc(ip);
		itrunc	itrunc(ip)
	pipe.c	readp	itrunc(ip);
	sig.c	core	itrunc(ip);
	sys2.c	open1	itrunc(rip);
iupdat	alloc.c	update	iupdat(ip, &time);
	iget.c	iput	iupdat(ip, &time);
		iupdat	iupdat(p, tm)
	sys3.c	stat1	iupdat(ip,&time);
	sys4.c	utime	iupdat(ip, tbuf);
jyclose	jy.c	jyclose	jyclose()
jyint	jy.c	jyint	jyint()
jyopen	jy.c	jyopen	jyopen(dev)
jypoll	jy.c	jyopen	int jypoll(); timeout(jypoll,0,2);
		jypoll	jypoll() timeout(jypoll,0,2);
jyread	jy.c	jyread	jyread()
kill	sys4.c	kill	kill()
	sysent.c	nullsys	extern utime(), stty(), gtty(), saccess(), nice(), sync(), kill(); 1, &kill, /* 37 = kill */
	sysent.tu.c		extern utime(), stty(), gtty(), saccess(), nice(), sync(), kill();

```
                                1, kill,                        /* 37 = kill */
        sysent.util.            extern utime(), stty(), gtty(), saccess(), nice(), sync(), kill();
                                1, kill,                        /* 37 = kill */

kiltout     clock.c     kiltout     kiltout(afun,aarg)
            hps.c       tabinit     kiltout(hp_pwrup, 0);
            hs.c                    kiltout(hs_pwrup, 0);
            trans.c     trsread     kiltout(&trswake, tp);

klclose     conf.70.c   _           /* 0*/      &klopen,        &klclose,       &klread,        &klwrite,       /*kl*/
                        nodev       extern klopen(), klclose(), klread(), klwrite(), klioctl(), kl11;
            kl.c        klclose     klclose(dev)

klcntrl     kl.c        klcntrl     klcntrl(dev, action)
                        klopen      klcntrl(dev, 1);

klioctl     conf.70.c   _           &klioctl,   &fmcntrl,       &kl11,
                        nodev       extern klopen(), klclose(), klread(), klwrite(), klioctl(), kl11;
            kl.c        klioctl     klioctl(dev, cmd, addr, flag)

klopen      conf.70.c   _           /* 0*/      &klopen,        &klclose,       &klread,        &klwrite,       /*kl*/
                        nodev       extern klopen(), klclose(), klread(), klwrite(), klioctl(), kl11;
            kl.c        klopen      klopen(dev, flag)

klread      conf.70.c   _           /* 0*/      &klopen,        &klclose,       &klread,        &klwrite,       /*kl*/
                        nodev       extern klopen(), klclose(), klread(), klwrite(), klioctl(), kl11;
            kl.c        klread      klread(dev)

klrint      kl.c        klrint      klrint(dev)
            low.70.s    klin        klin:       jsr     r0,call; jmp _klrint

klwrite     conf.70.c   _           /* 0*/      &klopen,        &klclose,       &klread,        &klwrite,       /*kl*/
                        nodev       extern klopen(), klclose(), klread(), klwrite(), klioctl(), kl11;
            kl.c        klwrite     klwrite(dev)

klxint      kl.c        klopen      extern klxint();
                                    pwr_init(kl11, NKL11+NDL11, klxint);
                        klxint      klxint(dev)
            low.70.s    klou        klou:       jsr     r0,call; jmp _klxint

ldiv        iget.c      iget        ip = bread(dev, ldiv(ino+31,16));
                        iupdat      bp = bread(rp->i_dev, ldiv(i,16));
            mch.70.s    _ldiv       _ldiv:
            nami.c      namei       u.u_count = ldiv(dp->i_size1, DIRSIZ+2);
            prf.c       printn      if(a = ldiv(n, b))
            rk.c        rkaddr      b = ldiv(b, m);
            rp.c        rpstrate    p1 = ldiv(p1, 10);
            sig.c       grow        si = ldiv(-sp, 64) - u.u_ssize + SINCR;

lfclose     lfh.c       lfclose     lfclose(dev,flag)

lfioctl     lfh.c       lfioctl     lfioctl(dev, cmd, addr, flag)

lfopen      lfh.c       sizeof      lfopen(dev,flag)

link        sys2.c      link        link()
```

```
        sysent.c    nullsys     extern creat(), link(), unlink(), exec(), chdir(), gtime(), mknod(), chmod();
                                2, &link,                   /*  9 = link */
        sysent.tu.c             extern creat(), link(), unlink(), exec(), chdir(), gtime(), mknod(), chmod();
                                2, link,                    /*  9 = link */
        sysent.util.            extern creat(), link(), unlink(), exec(), chdir(), gtime(), mknod(), chmod();
                                2, link,                    /*  9 = link */

lock    syscb.c     lock        lock()
                    syscb       lock();

logberr errlog.c    logberr     logberr(dp,err)
        errlogf.c   _           logberr() {}
        hps.c       hpintr      logberr(dp,bp->b_flags&B_ERROR);
                    hpustart    logberr(dp,B_ERROR);
        hs.c        hsintr      logberr(&hstab, bp->b_flags&B_ERROR);
        ht.c        htintr      logberr(&httab, bp->b_flags&B_ERROR);
        rf.c        rfintr      logberr(&rftab,bp->b_flags&B_ERROR);
        rk.c        rkintr      logberr(&rktab,bp->b_flags&B_ERROR);
        rp.c        rpintr      logberr(&rptab,bp->b_flags&B_ERROR);
        tm.c        tmstart     logberr(&tmtab,bp->b_flags&B_ERROR);

logovfl errlog.c    logovfl     logovfl(type)
        errlogf.c   _           logovfl() {}
        fio.c       falloc      logovfl(E_FILEO);
        iget.c      iget        logovfl(E_INODEO);
        sys1.c      fork        logovfl(E_PROCO);
        text.c      xalloc      logovfl(E_TEXTO);

logparity errlog.c  logparity   logparity(addr)
        errlogf.c   _           logparity() {}
        trap.c      trap        logparity(MEMORY);

logpower err.c      erropen     logpower();
        errlog.c    logpower    logpower()
        errlogf.c   _           logpower() {}

logprdev errlog.c   logprdev    logprdev(type, dev)
        errlogf.c   _           logprdev() {}
        prf.c       prdev       logprdev(type, dev);

logstart err.c      erropen     logstart();
        errlog.c    logstart    logstart()
        errlogf.c   _           logstart() {}

logstray errlog.c   logstray    logstray(addr)
        errlogf.c   _           logstray() {}
        hs.c        hsintr      logstray(HSADDR);
        rf.c        rfintr      logstray(RFADDR);
        tm.c        tmstart     logstray(TMADDR);
        trap.c      stray       logstray(addr);

logtchg errlog.c    logtchg     logtchg(nt)
        errlogf.c   _           logtchg() {}
        sys4.c      stime       logtchg(nt.me);

lpcanon lp.c        lpcanon     lpcanon(c)
```

```
                                        lpcanon(c2);
                    lpclose             lpcanon(FORM);
                    lpopenl             lpcanon(FORM4);
                    lpwrite             lpcanon(c);
          lpm.c     lpcanon             lpcanon(lpr, c)
                                        lpcanon(lpr, c2);
                    lpclose             lpcanon(lpr, FORM);
                    lpcpenl             lpcanon(lpr, FORM);
                    lpwrite             lpcanon(dev.d_minor, c);

lpclose   lp.c      lpclose             lpclose(dev, flag)
          lpm.c                         lpclose(dev, flag)

lpint     lp.c      lpint               lpint()
                    lpopenl             lpint();
          lpm.c     lpint               lpint(lpr)
                    lpopenl             lpint(lpr);

lpopen    lp.c      lpopen              lpopen(dev, flag)

lpopenl   lp.c      lpopen              lpopenl(dev, flag);
                    lpopenl             lpopenl(dev, flag)
          lpm.c                         lpopenl(dev, flag)

lpoutput  lp.c      lpcanon             lpoutput(c1);
                                        lpoutput('\r');
                                        lpoutput(' ');
                                        lpoutput(c1);
                    lpoutput            lpoutput(c)
          lpm.c     lpcanon             lpoutput(lpr, c1);
                                        lpoutput(lpr, '\r');
                                        lpoutput(lpr, ' ');
                                        lpoutput(lpr, c1);
                    lpoutput            lpoutput(lpr, c)

lprstrt   lp.c      lprstrt             lprstrt()
                    lpstart             int lprstrt();
                                        timeout(lprstrt,0,LPDELAY);
          lpm.c     lprstrt             lprstrt(lpr)
                    lpstart             int lprstrt();
                                        timeout(lprstrt,lpr,LPDELAY);

lpstart   lp.c      lpint               lpstart();
                    lpoutput            lpstart();
                    lprstrt             lpstart();
                    lpstart             lpstart()
          lpm.c     lpint               lpstart(lpr);
                    lpoutput            lpstart(lpr);
                    lprstrt             lpstart(lpr);
                    lpstart             lpstart(lpr)

lpwrite   lp.c      lpwrite             lpwrite()
          lpm.c                         lpwrite(dev)

lrem      iget.c    iget                copyio((ip->b_paddr + 32*lrem(ino+31, 16)), &p->i_mode,
                    iupdat              addr = paddr(bp) + 32*lrem(i, 16);
```

```
            mch.70.s    _lrem       _lrem:
            prf.c       printn      putchar(lrem(n, b) + '0');
            rk.c        rkaddr      d = lrem(b, m);
            rp.c        rpstrate    p2 = lrem(p1, 10);

lsgtty      dh.c        dhioctl     flg = lsgtty(addr, tp);
            dj.c        djioctl     lsgtty(addr, tp);
            dz.c        dzioctl     flg = lsgtty(addr, tp);
            kl.c        klioctl     lsgtty(addr, tp);
            tty.c       lsgtty      lsgtty(av, tp)

main        main.c      main        main()
            mch.70.s    start           jsr     pc,_main

maknode     iget.c      maknode     maknode(mode)
            mx1.c       mpxchan     ip = maknode((vec.m_arg[0]&0777)+IFMPC);
            sig.c       core        ip = maknode(0666);
            sys2.c      creat       ip = maknode(u.u_arg[1]&07777&(~ISVTX));
                        mknod       ip = maknode(u.u_arg[1]);

malloc      alloc.c     iinit       cp = malloc(swapmap, 1);
            errlog.c    geteslot    n = malloc(err.e_map,ns);
            lfh.c       lfalloc     if((start=malloc(bfp,size)) == 0) {
            main.c      main        if ((bufbase = malloc(coremap, btoc(BSIZE)*NXBUF)) == NULL)
                                            if((cpool = malloc(coremap,NPOOL)) == NULL)
                                    if (mauscore = malloc(coremap, ssize)) {
                                    if((xclistbase = malloc(coremap, i)) == 0)
                                    ssize = malloc(coremap, 0);
            malloc.c    malloc      malloc(mp, size)
            messag.c    messag      if((bp = malloc(msgmap, (n+movrhead)>>6)) == NULL) {
                        msginit     core = malloc(coremap, MSGMEM);
            rh.c        mapalloc    while((regno = malloc(ubmap, j)) == 0) {
            slp.c       expand      a2 = malloc(coremap, newsize);
                        newproc     a2 = malloc(coremap, n);
                        swapin      if ((a = malloc(coremap, p->p_size)) == NULL)
            sys1.c      exece       if ((bno = malloc(swapmap,NCMBLK)) == 0)
                        exit        a = malloc(swapmap, 8);
                        fork        if ((a = malloc(swapmap, ctod(MTXMEM))) == 0) {
            sys3.c      smount      if((smp->m_bufp = malloc(swapmap, 1)) == NULL)
            text.c      xalloc      if((xp->x_daddr = malloc(swapmap, ctod(ts))) == NULL)
                        xexpand     if ((xp->x_caddr = malloc(coremap, xp->x_size)) != NULL) {
                        xswap       a = malloc(swapmap, ctod(rp->p_size));
                        xswapin     if ((x = malloc(coremap, xp->x_size)) == NULL) {

mapalloc    dmc11.c     dmccmd      mapalloc(bp);
            rf.c        rfstrate    mapalloc(bp);
            rh.c        mapalloc    mapalloc(bp)
            rhf.c                   mapalloc()
            rk.c        rkstrate    mapalloc(bp);
            rp.c        rpstart     mapalloc(bp);
            tm.c        tmstrate    mapalloc(bp);

mapfree     bio.c       iodone      mapfree(bp);
            rh.c        mapfree     mapfree(bp)
            rhf.c                   mapfree()
```

```
maus        maus.c      maus        maus()
            sysent.c    nullsys     extern maus();
                                    1, &maus,                   /* 58 = set up MAUS segment reg */

max         rdwri.c     max         max(a, b)
            sys3.c      stat1       max (0, PIPSIZ - ip->i_size1);
            tty.c       cdelay      return(max((*colp>>4) + 3, 12));
            vtmon.c     vttinc      register char max;
                                    max = (hd_fram.hd_time[i-1] == ':') ? '5' : '9' ;
                                    if(++hd_fram.hd_time[i] <= max)

mcread      conf.70.c   ttread      extern mcread(), mcwrite();
                                    &mcread,    &nulldev,       &mcwrite,       &nulldev,
            mx1.c       mpxchan     extern mxopen(), mcread(), uchar();
                                    if (linesw[i].l_read==mcread) {
            mx2.c       mcread      mcread(cp)
                        mxread      more = mcread(cp);

mcstart     fakemx.c    _           mcstart() {}
            mx2.c       mcstart     mcstart(cp, q)
                        mxrstrt     mcstart(cp, q);
            trans.c     trsxint     mcstart(tp->ts_chan, (caddr_t)&tp->ts_outq);
            tty.c       ttxint      mcstart(tp->t_chan, (caddr_t)&tp->t_outq);

mcwrite     conf.70.c   ttread      extern mcread(), mcwrite();
                                    &mcread,    &nulldev,       &mcwrite,       &nulldev,
            mx2.c       mcwrite     mcwrite(cp)

mdown       mx1.c       mdown       mdown(sub,master)
                                    if((depth=mdown(sub->g_chans[i],master)) == -1)
                        mtree       if ((stresiz=mdown(sub,master)) <= 0) {

messag      messag.c    messag      messag()
            messagf.c               messag()
            sysent.c    nullsys     extern setgid(), getgid(), ssig(), messag(), sysacct();
                                    4, &messag,                 /* 49 = message */
            sysent.tu.c             extern setgid(), getgid(), ssig(), messag(), sysacct();
            sysent.util.            extern setgid(), getgid(), ssig(), messag(), sysacct();

mfree       errlog.c    errinit     mfree(err.e_map,err.e_nslot,1);
                        freeslot    mfree(err.e_map,ns,(((struct errslot *)eup)-err.e_slot)+1);
            lfh.c       lffree      mfree(bfp, size, start);
            main.c      main        mfree(coremap, 1, i);
                                    mfree(ubmap, 25, 6);
                                    mfree(swapmap, nswap, swplo);
            malloc.c    mfree       mfree(mp, size, a)
            messag.c    msgfree     mfree(msgmap, (size+movrhead)>>6, hp);
                        msginit     mfree(msgmap, MSGSEM, 1);
            rh.c        mapfree     mfree(ubmap, (bp->b_bcount-1)/8192+1, regno);
            slp.c       expand      mfree(coremap, n-newsize, a1+newsize);
                                    mfree(coremap, n, a1);
                        swapin      mfree(coremap,p->p_size,a);
                                    mfree(swapmap, ctod(p->p_size), p->p_addr);
            sys1.c      exece       mfree(swapmap, MCBLK, bno);
                                    mfree(swapmap, MCBLK, bno);
                        exit        mfree(coremap, p->p_size, p->p_addr);
```

```
                    fork        mfree(swapmap, ctod(MAXMEM), a);
                    freeproc    mfree(swapmap, 8, f);
          sys3.c    sumount     mfree(swapmap, 1, mp->m_bufp);
          text.c    xccdec      mfree(coremap, xp->x_size, xp->x_caddr);
                    xflsh       mfree(swapmap, ctod(xp->x_size), xp->x_daddr);
                    xfree       mfree(swapmap, ctod(xp->x_size), xp->x_daddr);
                                mfree(coremap, xp->x_size, xp->x_caddr);
                    xswap       mfree(coremap, os, xp->p_addr);

min       err.c     errread     n = min(eup->e_hdr.e_len, u.u_count);
          main.c    main        maxmem = min(maxmem, MAXMEM);
          maus.c    maus        int min;
                                min = (minor(ip->i_un.i_rdev) - 8) & 0377;
                                if(major(ip->i_un.i_rdev) != MEMDEV || min >= nmausent)
                                u.u_msav[j].ms_uisd = i | ACS | (mausmap[min].bsize-1)<<8;
                                u.u_msav[j].ms_uisa = mausmap[min].boffset + mauscore;
          mem.c     mmoff       *cnt = min(u.u_count, bn);
          messag.c  msgrecv     n = min(mhd->mq_size, u.u_ar0[R0]);
          pipe.c    writep      u.u_count = min(c, PIPSIZ);
          rdwri.c   min         min(a, b)
                    readi       n = min((unsigned)BSIZE-on, u.u_count);
                    writei      n = min((unsigned)BSIZE-on, u.u_count);
          rx.c      rxintr      bc  = min(bc, bp->b_resid);
                    rxstart     bc        = min(bc, bp->b_resid);
          vp.c      vpwrite     count = min(256,u.u_count);

mknod     sys2.c    mknod       mknod()
          sysent.c  nullsys     extern creat(), link(), unlink(), exec(), chdir(), gtime(), mknod(), chmod();
                                3, &mknod,                      /* 14 = mknod */
          sysent.tu.c           extern creat(), link(), unlink(), exec(), chdir(), gtime(), mknod(), chmod();
                                3, mknod,                       /* 14 = mknod */
          sysent.util.          extern creat(), link(), unlink(), exec(), chdir(), gtime(), mknod(), chmod();
                                3, mknod,                       /* 14 = mknod */

mmoff     mem.c     mmoff       mmoff(dev, cnt)
                    mmread      long mmoff();
                                while((offset=mmoff(dev,&cnt)) >= 0 && cnt != 0) {
                    mmwrite     long mmoff();
                                while((offset=mmoff(dev,&cnt)) >= 0 && cnt != 0) {

mmread    conf.70.c _           &nulldev,   &nulldev,       &mmread,        &mmwrite,       /*mm*/
                    nodev       extern mmread(), mmwrite();
          mem.c     mmread      mmread(dev)

mmwrite   conf.70.c _           &nulldev,   &nulldev,       &mmread,        &mmwrite,       /*mm*/
                    nodev       extern mmread(), mmwrite();
          mem.c     mmwrite     mmwrite(dev)

mpxchan   mx1.c     mpxchan     mpxchan()
          sysent.c  nullsys     extern mpxchan();
                                2, &mpxchan,                    /* 56 = creat mpx comm channel */

mpxname   mx2.c     mpxname     mpxname(cp)
                    mxopen      mpxname(cp);

mqsrch    messag.c  messag      if((rqp = mqsrch(u.u_arg[2])) == NULL) {
```

			if((rqp = mqsrch(u.u_arg[1])) == NULL) {
		mqsrch	mqsrch(pid)
		msgflush	if((mhd.mq_type) < 64 && (rqp2 = mqsrch(mhd.mq_sender))
	messagf.c	mqsrch	mqsrch()
msgenab	mx2.c	msgenab	msgenab(cp)
		msread	if(msgenab(cp))
		mxwcontr	if(msgenab(cp)){
msgflush	messag.c	messag	msgflush();
		msgflush	msgflush()
	messagf.c		msgflush()
	sys1.c	exit	msgflush();
msgfree	messag.c	messag	msgfree(hp,n);
		msgflush	msgfree(rhp,mhd.mq_size);
		msgfree	msgfree(hp,size)
		msgrecv	msgfree(rhp2,mhd->mq_size);
	messagf.c	msgfree	msgfree()
msghmov	messag.c	msgflush	msghmov(&mhd,rhp,MHREAD);
		msghmov	msghmov(mhd,hp,mode)
		msgrecv	msghmov(mhd,rhp2,MHREAD);
		msgremov	msghmov(&mhd,rhp1,MHREAD);
			msghmov(&mhd,rhp1,MHWRITE);
		msgsend	msghmov(mhd,hp,MHWRITE);
			msghmov(mhd,qp->mq_last,MHREAD);
			msghmov(mhd,qp->mq_last,MHWRITE);
msginit	main.c	main	maxmem -= msginit();
	messag.c	msginit	msginit().
	messagf.c		msginit()
msgmove	messag.c	messag	msgmove(hp, n, MSGIN);
		msgmove	msgmove(hp, len, mode)
		msgrecv	msgmove(rhp2, n, MSGOUT);
	messagf.c	msgmove	msgmove()
msgrecv	messag.c	messag	while(!msgrecv(rqp, n, &mhd)) {
		msgrecv	msgrecv(qp, type, mhd)
	messagf.c		msgrecv()
msgremov	messag.c	msgflush	msgremov(rqp1, rqp1, &mhd);
		msgrecv	msgremov(qp, rhp1, mhd);
		msgremov	msgremov(qp, rhp1, mhd2)
	messagf.c		msgremov()
msgsend	messag.c	messag	msgsend(rqp, &mhd, hp);
		msgflush	msgsend(rqp2, &mhd, rhp);
		msgsend	msgsend(qp, mhd, hp)
	messagf.c		msgsend()
msgsetup	messagf.c	msgsetup	msgsetup()
msread	mx2.c	msread	msread(fmp, cp)
		mxread	msread(fmp, FP->f_un.f_chan);

mswrite	mx2.c	mswrite	mswrite(fmp, cp)
		mxwrite	mswrite(fmp, FP->f_un.f_chan);
mtree	mx1.c	mpxchan	if (mtree(ngp, gp) == NULL)
		mtree	mtree(sub,master)
mup	mx1.c	mtree	if ((mtresiz=mup(master,sub)) == NULL) {
		mup	mup(master,sub)
mxclose	conf.70.c	—	&mxopen, &mxclose, &mxread, &mxwrite, /*mx*/
		nodev	extern mxopen(), mxclose(), mxread(), mxwrite(), mxioctl();
	mx2.c	mxclose	mxclose(dev, flag, cp)
mxfalloc	mx1.c	mpxchan	mxfalloc(((struct group *)cp)->g_file);
			mxfalloc(fp);
		mxfalloc	mxfalloc(fp)
mxioctl	conf.70.c	—	&mxioctl, &nulldev, 0,
		nodev	extern mxopen(), mxclose(), mxread(), mxwrite(), mxioctl();
	mx2.c	mxioctl	mxioctl(dev, cmd, addr, flag)
mxmove	mx2.c	mcread	cc = mxmove(q, B_READ);
		mcwrite	cc = mxmove(q, B_WRITE);
		msread	while (mxmove(q, B_READ) > 0)
		mswrite	cc = mxmove(q, B_WRITE);
		mxmove	mxmove(q, dir)
mxopen	conf.70.c	—	&mxopen, &mxclose, &mxread, &mxwrite, /*mx*/
		nodev	extern mxopen(), mxclose(), mxread(), mxwrite(), mxioctl();
	mx1.c	mpxchan	extern mxopen(), mcread(), uchar();
			if (cdevsw[i].d_open==mxopen) {
	mx2.c	mxopen	mxopen(dev, flag)
mxread	conf.70.c	—	&mxopen, &mxclose, &mxread, &mxwrite, /*mx*/
		nodev	extern mxopen(), mxclose(), mxread(), mxwrite(), mxioctl();
	mx2.c	mxread	mxread(dev)
mxrstrt	mx2.c	msread	mxrstrt(cp, q, SIGBLK);
		mxread	mxrstrt(cp, &cp->cx.datq, BLOCK|ALT);
		mxrstrt	mxrstrt(cp, q, b)
mxwcontrol	mx2.c	mxwcontrol	mxwcontrol(cp)
		mxwrite	} else mxwcontrol(cp);
mxwrite	conf.70.c	—	&mxopen, &mxclose, &mxread, &mxwrite, /*mx*/
		nodev	extern mxopen(), mxclose(), mxread(), mxwrite(), mxioctl();
	mx2.c	mxwrite	mxwrite(dev)
namei	acct.c	sysacct	if ((ip = namei(&uchar, 0))==NULL)
	fio.c	owner	if ((ip = namei(uchar, 0)) == NULL)
	maus.c	maus	if((ip = namei(&uchar, 0)) == NULL)
	mx1.c	mpxchan	ip = namei(uchar,1);
	nami.c	namei	namei(func, flag)
	sig.c	core	ip = namei(schar, 1);
	sys1.c	gethead	if ((ip = namei(uchar, 0)) == NULL)

```
           sys2.c      creat       ip = namei(&uchar, 1);
                       link        ip = namei(&uchar, 0);
                                   xp = namei(&uchar, 1);
                       mknod       ip = namei(&uchar, 1);
                       open        ip = namei(&uchar, 0);
                       saccess     ip = namei(&uchar, 0);
           sys3.c      getmdev     ip = namei(&uchar, 0);
                       smount      ip = namei(&uchar, 0);
                       stat        ip = namei(&uchar, 0);
           sys4.c      chdir       ip = namei(&uchar, 0);
                       chroot      ip = namei(&uchar, 0);
                       unlink      pp = namei(&uchar, 2);

newproc    main.c      main        if(newproc()) {
           slp.c       newproc     newproc()
           sys1.c      fork        if(newproc()) {

nextcp     mx2.c       mxread      cp = nextcp(gp);
                       nextcp      nextcp(gp)
                       xcp         struct chan *xcp(),*addch(),*nextcp();

nice       sys4.c      nice        nice()
           sysent.c    nullsys     extern utime(), stty(), gtty(), saccess(), nice(), sync(), kill();
                                   0, &nice,                   /* 34 = nice */
           sysent.tu.c             extern utime(), stty(), gtty(), saccess(), nice(), sync(), kill();
                                   0, nice,                    /* 34 = nice */
           sysent.util.            extern utime(), stty(), gtty(), saccess(), nice(), sync(), kill();
                                   0, nice,                    /* 34 = nice */

nodev      acct.c      -           sysacct() { nodev(); }
           conf.70.c               &vpopen,    &vpclose,       &nodev,         &vpwrite,       /*lp*/
                                   &nodev,             &nodev,         &nodev,         &nodev,         /*dc*/
                                   &nodev,             &nulldev,       0,
                                   /* 5*/     &vtopen,        &vtclose,       &nodev,         &vtwrite,       /*dp*/
                                   &nodev,             &nulldev,       0,
                                   &nodev,             &nodev,         &nodev,         &nodev,         /*dj*/
                                   &nodev,             &nulldev,       0,
                                   &dnopen,    &dnclose,       &nodev,         &dnwrite,       /*dn*/
                                   &nodev,             &nulldev,       0,
                                   &nodev,             &nulldev,       0,
                                   &nodev,             &nodev,         &nodev,         &nodev,         /*rk*/
                                   &nodev,             &nulldev,       0,
                                   /*10*/      &nodev,         &nodev,         &nodev,         &nodev,         /*rf*/
                                   &nodev,             &nulldev,       0,
                                   &nodev,             &nodev,         &nodev,         &nodev,         /*rp*/
                                   &nodev,             &nulldev,       0,
                                   &nodev,             &nodev,         &nodev,         &nodev,         /*tm*/
                                   &nodev,             &nulldev,       0,
                                   &nodev,             &nulldev,       0,
                                   &nodev,             &nulldev,       0,
                                   /*15*/      &nodev,         &nodev,         &nodev,         &nodev,         /*hs*/
                                   &nodev,             &nulldev,       0,
                                   &nodev,             &nodev,         &nodev,         &nodev,
                                   &nodev,             &nulldev,       0,
                                   /*20*/      &erropen,       &errclose,      &errread,       &nodev,         /*er*/
                                   &nodev,             &nulldev,       0,
```

```
                    hp45inpu    &nulldev,   &nulldev,       &nodev,         /*TEC*/
                                &nulldev,   &nulldev,       &nodev,         /*VT61*/
                                &nulldev,   &nulldev,       &nodev,         /*Tektronix 4023*/
                                &nulldev,   &nulldev,       &nodev,         /*TTY 40/1*/
                    nodev       extern nodev(), nulldev();
                                &nodev,             &nodev,         &nodev,         0,              /*rp*/
                                &nodev,             &nodev,         &nodev,         0,              /*rf*/
                                &nodev,             &nodev,         &nodev,         0,              /*tm*/
                                &nodev,             &nodev,         &nodev,         0,              /*tc*/
                    ttread      &nodev,             &nodev,         &nodev,         &nodev,
                                &nodev,             &nulldev,       &nodev,         &nodev,
                                &nodev,
                                &nodev,             &nodev,         &nodev,         &nodev,
                                &nodev,             &nulldev,       &nodev,         &nodev,
                                &nodev,
                                &nodev,             &nodev,         &nodev,         &nodev,
                                &nodev,             &nulldev,       &nodev,         &nodev,
                                &nodev,
                                &nodev,             &nodev,         &nodev,         &nodev,
                                &nodev,             &nulldev,       &nodev,         &nodev,
                                &nodev,
    ioctl.c     ttioccomm   extern nodev();
                            if (t >= nldisc || *linesw[t].l_ioctl == &nodev) {
    subr.c      nodev       nodov()
    tty.c       lsgtty      extern nodev();
                            if ((unsigned)i >= nldisc || *linesw[i].l_ioctl == &nodev)

nosys   messagf.c   messag      nosys();
    sysent.c    _               0, &nosys
                nullsys     extern nullsys(), nosys();
                                0, &nosys,                  /* 50 = x (formerly serrlog) */
                                0, &nosys,                  /* 52 = x */
                                        0, &nosys,                  /* 53 = smux */
                                0, &nosys,                  /* 55 = x */
                                0, &nosys,                  /* 58 = maus    */
    sysent.tu.c _               0, nosys
                nullsys     extern nullsys(), nosys();
                                0, nosys,                   /* 49 = message */
                                0, nosys,                   /* 50 = x (formerly serrlog) */
                                0, nosys,                   /* 51 = turn acct off/on */
                                0, nosys,                   /* 52 = x */
                                        0, nosys,                   /* 53 = smux */
                                0, nosys,                   /* 55 = x */
                                0, nosys,                   /* 56 = creat mpx comm channel */
                                0, nosys,                   /* 57 = x (was scall) */
                                0, nosys,                   /* 58 = maus    */
    sysent.util._               0, nosys
                nullsys     extern nullsys(), nosys();
                                0, nosys,                   /* 49 = message */
                                0, nosys,                   /* 50 = x (formerly serrlog) */
                                0, nosys,                   /* 51 = turn acct off/on */
                                0, nosys,                   /* 52 = x */
                                        0, nosys,                   /* 53 = smux */
                                0, nosys,                   /* 55 = x */
                                0, nosys,                   /* 56 = creat mpx comm channel */
                                0, nosys,                   /* 57 = x (was scall) */
```

```
                                    0, nosys,                       /* 58 = maus    */
            trap.c      nosys       nosys()

notavail    bio.c       aagetblk    notavail(bp);
                                    notavail(bp);
                        bflush      notavail(bp);
                        getablk     notavail(bp);
                        notavail    notavail(bp)

npclose     conf.70.c   _           &npopen,    &npclose,       &npread,        &npwrite,       /*np*/
                        nodev       extern npopen(), npclose(), npread(), npwrite(), npioctl();
            nmpipe.c    npclose     npclose(dev, flag)

npioctl     conf.70.c   _           &npioctl,   &nulldev,       0,
                        nodev       extern npopen(), npclose(), npread(), npwrite(), npioctl();
            nmpipe.c    npioctl     npioctl(dev, cmd, addr, flag)

npopen      conf.70.c   _           &npopen,    &npclose,       &npread,        &npwrite,       /*np*/
                        nodev       extern npopen(), npclose(), npread(), npwrite(), npioctl();
            nmpipe.c    npopen      npopen(dev, flag)

npread      conf.70.c   _           &npopen,    &npclose,       &npread,        &npwrite,       /*np*/
                        nodev       extern npopen(), npclose(), npread(), npwrite(), npioctl();
            nmpipe.c    npread      npread(dev)

npwrite     conf.70.c   _           &npopen,    &npclose,       &npread,        &npwrite,       /*np*/
                        nodev       extern npopen(), npclose(), npread(), npwrite(), npioctl();
            nmpipe.c    npwrite     npwrite(dev)

nulldev     conf.70.c   _           &vpioctl,   &nulldev,       0,
                                    &nodev,             &nulldev,       0,
                                    &nodev,             &nulldev,       0,
                                    &nodev,             &nulldev,       0,
                                    &nodev,             &nulldev,       0,
                                    &nulldev,   &nulldev,       &mmread,        &mmwrite,       /*mm*/
                                    &nodev,             &nulldev,       0,
                                    &nodev,             &nulldev,       0,
                                    &nodev,             &nulldev,       0,
                                    &nodev,             &nulldev,       0,
                                    &nodev,             &nulldev,       0,
                                    &hpopen,    &nulldev,       &hpread,        &hpwrite,       /*hp*/
                                    &nodev,             &nulldev,       0,
                                    &nodev,             &nulldev,       0,
                                    &nodev,             &nulldev,       0,
                                    &syopen,    &nulldev,       &syread,        &sywrite,       /*sy*/
                                    &syioctl,   &nulldev,       0,
                                    &nodev,             &nulldev,       0,
                                    &npioctl,   &nulldev,       0,
                                    &mxioctl,   &nulldev,       0,
                                    &nodev,             &nulldev,       0,
                        hp45inpu    &nulldev,   &nulldev,       &nulldev,       /*none (tty)*/
                                    &nulldev,   &nulldev,       &nodev,         /*TEC*/
                                    &nulldev,   &nulldev,       &nodev,         /*VT61*/
                                    &nulldev,   &nulldev,       &nodev,         /*Tektronix 4023*/
                                    &nulldev,   &nulldev,       &nodev,         /*TTY 40/1*/
                        nodev       extern nodev(), nulldev();
```

```
                                    &rxopen,    &nulldev,       &rxstrategy,    &rxtab,         /*rx*/
                                    &hpopen,    &nulldev,       &hpstrategy,    &hptab,         /*hp*/
                        ttread      &nodev,             &nulldev,       &nodev,         &nodev,
                                    &nodev,             &nulldev,       &nodev,         &nodev,
                                    &nodev,             &nulldev,       &nodev,         &nodev,
                                    &nodev,             &nulldev,       &nodev,         &nodev,
                                    &mcread,    &nulldev,       &mcwrite,       &nulldev,
                                    &nulldev,   &nulldev,       &nulldev,       &nulldev,
                                    &nulldev,
            subr.c      nulldev     nulldev()

nullsys     sysent.c    nullsys     extern nullsys(), nosys();
                                    0, &nullsys,                            /*  0 = indir */
            sysent.tu.c             extern nullsys(), nosys();
                                    0, nullsys,                     /*  0 = indir */
            sysent.util.            extern nullsys(), nosys();
                                    0, nullsys,                     /*  0 = indir */
            trap.c                  nullsys()

open        sys2.c      open        open()
            sysent.c    nullsys     extern rexit(), fork(), read(), write(), open(), close(), wait();
                                    2, &open,                       /*  5 = open */
            sysent.tu.c             extern rexit(), fork(), read(), write(), open(), close(), wait();
                                    2, open,                        /*  5 = open */
            sysent.util.            extern rexit(), fork(), read(), write(), open(), close(), wait();
                                    2, open,                        /*  5 = open */

openl       sys2.c      creat       openl(ip, FWRITE, 2);
                                    openl(ip, FWRITE, 1);
                        open        openl(ip, u.u_arg[1], 0);
                        openl       openl(ip, mode, trf)

openi       fio.c       openi       openi(ip, rw)
            sys2.c      openl       openi(rip, m&FWRITE);

owner       fio.c       owner       owner()
            sys4.c      chgrp       if ((ip = owner()) == NULL)
                        chmod       if ((ip = owner()) == NULL)
                        chown       if ((ip = owner()) == NULL)
                        utime       if ((ip = owner()) == NULL)

panic       alloc.c     getfs       panic("no fs");
                        iinit       panic("iinit");
                        putfs       panic("putfs");
                        rofs        panic("no fs");
            bio.c       aagetblk    panic("blkdev");
                                    panic("devtab");
                        swap        panic("IO err in swap");
            clock.c     timeout     panic("Timeout table overflow");
            iget.c      iget        panic("no imt");
            main.c      main        panic("buffers");
                                                panic("ddcmp char pool too large\n");
                                    panic("external clist too big\n");
                                    panic("no clock");
            prf.c       panic       panic(s)
            slp.c       newproc     panic("no procs");
```

```
                    setrun      panic("Running a dead proc");
          subr.c    bmap        panic("bmap");
          sys1.c    exece       panic(ooswap);
          sys3.c    smount      panic(ooswap);
          text.c    xalloc      panic(ooswap);
                    xswap       panic(ooswap);
          trap.c    trap        panic("trap");
                                panic("parity");

passc     hf.c      hfread      if((c =passc(c1)) < 0)                  break;
                                c = passc(c1);
          mx2.c     mcread      passc(*mp++);
          subr.c    passc       passc(c)
          trans.c   trsread     passc(c);
                                if(passc(c) == -1)
          tty.c     ttread      while (tp->t_canq.c_cc && passc(getc(&tp->t_canq))>=0);
          vs.c      vsread      passc(c);
                                if(passc(c) == -1)

pause     sys4.c    pause       pause()
          sysent.c  nullsys     extern setuid(), getuid(), stime(), ptrace(), alarm(), fstat(), pause();
                                0, &pause,                  /* 29 = pause */
          sysent.tu.c           extern setuid(), getuid(), stime(), ptrace(), alarm(), fstat(), pause();
                                0, pause,                   /* 29 = pause */
          sysent.util.          extern setuid(), getuid(), stime(), ptrace(), alarm(), fstat(), pause();
                                0, pause,                   /* 29 = pause */

pfail     mch.70.s  _pfail      _pfail:
                                _pfail:

physio    bio.c     physio      physio(strat, dev, rw, nblocks)
          dmc11.c   dmcboot     physio(&dmcstrategy,dev,B_WRITE,0);
                    dmcread     physio(dmcstrategy,dev,B_READ,0);
                    dmcwrite    physio(dmcstrategy,dev,B_WRITE,0);
          hps.c     hpread      physio(hpstrategy, dev, B_READ, nblks);
                    hpwrite     physio(hpstrategy, dev, B_WRITE, nblks);
          hs.c      hsread      physio(hsstrategy, dev, B_READ,
                    hswrite     physio(hsstrategy, dev, B_WRITE,
          ht.c      htread      physio(htstrategy, dev, B_READ, 0);
                    htwrite     physio(htstrategy, dev, B_WRITE, 0);
          lfh.c     lfget       physio(*bdevsw[lfdev>>8].d_strategy,lfdev,rwflag,0);
          rf.c      rfread      physio(rfstrategy, dev, B_READ, NRFBLK*(dev.d_minor+1));
                    rfwrite     physio(rfstrategy, dev, B_WRITE, NRFBLK*(dev.d_minor+1));
          rk.c      rkread      physio(rkstrategy, dev, B_READ, NRKBLK*nblks);
                    rkwrite     physio(rkstrategy, dev, B_WRITE, NRKBLK*nblks);
          rp.c      rpread      physio(rpstrategy, dev, B_READ, nblks);
                    rpwrite     physio(rpstrategy, dev, B_WRITE, nblks);
          rx.c      rxread      physio(rxstrategy, dev, B_READ, nblks);
                    rxwrite     physio(rxstrategy, dev, B_WRITE, nblks);
          tm.c      tmstart     physio(tmstrategy, dev, B_READ, 0);
                                physio(tmstrategy, dev, B_WRITE, 0);

pimove    mx2.c     iomove      pimove(padr, cnt, flg);
          rdwri.c   pimove      pimove(cp, n, flag)
                    readi       pimove(paddr(bp)+on, n, B_READ);
                    writei      pimove(paddr(bp)+on, n, B_WRITE);
```

```
            vp.c          vpwrite     pimove(vp->vp_buf->b_paddr + vp->vp_offset, count, B_WRITE);

pipe        nmpipe.c      npopen      pipe();
            pipe.c        pipe        pipe()
            sysent.c      nullsys     extern ulimit(), setpgrp(), tell(), dup(), pipe(), times(), profil();
                                      0, &pipe,                    /* 42 = pipe */
            sysent.tu.c               extern ulimit(), setpgrp(), tell(), dup(), pipe(), times(), profil();
                                      0, pipe,                     /* 42 = pipe */
            sysent.util.              extern ulimit(), setpgrp(), tell(), dup(), pipe(), times(), profil();
                                      0, pipe,                     /* 42 = pipe */

pir         clock.c       clock       pir(0);
                          int         int (*pir_fn)()     &pir;
                                      pir(pri)

plock       acct.c        acct        plock(ip);
                          sysacct     plock(acctp);
            fio.c         closef      plock(ip);
            mx1.c         addch       plock(ip);
            mx2.c         mxclose     plock(ip);
                          mxopen      plock(mpxip);
            nami.c        namei       plock(mpxip);
                                      plock(dp);
            pipe.c        plock       plock(ip)
                          readp       plock(ip);
                          writep      plock(ip);
            sys1.c        exit        plock(u.u_cdir);
                                      plock(u.u_rdir);
            sys2.c        rdwr        plock(ip);
            sys3.c        sumount     plock(ip);
            sys4.c        chdir       plock(u.u_cdir);
                          chroot      plock(oldroot);

prdev       alloc.c       alloc       prdev(E_FSNS, dev);
                          badblock    prdev(E_FSBB, dev);
                          getfs       prdev(E_FSBC, dev);
                          ialloc      prdev(E_FSOI, dev);
            prf.c         prdev       prdev(type, dev)

prefix      vtmon.c       vtprocent   if (prefix("getty", pcp->pr_name))
                                      if (prefix("getty", pcp->pr_name))
                                      if(!prefix("getty", pcp->pr_name));
            vtutil.c      prefix      prefix(a, b)

prele       acct.c        acct        prele(ip);
                          sysacct     prele(ip);
            iget.c        iput        prele(ip);
                                      prele(ip);
            mx1.c         addch       prele(ip);
                          mpxchan     prele(ip);
            mx2.c         mcread      prele(mpxip);
                          mxclose     prele(ip);
                          mxopen      prele(mpxip);
            pipe.c        prele       prele(ip)
                          readp       prele(ip);
                                      prele(ip);
```

```
                    writep      prele(ip);
                                prele(ip);
                                prele(ip);
                                prele(ip);
          sys2.c    link        prele(ip);
                    open1       prele(rip);
                    rdwr        prele(ip);
          sys3.c    smount      prele(ip);
          sys4.c    chdir       prele(ip);
                    chroot      prele(ip);

printf    dmc11.c   dmcoint     printf("DMC11(%d): block not found\n",dev.d_minor);
          flo.c     falloc      printf("no file\n");
          hps.c     hp_pwrup    printf("\nRP06/5/4 Disk Drive(s)");
                                printf(" %d", unit);
                                printf(" offline.  Manual attention required.\n\n");
                    hpstrate    printf("\nhpstrategy: unit out of range\n");
                    hpustart    printf("\nRP06/5/4 drive %d offline\n", unit);
          hs.c      hs_pwrup    printf("\n**RS04 Disk Drive(s) offline.  Manual attention required.\n");
          iget.c    iget        printf("Inode table overflow\n");
          main.c    main        printf("CB-UNIX RELEASE 2.1");
                                printf("    *** SYSTEM SIZE=%d.%dK **** AVAIL MEM=%d.%dK ***\n",
          mx2.c     mxwcontrol  printf("M_IOCTL\n");
          prf.c     panic       printf("panic: %s\n", s);
                    prdev       printf("%s on dev %l/%l\n", emtab[type], major(dev), minor(dev));
                    printf      printf(fmt,x1)
                    restart     printf("\177\n%d", pwr_fail);
                                printf("\177\n\n*** POWER FAIL \177");
                                printf("RESTART ***\n\n");
          sys5.c    event       printf("semaop %d(%d): %s; uid=%d; pid=%d\n", op, sematest,
          text.c    xalloc      printf("out of text");
          trap.c    stray       printf("stray interrupt at %o\n", addr);
                    trap        printf("ka6 = %o\n", *ka6);
                                printf("aps = %o\n", aps);
                                printf("trap type %o\n", dev);
                                printf("parity\n");
                                printf("%o ", MEMORY->r[i]);
                                printf("\n");

printn    prf.c     printf      printn(*adx, c=='o'? 8: 10);
                    printn      printn(n, b)
                                printn(a, b);

proclock  sysch.c   lock        proclock();
                    proclock    proclock()

procxmt   sig.c     procxmt     procxmt()
                    stop        if ((cp->p_flag&STRC)==0 || procxmt())

profil    sys4.c    profil      profil()
          sysent.c  nullsys     extern ulimit(), setpgrp(), tell(), dup(), pipe(), times(), profil();
                                4, &profil,                 /* 44 = prof */
          sysent.tu.c           extern ulimit(), setpgrp(), tell(), dup(), pipe(), times(), profil();
                                4, profil,                  /* 44 = prof */
          sysent.util.          extern ulimit(), setpgrp(), tell(), dup(), pipe(), times(), profil();
                                4, profil,                  /* 44 = prof */
```

psig	clock.c	clock	psig();
	sig.c	psig	psig()
	slp.c	HASH	goto psig;
			goto psig;
			psig:
	trap.c	trap	psig();
psignal	clock.c	clock	psignal(pp,SIGCLK);
	mx2.c	mswrite	psignal(u.u_procp, SIGPIPE);
	pipe.c	writep	psignal(u.u_procp, SIGPIPE);
	prf.c	int	psignal(pp, SIGPWR);
	sig.c	procxmt	psignal(u.u_procp, ipc.ip_data);
		psignal	psignal(p, sig)
		signal	psignal(p, sig);
	sys1.c	exece	psignal(u.u_procp, SIGKIL);
		exit	psignal(&proc[1], SIGCLD);
			psignal(q, SIGCLD);
		getxfile	psignal(u.u_procp, SIGTRC);
	sys2.c	seek	psignal(u.u_procp, SIGSYS);
	sys4.c	kill	psignal(p, signo >= 0 ? signo : -signo);
		ssig	psignal(u.u_procp, SIGCLD);
	text.c	xalloc	psignal(u.u_procp, SIGKIL);
	trap.c	trap	psignal(u.u_procp, SIGFPT);
			psignal(u.u_procp, i);
ptrace	sig.c	ptrace	ptrace()
	sysent.c	nullsys	extern setuid(), getuid(), stime(), ptrace(), alarm(), fstat(), pause();
			3, &ptrace, /* 26 = ptrace */
	sysent.tu.c		extern setuid(), getuid(), stime(), ptrace(), alarm(), fstat(), pause();
			3, ptrace, /* 26 = ptrace */
	sysent.util.		extern setuid(), getuid(), stime(), ptrace(), alarm(), fstat(), pause();
			3, ptrace, /* 26 = ptrace */
punlock	sys1.c	exit	if(punlock())
	syscb.c	lock	if(punlock() == 0)
		punlock	punlock()
putc	ds40.c	ds40outp	putc(ESC,qp);
			putc(QESC,qp);
			putc(0213,qp);
			putc(ESC,qp);
			putc('G',qp);
			putc(ESC,qp);
			putc('B',qp);
			putc(QESC,qp);
			putc(0213,qp);
			putc(QESC,qp);
			putc(0216,qp);
			putc(QESC,qp);
			putc(0212,qp);
	hf.c	hfrint	putc(QESC,&tp->t_rawq);
			putc(QESC,&tp->t_rawq);
			if(putc(0377,&tp->t_rawq) == 0) tp->t_delct++;
	hp45.c	hp45outp	putc(ESC, qp);
			putc('&', qp);

```
                                putc(*cp, qp);
                                putc(001,qp);
                                putc(' ',qp);
                                putc(000,qp);
                                putc(' ',qp);
                                putc(vid_attr, qp);
                                putc(ESC, qp);
                                putc(0112, qp);
lp.c            lpoutput        putc(c, &lp11);
lpm.c                           putc(c, lp);
mch.70.s        _putc           _putc:
mx2.c           b_to_q          if (putc(*buf++, q) == -1)
subr.c          putw            putc(c, p);
                                putc(c>>8, p);
tex.c           texoutput       putc(ESC, qp);
                                putc(0126,qp);
                                putc(US, qp);
                                putc(QESC, qp);
                                putc(0203, qp);
                                putc(QESC, qp);
                                putc(0201, qp);
                                putc(ESC, qp);
                                putc(0132, qp);
                                putc(QESC, qp);
                                putc(0203, qp);
                                putc(034, qp);
                                putc(QESC, qp);
                                putc(0201, qp);
trans.c         trsinput        putc(c, &tp->ts_rawq);
                trswrite        while(putc(c, &tp->ts_outq)) {
tty.c           canon           putc(*bp++, c);
                                if (putc(XON, &tp->t_outq)==0) {
                qputc           putc(c, qp);
                                putc(c, qp);
                ttwrite         putc(0177, &tp->t_outq);
                                putc(0177, &tp->t_outq);
                                putc(QESC, &tp->t_outq);
                                putc(QERK, &tp->t_outq);
                                putc(QESC, &tp->t_outq);
                                putc(XOFF, &tp->t_outq);
                ttxint          putc(c, &spytp->t_outq);
                ttyinput        putc(ESC, &tp->t_rawq);
                                putc(c, &tp->t_rawq);
                                putc(c, &tp->t_rawq);
                                putc('\b', &tp->t_outq);
                                        putc(' ', &tp->t_outq);
                                putc('\b', &tp->t_outq);
                                putc(c, &tp->t_rawq);
                                if (putc(XOFF, &tp->t_outq)==0) {
                                putc(c, &tp->t_rawq);
                                if ((t_flags&RAW)==0 && putc(0377, &tp->t_rawq)==0)
                ttyoutl         putc(cbave, qp);
                                putc(QESC, qp);
                                putc(c|0200, qp);
                ttyoutput       putc(c, &tp->t_rawq);
vs.c            vsinput         putc(c, &tp->t_rawq);
```

```
                    vswrite     putc(0, &tp->t_outq);
                                putc(c, &tp->t_outq);
                                putc(0, &tp->t_outq);
            vt100.c cvtdec      putc( i/10 + '0', qp);
                                putc( i%10 + '0', qp);
                    vttrans     putc(ESC, qp);
                                putc(ESC, qp);
                                putc(QESC, qp);
                                putc(QESC, qp);
                                putc(SEMI, qp);
                                putc(*ap, qp);
            vt61.c  vt61output  putc(' ', qp);
                                putc(ESC, qp);
                                putc('O', qp);
                                else    putc('P', qp);
                                putc(*cp, qp);
                                putc(' ', qp);
                                putc(ESC, qp);
                                putc(0112, qp);
                                putc(QESC, qp);
                                putc(TDELAY, qp);

putchar     prf.c   printf      putchar(c);
                                putchar('-');
                                putchar(c);
                                putchar(*adx);
                    printn      putchar(lrem(n, b) + '0');
                    putchar     putchar(c)
                                putchar('\r');
                                putchar(0177);
                                putchar(0177);
                                putchar(0);

puterec     errlog.c logberr    puterec(eup,E_BLK);
                    logovfl     puterec(eup, E_OVFL);
                    logparit    puterec(eup,E_PRTY);
                    logpower    puterec(eup, E_POWER);
                    logprdev    puterec(eup, E_PRDV);
                    logstart    puterec(eup,E_GOCH);
                    logstray    puterec(eup,E_STRAY);
                    logtchg     puterec(eup,E_XCHG);
                    puterec     puterec(eup,type)

putfs       alloc.c alloc       putfs(fp);
                                putfs(fp);
                    free        putfs(fp);
                    ialloc      putfs(fp);
                                putfs(fp);
                    ifree       putfs(fp);
                    putfs       putfs(bp)

putw        mx2.c   scontrol    putw(event,q);
                                putw(value,q);
            subr.c  putw        putw(c, p)

pwr_on      mch.70.s _pwr_on    _pwr_on:
```

```
            prf.c       int         pwr_on();

pwrfail     mch.70.s    pfail           jsr     r0,call; jmp _pwrfail
            prf.c       int         pwrfail(dev, sp, r1, nps, r0, pc, ps)
                                    pir_fn = pwrfail;

qputc       ds40.c      ds40outp    qputc(*cp&0177,qp);
            hp45.c      hp45outp    qputc(c&0177, qp);
                                    qputc(c/10 + '0', qp);
                                    qputc(c%10 + '0', qp);
                                    qputc('r', qp);
                                    qputc(c/10 + '0', qp);
                                    qputc(c%10 + '0', qp);
                                    qputc('C', qp);
            tec.c       tecoutput   qputc(cp[-2], qp);
                                    qputc(~atp->t_col, qp);
                                    qputc(~atp->t_row, qp);
            tex.c       texoutput   qputc(c&0177, qp);
                                    qputc(034, qp);
                                    qputc(32, qp);
                                    qputc(c+32, qp);
                                    qputc(atp->t_col+32, qp);
                                    qputc(atp->t_row+32, qp);
            tty.c       qputc       qputc(ac, aqp)
                        ttyout1     qputc(csave, qp);
            vt100.c     vttrans     qputc(ESC, qp);
                                    qputc( c, qp);
                                    qputc(c, qp);
            vt61.c      vt61output  qputc(c&0177, qp);
                                    qputc(atp->t_row+32, qp);
                                    qputc(atp->t_col+32, qp);

qswtch      mch.70.s    call            jsr     pc,_qswtch
            slp.c       expand      qswtch();
                        qswtch      qswtch()
            text.c      xexpand     qswtch();
            trap.c      trap        qswtch();

rdwr        sys2.c      rdwr        rdwr(mode)
                        read        rdwr(FREAD);
                        write       rdwr(FWRITE);

read        sys2.c      read        read()
            sysent.c    nullsys     extern rexit(), fork(), read(), write(), open(), close(), wait();
                                    2, &read,                   /*  3 = read */
            sysent.tu.c             extern rexit(), fork(), read(), write(), open(), close(), wait();
                                    2, read,                    /*  3 = read */
            sysent.util.            extern rexit(), fork(), read(), write(), open(), close(), wait();
                                    2, read,                    /*  3 = read */

readi       pipe.c      readp       readi(ip);
            rdwri.c     readi       readi(ip)
            sys1.c      gethead     readi(ip);
                        getxfile    readi(ip);
            sys2.c      rdwr        readi(ip); else
            text.c      xalloc      readi(rp);
```

```
readp       nmpipe.c    npread      readp(npp->p_rfp);
            pipe.c      readp       readp(fp)
            sys2.c      rdwr        readp(fp);

reboot      low.70.s    _reboot     _reboot:
            syscb.c     syscb       reboot();   /* no return */

restart     prf.c       int         restart(dev);
                        restart     restart(state)

retu        mch.70.s    _retu       _retu:
            slp.c       expand      retu(p->p_addr);
                        swtch       retu(proc[0].p_addr);
                                    retu(p->p_addr);

rexit       sys1.c      rexit       rexit()
            sysent.c    nullsys     extern rexit(), fork(), read(), write(), open(), close(), wait();
                                    0, &rexit,                      /*  1 = exit */
            sysent.tu.c             extern rexit(), fork(), read(), write(), open(), close(), wait();
                                    0, rexit,                       /*  1 = exit */
            sysent.util.            extern rexit(), fork(), read(), write(), open(), close(), wait();
                                    0, rexit,                       /*  1 = exit */

rfintr      rf.c        rfintr      rfintr()

rfopen      rf.c        tabinit     rfopen(dev, flag)

rfread      rf.c        rfread      rfread(dev)

rfstart     rf.c        rfintr      rfstart();
                        rfstart     rfstart();
                        rfstart     rfstart()
                        rfstrategy  rfstart();
                        tabinit     rfstart();

rfstrategy  rf.c        rfread      physio(rfstrategy, dev, B_READ, NRFBLK*(dev.d_minor+1));
                        rfstrategy  rfstrategy(bp)
                        rfwrite     physio(rfstrategy, dev, B_WRITE, NRFBLK*(dev.d_minor+1));

rfwrite     rf.c        rfwrite     rfwrite(dev)

rhstart     hps.c       hpstart     rhstart(bp, &HPADDR->hpda, bp->trksec, &HPADDR->hpbae);
            hs.c        hsstart     rhstart(bp, &HSADDR->hsda, addr<<1, &HSADDR->hsbae);
            ht.c        htstart     rhstart(bp, &HTADDR->htfc, -bp->b_bcount, &HTADDR->htbae);
            rh.c        rhstart     rhstart(bp, devloc, devblk, abae)

rkaddr      rk.c        rkaddr      rkaddr(bp)
                        rkstart     a = rkaddr(bp);

rkintr      rk.c        rkintr      rkintr()

rkopen      rk.c        tabinit     rkopen(dev, flag)

rkread      rk.c        rkread      rkread(dev)
```

```
rkstart     rk.c        rkintr      rkstart();
                                    rkstart();
                        rkstart     rkstart()
                        rkstrategy  rkstart();
                        tabinit     rkstart();

rkstrategy  rk.c        rkread      physio(rkstrategy, dev, B_READ, NRKBLK*nblks);
                        rkstrategy  rkstrategy(bp)
                        rkwrite     physio(rkstrategy, dev, B_WRITE, NRKBLK*nblks);

rkwrite     rk.c        rkwrite     rkwrite(dev)

rodst       rop.c       rodst       rodst(tp, csr, lsr)

rofs        alloc.c     rofs        rofs(dev)
            fio.c       access      if(rofs(ip->i_dev)) {
            iget.c      iupdat      if(rofs(rp->i_dev))

roopen      rop.c       roopen      roopen(tp)

rpintr      rp.c        rpintr      rpintr()

rpopen      rp.c        tabinit     rpopen(dev, flag)

rpread      rp.c        rpread      rpread(dev)

rpstart     rp.c        rpintr      rpstart();
                                    rpstart();
                        rpstart     rpstart()
                        rpstrategy  rpstart();
                        tabinit     rpstart();

rpstrategy  rp.c        rpread      physio(rpstrategy, dev, B_READ, nblks);
                        rpstrategy  rpstrategy(bp)
                        rpwrite     physio(rpstrategy, dev, B_WRITE, nblks);

rpwrite     rp.c        rpwrite     rpwrite(dev)

rx02wait    rx.c        rx02wait    rx02wait()
                        rxstart     rx02wait();
                                    rx02wait();
                                    rx02wait();
                                    rx02wait();
                                    rx02wait();

rx2factr    rx.c        rx2factr    rx2factr(sectr, psectr, ptrck)
                        rxstart     rx2factr( ((BSIZE/bc)*bp->b_blkno + (OFFSET/bc)),

rxdone      rx.c        rxdone      rxdone()
                        rxintr      rxdone();
                                    rxdone();
                                    rxdone();
                        rxstart     rxdone();

rxintr      rx.c        rxintr      rxintr()
```

```
rxioctl     rx.c        rxioctl     rxioctl(dev, cmd, addr, flag)

rxopen      conf.70.c   nodev       extern rxopen(), rxstrategy();
                                    &rxopen,    &nulldev,       &rxstrategy,    &rxtab,         /*rx*/
            rx.c        rxopen      rxopen(dev)

rxread      rx.c        rxread      rxread(dev)

rxstart     rx.c        NRXDEV      rxstart();
                        rxintr      rxstart();
                        rxstart     rxstart()

rxstrategy  conf.70.c   nodev       extern rxopen(), rxstrategy();
                                    &rxopen,    &nulldev,       &rxstrategy,    &rxtab,         /*rx*/
            rx.c        NRXDEV      rxstrategy(bp)
                        rxioctl     rxstrategy(bp);
                        rxopen      rxstrategy(bp);
                        rxread      physio(rxstrategy, dev, B_READ, nblks);
                        rxwrite     physio(rxstrategy, dev, B_WRITE, nblks);

rxwrite     rx.c        rxwrite     rxwrite(dev)

saccess     sys2.c      saccess     saccess()
            sysent.c    nullsys     extern utime(), stty(), gtty(), saccess(), nice(), sync(), kill();
                                    2, &saccess,                            /* 33 = access */
            sysent.tu.c             extern utime(), stty(), gtty(), saccess(), nice(), sync(), kill();
                                    2, saccess,                         /* 33 = access */
            sysent.util.            extern utime(), stty(), gtty(), saccess(), nice(), sync(), kill();
                                    2, saccess,                         /* 33 = access */

savfp       mch.70.s    _savfp      _savfp:
                        call            jsr     pc,_savfp
            prf.c       int         savfp();
            trap.c      trap        savfp();

savu        mch.70.s    _savu       _savu:
            slp.c       expand      savu(u.u_rsav);
                                    savu(u.u_ssav);
                        newproc     savu(u.u_rsav);
                                    savu(u.u_ssav);
                        swtch       savu(u.u_rsav);
            text.c      xexpand     savu(u.u_rsav);
                                    savu(u.u_ssav);
            trap.c      trap1       savu(u.u_qsav);

sbreak      sys1.c      sbreak      sbreak()
            sysent.c    nullsys     extern chown(), sbreak(), stat(), seek(), getpid(), smount(), sumount();
                                    1, &sbreak,                             /* 17 = break */
            sysent.tu.c             extern chown(), sbreak(), stat(), seek(), getpid(), smount(), sumount();
                                    1, sbreak,                              /* 17 = break */
            sysent.util.            extern chown(), sbreak(), stat(), seek(), getpid(), smount(), sumount();
                                    1, sbreak,                              /* 17 = break */

schar       nami.c      schar       schar()
            sig.c       core        extern schar();
                                    ip = namei(schar, 1);
```

```
sched       main.c      main        sched();
            slp.c       sched       sched()

schgrp      sys4.c      chgrp       schgrp(ip,u.u_arg[1].lobyte);
                        chown       schgrp(ip,u.u_arg[2].lobyte);
                        schgrp      schgrp(ip,a)

scontrol    fakemx.c    _           scontrol() {}
            mx2.c       mcstart     scontrol(cp, M_UBLK, 0);
                        msread      scontrol(cp, M_UBLK, 0);
                        mxclose     scontrol(cp, M_CLOSE, 0);
                        mxopen      scontrol(cp, msg+(cp->c_index<<8), u.u_uid);
                        mxread      scontrol(cp, M_CLOSE, 0);
                        mxwcontrol  scontrol(cp, M_BLK, 0);
                        scontrol    scontrol(cp,event,value)

sdata       fakemx.c    _           sdata() {}
            mx2.c       mswrite     sdata(cp);
                                    if (cp->c_flags&YGRP)  sdata(cp);
                                    if (cp->c_flags&XGRP)  sdata(cp);
                        mxread      sdata(cp);
                                    sdata(cp);
                        scontrol    if (sdata(cp) == NULL)
                        sdata       sdata(gp)
                        wflush      sdata(cp);
            tty.c       ttyinput    sdata(tp->t_chan);

seek        sys2.c      seek        seek()
            sysent.c    nullsys     extern chown(), sbreak(), stat(), seek(), getpid(), smount(), sumount();
                                    3, &seek,                       /* 19 = seek */
            sysent.tu.c             extern chown(), sbreak(), stat(), seek(), getpid(), smount(), sumount();
                                    3, seek,                        /* 19 = seek */
            sysent.util.            extern chown(), sbreak(), stat(), seek(), getpid(), smount(), sumount();
                                    3, seek,                        /* 19 = seek */

semafree    sys1.c      exit        semafree();
            sys5.c      semafree    semafree()

setgid      sys4.c      setgid      setgid()
            sysent.c    nullsys     extern setgid(), getgid(), ssig(), messag(), sysacct();
                                    0, &setgid,                     /* 46 = setgid */
            sysent.tu.c             extern setgid(), getgid(), ssig(), messag(), sysacct();
                                    0, setgid,                      /* 46 = setgid */
            sysent.util.            extern setgid(), getgid(), ssig(), messag(), sysacct();
                                    0, setgid,                      /* 46 = setgid */

setpgrp     sys5.c      setpgrp     setpgrp()
            sysent.c    nullsys     extern ulimit(), setpgrp(), tell(), dup(), pipe(), times(), profil();
                                    0, &setpgrp,                        /* 39 = setpgrp */
            sysent.tu.c             extern ulimit(), setpgrp(), tell(), dup(), pipe(), times(), profil();
                                    0, setpgrp,                     /* 39 = setpgrp */
            sysent.util.            extern ulimit(), setpgrp(), tell(), dup(), pipe(), times(), profil();
                                    0, setpgrp,                     /* 39 = setpgrp */

setpri      clock.c     clock       setpri(pp);
```

```
                                setpri(u.u_procp);
          slp.c       setpri    setpri(up)
          trap.c      trap      curpri = setpri(u.u_procp);

setregs   sys1.c      exece     setregs();
                    setregs     setregs()

setrq     slp.c       newproc   setrq(rpp);
                    qswtch      setrq(u.u_procp);
                    setrq       setrq(p)
                    setrun      setrq(p);

setrun    sig.c       psignal   setrun(p);
                    ptrace      setrun(p);
          slp.c     setrun      setrun(p)
                    wakeup      setrun(p);
          sys1.c    exit        setrun(q);

setuid    sys4.c    setuid      setuid()
          sysent.c  nullsys     extern setuid(), getuid(), stime(), ptrace(), alarm(), fstat(), pause();
                                0, &setuid,                 /* 23 = setuid */
          sysent.tu.c           extern setuid(), getuid(), stime(), ptrace(), alarm(), fstat(), pause();
                                0, setuid,                  /* 23 = setuid */
          sysent.util.          extern setuid(), getuid(), stime(), ptrace(), alarm(), fstat(), pause();
                                0, setuid,                  /* 23 = setuid */

sgtty     tty.c     gtty        sgtty(vp);
                    sgtty       sgtty(v)
                    stty        sgtty(0);

shuffle   slp.c     sched       shuffle();
                    shuffle     shuffle()

signal    mx1.c     mpxchan     signal(cp->c_pgrp, i);
          sig.c     signal      signal(pgrp, sig)
          tty.c     ttydst      signal(tp->t_pgrp, SIGHUP);
                    ttyinput    signal(tp->t_pgrp, c==034 ? SIGQIT:SIGINT);
          vs.c      vsdst       signal(tp->t_pgrp, SIGHUP);

sleep     alloc.c   alloc       sleep(&fp->s_flock, PINOD);
                    free        sleep(&fp->s_flock, PINOD);
                    getfs       sleep(p, PRIBIO);
                    ialloc      sleep(&fp->s_ilock, PINOD);
          bio.c     aagetblk    sleep((caddr_t)bp, PRIBIO+1);
                                sleep((caddr_t)&bfreelist, PRIBIO+1);
                    getablk     sleep((caddr_t)bp, PRIBIO+1);
                    getbfh      sleep((caddr_t)&bfreelist,PRIBIO+1);
                    iowait      sleep((caddr_t)bp, PRIBIO);
                    physio      sleep(bp, PRIBIO);
                    swap        sleep((caddr_t)bp, PSWP+1);
                                sleep((caddr_t)bp, PSWP);
          dmc11.c   dmcboot     sleep(bp, DMCSPRI);
                                sleep(&bp, DMCSPRI);
                                sleep(bp, DMCSPRI);
                    dmcioctl    sleep(uaddr,DMCSPRI);
                                sleep(uaddr,DMCSPRI);
```

```
                                sleep(uaddr, DMCSPRI);
                                sleep(bp,DMCSPRI);
dn.c            dnopen          sleep(&(dnadr->dn_datum),DNPRI);
                                sleep(dnadr,DNPRI);
                                sleep(dnadr,DNPRI);
                dnwrite         sleep(dnadr,DNPRI);
                                sleep(dnadr,DNPRI);
                                sleep(dnadr,DNPRI);
                                sleep(dnadr,DNPRI);
                                sleep(dnadr,DNPRI);
errlog.c        geterec         sleep(&err.e_orq,PZERO+1);
hf.c            hfclose         sleep(tp,TTOPRI);
                hfread          sleep(tp,TTOPRI);
ht.c            hcommand        sleep(bp, PRIBIO);
                tabinit         sleep(&lbolt,-1);
iget.c          iget            sleep(p, PINOD);
jy.c            jyread          sleep(jyq,JYSPRI);
lfh.c           lfio            sleep(bp,PRIBIO);
                lflock          sleep(&lfflag,LFPRI);
lp.c            lpoutput        sleep(&lp11, LPPRI);
lpm.c                           sleep(lp, LPPRI);
messag.c        messag          sleep(&rqp->mq_flag, PMSG);
                                sleep(&msgbase, PMSG);
                                sleep(rqp, PMSG);
mx2.c           msread          sleep((caddr_t)q,TTIPRI);
                mswrite         sleep((caddr_t)q+1,TTOPRI);
                mxopen          sleep((caddr_t)cp,PZERO);
                mxread          sleep((caddr_t)&qp->q_datq, TTIPRI);
                mxwcontrol      sleep(cp, TTOPRI);
                mxwrite         sleep(waddr, TTOPRI);
                wflush          sleep((caddr_t)q+2,TTOPRI);
nmpipe.c        npopen          sleep((caddr_t)rpp, -1);
pipe.c          plock           sleep((caddr_t)ip, -3);
                readp           sleep((caddr_t)ip+2, PPIPE);
                writep          sleep((caddr_t)ip+1, PPIPE);
rh.c            mapalloc        sleep(ubmap, PSWP);
sig.c           ptrace          sleep(&ipc, IPCPRI);
                                sleep(&ipc, IPCPRI);
slp.c           HASH            sleep(chan, pri)
                sched           sleep(&runout, PSWP);
                                sleep(&runin, PSWP);
sys1.c          wait            sleep(u.u_procp, PWAIT);
sys4.c          pause           sleep(&u, PSLEP);
sys5.c          event           while (semi[s]==0) sleep(&semi[s],PEVT);
                                sleep(&semi[s],PEVT);
                                sleep(&semi[s], PEVT);
syscb.c         lock            sleep(&runlock,-1);
text.c          xlock           sleep(xp, PSWP);
tm.c            tabinit         sleep(&lbolt, -1);
                tcommand        sleep(bp, PRIBIO);
trans.c         trsread         sleep(tp, TTIPRI);
                                sleep(&tp->ts_canq, TTIPRI);
                trswrite        sleep(tp, TTOPRI);
                                sleep((caddr_t)&tp->ts_outq, PZERO);
                                sleep((caddr_t)&lbolt, -1);
tty.c           canon           sleep(&tp->t_rawq, TTIPRI);
```

```
                    ttread      sleep(tp, TTIPRI);
                    ttwrite     sleep(tp, TTOPRI);
                                sleep((caddr_t)&tp->t_outq, hqflag?-1:PZERO);
                    wflushtty   sleep(&tp->t_outq, TTOPRI);
          vp.c      vpready     sleep(vp, VPPRI);
          vs.c      vsread      sleep(tp, TTIPRI);
                                sleep(&tp->t_rawq, TTIPRI);
                    vswrite     sleep(tp, TTOPRI);
                                sleep(&tp->t_outq, TTOPRI);
          vt11.c    vtfrmfree   sleep(&vp->v_fmmpc,VTSPRI);
                    vtnxpc      sleep(&vttab.v_fempc,VTSPRI);
                    vtopen      sleep(vp,VTSPRI);
                    vtread      sleep(&vttab,VTSPRI);
          vtlp.c    vtlpread    sleep(vp11,LPSPRI);

smount    sys3.c    smount      smount()
          sysent.c  nullsys     extern chown(), sbreak(), stat(), seek(), getpid(), smount(), sumount();
                                3, &smount,                 /* 21 = mount */
          sysent.tu.c           extern chown(), sbreak(), stat(), seek(), getpid(), smount(), sumount();
                                3, smount,                  /* 21 = mount */
          sysent.util.          extern chown(), sbreak(), stat(), seek(), getpid(), smount(), sumount();
                                3, smount,                  /* 21 = mount */

spl0      alarm.c   almopen     spl0();
          alloc.c   getfs       spl0();
                                spl0();
          bio.c     aagetblk    spl0();
                                spl0();
                                spl0();
                    bflush      spl0();
                    getablk     spl0();
                    iowait      spl0();
                    physio      spl0();
                    swap        spl0();
          dh.c      dhcntrl     spl0();
                    dhparam     spl0();
          dmc11.c   dmcclose    spl0();
                    dmcioctl    spl0();
                    dmcopen     spl0();
          dn.c      dnopen      spl0();
                    dnwrite     spl0();
                                spl0();
          dz.c      dzparam     spl0();
          hf.c      hfioctl     spl0();
                    hfread      spl0();
          hps.c     hpstrate    spl0();
          hs.c      hsstrate    spl0();
          ht.c      hcommand    spl0();
                    htstrate    spl0();
          jy.c      jyread      spl0();
          lfh.c     lfio        spl0();
          lp.c      lpoutput    spl0();
          lpm.c                 spl0();
          mch.70.s  _spl0       _spl0:
          ms2.c     msvwrite    spl0();
                                spl0();
```

```
                                        spl0();
        rf.c            rfstrategy      spl0();
        rh.c            mapalloc        spl0();
        rk.c            rkstrategy      spl0();
        rp.c            rpstrategy      spl0();
        rx.c            RX2_BC2         #define RXLOPRI()   spl0()
        slp.c           HASH            spl0();
                                        spl0();
                                        spl0();
                        sched           spl0();
                                        spl0();
                        swtch           spl0();
                                        spl0();
        sys4.c          alarm           spl0();
                        ftime           spl0();
                        pause           spl0();
        tm.c            tcommand        spl0();
                        tmstrategy      spl0();
        trans.c         trsread         spl0();
                                        spl0();
                                        spl0();
                        trswrite        spl0();
                                        spl0();
                                        spl0();
        tty.c           canon           spl0();
                        hqrelse         spl0();
                        ttread          spl0();
                        ttwrite         spl0();
                                        spl0();
                                        spl0();
                        ttyopen         spl0();
                        wflushtty       spl0();
        vp.c            vpready         spl0();
        vs.c            vsioctl         spl0();
                                        spl0();
                        vsopen          spl0();
                        vsread          spl0();
                                        spl0();
                                        spl0();
                        vswrite         spl0();
        vtl1.c          vtfrmfree       spl0();
                        vtnxpc          spl0();
                        vtopen          spl0();
                        vtread          spl0();
        vtlp.c          vtlpread        spl0();

spl1    clock.c         clock           spl1();
        mch.70.s        _spl1           _spl1:

spl4    lp.c            lpoutput        spl4();
        lpm.c                           spl4();
        mch.70.s        _spl4           _spl4:
        vp.c            vpready         spl4();
        vtl1.c          vtfrmfree       spl4();
                        vtnxpc          spl4();
                        vtopen          spl4();
```

```
                    vtread      spl4();
            vtlp.c  vtlpread    spl4();

spl5        alarm.c alrmopen    spl5();
            clock.c int         spl5();
            dh.c    dhcntrl     spl5();
                    dhparam     spl5();
            dhdm.c  dmcntrl     sps = spl5();
            dmc11.c dmcclose    spl5();
                    dmcclr      spl5();
                    dmcinit     spl5();
                    dmcioctl    spl5();
                    dmcopen     spl5();
                    dmcstrat    spl5();
            dn.c    dnopen      spl5();
                    dnwrite     spl5();
            dz.c    dzparam     spl5();
            hf.c    hfioctl     spl5();
                    hfread      spl5();
            hps.c   hpstrate    spl5();
            hs.c    hsstrate    spl5();
            ht.c    hcommand    spl5();
                    htstrate    spl5();
                    tabinit     spl5();
            jy.c    jyread      spl5();
            mch.70.s _spl5      _spl5:
            rf.c    rfstrategy  spl5();
            rk.c    rkstrategy  spl5();
            rp.c    rpstrategy  spl5();
            rx.c    RX2_BC2     #define RXHIPRI()   spl5()
            tm.c    tabinit     spl5();
                    tcommand    spl5();
                    tmstrategy  spl5();
            trans.c trsread     spl5();
                                spl5();
                    trswrite    spl5();
                                spl5();
            tty.c   canon       spl5();
                    flushtty    sps = spl5();
                    hqrelse     spl5();
                    ttread      spl5();
                    ttstart     sps = spl5();
                    ttwrite     spl5();
                                spl5();
                    ttyctl      sps = spl5();
                    ttyopen     spl5();
                    wflushtty   spl5();
            vs.c    vsioctl     spl5();
                                spl5();
                    vsopen      spl5();
                    vsread      spl5();
                                spl5();
                    vswrite     spl5();

spl6        alloc.c getfs       spl6();
            bio.c   aagetblk    spl6();
```

```
                                    spl6();
                        bflush      spl6();
                        brelse      s = spl6();
                        getablk     spl6();
                        getbfh      sps = spl6();
                        hrelse      sps = spl6();
                        iowait      spl6();
                        physio      spl6();
                        swap        spl6();
                                    spl6();
              lfh.c     lfio        spl6();
                        lflock      spl6();
                        lfrlse      spl6();
              mch.70.s  _spl6       _spl6:
              messag.c  msgfree     s = spl6();
                        msgremov    s = spl6();
                        msgsend     s = spl6();
              mx1.c     xcp         s = spl6();
              mx2.c     mcwrite     s = spl6();
                        msread      s = spl6();
                        mswrite     spl6();
                        mxread      s = spl6();
                        mxrstrt     s = spl6();
                        mxwcontrol  s = spl6();
                        scontrol    s = spl6();
                        sdata       s = spl6();
                        wflush      s = spl6();
              rh.c      mapalloc    spl6();
              slp.c     HASH        s = spl6();
                        sched       spl6();
                                    spl6();
                                    spl6();
                        setrq       s = spl6();
                        swtch       spl6();
                        wakeup      s = spl6();
              subr.c    putw        s = spl6();

spl7          clock.c   kiltout     s = spl7();
                        timeout     s = spl7();
              errlog.c  freeslot    sps = spl7();
                        geterec     sps = spl7();
                        geteslot    sps = spl7();
                        puterec     sps = spl7();
              mch.70.s  _spl7       _spl7:
              sys4.c    alarm       spl7();
                        ftime       spl7();
                        pause       spl7();

splx          bio.c     brelse      splx(s);
                        getbfh      splx(sps);
                        hrelse      splx(sps);
              clock.c   kiltout     splx(s);
                                    splx(s);
                        timeout     splx(s);
              dhdm.c    dmcntrl     splx(sps);
              errlog.c  freeslot    splx(sps);
```

```
                    geterec     splx(sps);
                    geteslot    splx(sps);
                    puterec     splx(sps);
          mch.70.s  _splx       _splx:
          messag.c  msgfree     splx(s);
                    msgremov    splx(s);
                    msgsend     splx(s);
          mx1.c     xcp         splx(s);
                                splx(s);
          mx2.c     mcwrite     splx(s);
                                splx(s);
                    msread      splx(s);
                                splx(s);
                    mxread      splx(s);
                    mxrstrt     splx(s);
                    mxwcontrol  splx(s);
                    scontrol    splx(s);
                    sdata       splx(s);
                    wflush      splx(s);
          slp.c     HASH        splx(s);
                    setrq       splx(s);
                    wakeup      splx(s);
          subr.c    putw        splx(s);
                                splx(s);
          tty.c     flushtty    splx(sps);
                    ttstart     splx(sps);
                                splx(sps);
                    ttyctl      splx(sps);

sprofoff  sys1.c    NCABLK      extern      sprofoff();
                    exit        sprofoff();
          syscb.c   sprofil     sprofoff();
                    sprofoff    sprofoff()

ssig      sys4.c    ssig        ssig()
          sysent.c  nullsys     extern setgid(), getgid(), ssig(), messag(), sysacct();
                                2, &ssig,                        /* 48 = sig */
          sysent.tu.c           extern setgid(), getgid(), ssig(), messag(), sysacct();
                                2, ssig,                         /* 48 = sig */
          sysent.util.          extern setgid(), getgid(), ssig(), messag(), sysacct();
                                2, ssig,                         /* 48 = sig */

stat      sys3.c    _cstat      register struct     stat *bp7;
                    stat        stat()
                    stat1       struct stat ds;
          sysent.c  nullsys     extern chown(), sbreak(), stat(), seek(), getpid(), smount(), sumount();
                                2, &stat,                        /* 18 = stat */
          sysent.tu.c           extern chown(), sbreak(), stat(), seek(), getpid(), smount(), sumount();
                                2, stat,                         /* 18 = stat */
          sysent.util.          extern chown(), sbreak(), stat(), seek(), getpid(), smount(), sumount();
                                2, stat,                         /* 18 = stat */
          tty.c     ttydst      ttydst(cp, acsr, stat)
                                if(stat&CARRIER) {

stat1     sys3.c    fstat       stat1(ip, u.u_arg[0],fp);
                    stat        stat1(ip, u.u_arg[1],NULL);
```

```
                    stat1       stat1(ip, ub, fp)

stime       sys4.c  stime       stime()
            sysent.c nullsys    extern setuid(), getuid(), stime(), ptrace(), alarm(), fstat(), pause();
                                0, &stime,                  /* 25 = stime */
            sysent.tu.c         extern setuid(), getuid(), stime(), ptrace(), alarm(), fstat(), pause();
                                0, stime,                   /* 25 = stime */
            sysent.util.        extern setuid(), getuid(), stime(), ptrace(), alarm(), fstat(), pause();
                                0, stime,                   /* 25 = stime */

stop        sig.c   psig        stop();
                    stop        stop()

stray       mch.70.s tracet         jsr     r0,call1; jmp _stray
            trap.c  stray       stray(junk, sp, r1, addr, r0, pc, ps)

stty        sysent.c nullsys    extern utime(), stty(), gtty(), saccess(), nice(), sync(), kill();
                                1, &stty,                   /* 31 = stty */
            sysent.tu.c         extern utime(), stty(), gtty(), saccess(), nice(), sync(), kill();
                                1, stty,                    /* 31 = stty */
            sysent.util.        extern utime(), stty(), gtty(), saccess(), nice(), sync(), kill();
                                1, stty,                    /* 31 = stty */
            tty.c   stty        stty()

subyte      mch.70.s _subyte    _subyte:
            subr.c  passc       if(id?suibyte(u.u_base, c):subyte(u.u_base, c) < 0) {
            sys1.c  exece           subyte((caddr_t)ucp++,(c=xgetc(paddr(bp)+(unsigned)cp)));

suibyte     mch.70.s _suibyte   _suibyte:
            subr.c  passc       if(id?suibyte(u.u_base, c):subyte(u.u_base, c) < 0) {

suiword     main.c  main        if(suiword(0, 0) < 0)
            mch.70.s _suiword   _suiword:
            sig.c   procxmt     i = suiword(ipc.ip_addr, 0);
                                suiword(ipc.ip_addr, ipc.ip_data);
            tty.c   cinit       suiword(&((ccp&017777)->c_next), xcfreelist);

sumount     sys3.c  sumount     sumount()
            sysent.c nullsys    extern chown(), sbreak(), stat(), seek(), getpid(), smount(), sumount();
                                1, &sumount,                /* 22 = umount */
            sysent.tu.c         extern chown(), sbreak(), stat(), seek(), getpid(), smount(), sumount();
                                1, sumount,                 /* 22 = umount */
            sysent.util.        extern chown(), sbreak(), stat(), seek(), getpid(), smount(), sumount();
                                1, sumount,                 /* 22 = umount */

sureg       main.c  estabur     sureg();
                    sureg       sureg()
            slp.c   expand      sureg();
                    swtch       sureg();

suser       acct.c  sysacct     if (suser()) {
            fio.c   owner       if (suser())
                    suser       suser()
            messag.c messag     &&  !suser() )
            sys2.c  link        if((fmt==IFDIR || fmt==IFLDR) && u.u_dbuf[0]!='.' && !suser()) {
                    mknod       if(suser()) {
```

```
          sys3.c      getmdev     if(!suser())
          sys4.c      chroot      if (!suser())
                      nice        if(n < -1 && !suser())
                      setgid      if(u.u_rgid == gid.lobyte || suser()) {
                      setuid      if(u.u_ruid == uid.lobyte || suser()) {
                      stime       if(suser()) {
                      unlink        && u.u_dbuf[0]!='.' && !suser())
          syscb.c     lock        if(!suser())
                      syscb       if(!suser())
          tty.c       ttyioctl    if (!suser())

suword    dmc11.c     dmcioctl    suword(uaddr++,*saddr);
          mch.70.s    _suword     _suword:
          messag.c    msgrecv     if(suword((unsigned)u.u_arg[2], mhd->mq_sender) < 0 ||
                                    suword((unsigned)u.u_arg[2]+2, mhd->mq_type) < 0)
          rx.c        rxioctl     suword( addr, (rx2_density[minor(dev)]==RX2_SINGLE)?1:2 );
          sig.c       procxmt     if (suword(ipc.ip_addr, 0) < 0)
                                  suword(ipc.ip_addr, ipc.ip_data);
                      psig        suword(n+2, u.u_ar0[R0]);
                                  suword(n, u.u_ar0[R7]);
          sys1.c      exece       suword((caddr_t)ap, na-ne);
                                  suword((caddr_t)ap, ucp);
          sys4.c      times       suword(u.u_arg[0], *p++);
          tty.c       gtty        suword(up, *vp++);
                                  suword(++up, *vp++);
                                  suword(++up, *vp++);
          vp.c        vpioctl     suword(addr, vp->vp_state);

swap      bio.c       swap        swap(blkno, coreaddr, count, rdflg)
          slp.c       swapin      swap(p->p_addr, a, p->p_size, B_READ);
          text.c      xccdec      swap(xp->x_daddr,xp->x_caddr,xp->x_size,B_WRITE);
                      xexpand     swap(xp->x_daddr, xp->x_caddr, xp->x_size, B_READ);
                      xswap       swap(a, rp->p_addr, os, B_WRITE);
                      xswapin     swap(xp->x_daddr,x,xp->x_size,B_READ);

swapin    slp.c       sched       if (swapin(rp))
                      shuffle     swapin(pp);
                      swapin      swapin(pp)

swtch     sig.c       stop        swtch();
          slp.c       HASH        swtch();
                                  swtch();
                      qswtch      swtch();
                      swtch       swtch()
          sys1.c      exit        swtch();

syioctl   conf.70.c   _           &syioctl,   &nulldev,       0,
                      nodev       extern syopen(), syread(), sywrite(), syioctl();
          sys.c       syioctl     syioctl(dev, com, addr, flag)

sync      sys4.c      sync        sync()
          sysent.c    nullsys     extern utime(), stty(), gtty(), saccess(), nice(), sync(), kill();
                                  0, &sync,                       /* 36 = sync */
          sysent.tu.c             extern utime(), stty(), gtty(), saccess(), nice(), sync(), kill();
                                  0, sync,                        /* 36 = sync */
          sysent.util.            extern utime(), stty(), gtty(), saccess(), nice(), sync(), kill();
```

```
                                0, sync,                        /* 36 = sync */

syopen          conf.70.c       _       &syopen,    &nulldev,       &syread,        &sywrite,       /*sy*/
                                nodev   extern syopen(), syread(), sywrite(), syioctl();
                sys.c           syopen  syopen(dev, flag)

syread          conf.70.c       _       &syopen,    &nulldev,       &syread,        &sywrite,       /*sy*/
                                nodev   extern syopen(), syread(), sywrite(), syioctl();
                sys.c           syread  syread(dev)

sysacct         acct.c          _       sysacct() { nodev(); }
                                sysacct sysacct()
                sysent.c        nullsys extern setgid(), getgid(), ssig(), messag(), sysacct();
                                1, &sysacct,                    /* 51 = turn acct off/on */
                sysent.tu.c             extern setgid(), getgid(), ssig(), messag(), sysacct();
                sysent.util.            extern setgid(), getgid(), ssig(), messag(), sysacct();

syscb           syscb.c         syscb   syscb()
                sysent.c        nullsys extern syscb(), chroot(), umask(), event();
                                3, &syscb,                      /* 45 = syscb (tiu on research) */
                sysent.tu.c             extern syscb(), chroot(), umask(), event();
                                3, syscb,                       /* 45 = syscb (tiu on research) */
                sysent.util.            extern syscb(), chroot(), umask(), event();
                                3, syscb,                       /* 45 = syscb (tiu on research) */

sywrite         conf.70.c       _       &syopen,    &nulldev,       &syread,        &sywrite,       /*sy*/
                                nodev   extern syopen(), syread(), sywrite(), syioctl();
                sys.c           sywrite sywrite(dev)

tcommand        tm.c            tabinit ds = tcommand(unit, NOP);
                                tcommand tcommand(unit, com)
                                tmclose tcommand(unit, WEOF);
                                        tcommand(unit, WSOF);
                                        tcommand(unit, REW);
                                        tcommand(unit, SREV);
                                        tcommand(unit, NOP);

tecinput        tec.c           tecinput tecinput(ac, atp)

tecioctl        tec.c           tecioctl tecioctl(tp, flag, nvrow)

tecoutput       tec.c           tecoutput tecoutput(ac, atp)

tell            sys2.c          tell    tell()
                sysent.c        nullsys extern ulimit(), setpgrp(), tell(), dup(), pipe(), times(), profil();
                                0, &tell,                       /* 40 = tell */
                sysent.tu.c             extern ulimit(), setpgrp(), tell(), dup(), pipe(), times(), profil();
                                0, tell,                        /* 40 = tell */
                sysent.util.            extern ulimit(), setpgrp(), tell(), dup(), pipe(), times(), profil();
                                0, tell,                        /* 40 = tell */

texinput        tex.c           texinput texinput(ac, atp)

texioctl        tex.c           texioctl texioctl(tp, flag, nvrow)

texoutput       tex.c           texoutput texoutput(ac, atp)
```

```
textlock    syscb.c     lock        if((u.u_lock&(PROCLOCK|TXTLOCK))  || textlock() == 0)
                                    textlock();
                        textlock    textlock()

timeout     alarm.c     almreset    timeout(almreset,1,ALDELAY);
                                    timeout(almreset,0,ALPERIOD);
            clock.c     timeout     timeout(fun, arg, tim)
            dh.c        dhcntrl     timeout(&dhscan,0,DHSCNRAT);
                        dhscan      timeout(&dhscan, 0, DHSCNRAT);
                        dhstart           timeout(dhrstrt,tp,(cnt&0177)+chrdelay[tp->t_speeds.hibyte&017]);
            dj.c        djcntrl     timeout(&jscan, 0, 4);
                        djscan      timeout(djscan, 0, 4);
                        djxint      timeout(ttrstrt, tp, (c&0177)+1);
            dmc11.c     dmcioctl    timeout(dmcinit,dev,DELAY);
            dn.c        dnwrite     timeout(wakeup,dnadr,SEC2);
                                    timeout(wakeup,dnadr,DELAY);
                                    timeout(wakeup,dnadr,SEC1);
            dz.c        dzcntrl     timeout(&dzscan,0,DSRATE);
                        dzscan      timeout(dzscan,0,DSRATE);
                        dzxint      timeout(dzrstrt,tp,(c & 0177)+chrdelay[tp->t_speeds
            hf.c        hfdst       timeout(ttrstrt,tp,0);
                                    timeout(ttrstrt,tp,0);
                        hfxint      timeout(ttrstrt,tp,PAUSE);
                                    timeout(hftrnrd,tp,PAUSE);
                                    timeout(ttrstrt,tp,PAUSE);
                                    timeout(ttrstrt,tp,PAUSE);
            hps.c       hp_pwrfa    timeout(hp_pwrup, 0, 600);
                        tabinit     timeout(hp_pwrup, 0, 7200);  /* Two minutes */
            hs.c                    timeout(hs_pwrup, 0, 7200);
            jy.c        jyopen      timeout(jypoll,0,2);
                        jypoll      timeout(jypoll,0,2);
            lp.c        lpstart     timeout(lprstrt,0,LPDELAY);
            lpm.c                   timeout(lprstrt,lpr,LPDELAY);
            rop.c       rodst       timeout(ttrstrt,tp,0);
            trans.c     trsread     timeout(&trswake, tp, tp->ts_quanta*60);
            ttdma.c     ttxdma      timeout(ttrstrt, tp,
            tty.c       ttstart     timeout(ttrstrt, tp, c&0177);

times       sys4.c      times       times()
            sysent.c    nullsys     extern ulimit(), setpgrp(), tell(), dup(), pipe(), times(), profil();
                                    1, &times,                      /* 43 = times */
            sysent.tu.c             extern ulimit(), setpgrp(), tell(), dup(), pipe(), times(), profil();
                                    1, times,                       /* 43 = times */
            sysent.util.            extern ulimit(), setpgrp(), tell(), dup(), pipe(), times(), profil();
                                    1, times,                       /* 43 = times */

tmclose     tm.c        tmclose     tmclose(dev, flag)

tmintr      tm.c        tmstart     tmintr()

tmopen      tm.c        tabinit     tmopen(dev, flag)

tmphys      tm.c        tmstart     tmphys(dev);
                                    tmphys(dev);
                                    tmphys(dev)
```

```
tmread      tm.c        tmstart     tmread(dev)

tmstart     tm.c        tabinit     tmstart();
                        tmstart     tmstart()
                                    tmstart();
                                    tmstart();
                        tmstrategy  tmstart();

tmstrategy  tm.c        tcommand    tmstrategy(bp);
                        tmstart     physio(tmstrategy, dev, B_READ, 0);
                                    physio(tmstrategy, dev, B_WRITE, 0);
                        tmstrategy  tmstrategy(bp)

tmwrite     tm.c        tmstart     tmwrite(dev)

trap        mch.70.s    trap            jsr     r0,call1; jmp _trap
            trap.c                  trap(dev, sp, r1, nps, r0, pc, ps)

trap1       trap.c      trap        trap1(callp->call);
                        trap1       trap1(f)

trsinput    trans.c     trsinput    trsinput(ac, atp)

trsioctl    trans.c     trsioctl    trsioctl(com, tp, addr, flag)

trsread     trans.c     trsread     trsread(atp)

trswake     trans.c     trsread     extern trswake();
                                    timeout(&trswake, tp, tp->ts_quanta*60);
                                    killtout(&trswake, tp);
                        trswake     trswake(tp)

trswrite    trans.c     trswrite    trswrite(tp)

trsxint     trans.c     trsxint     trsxint(tp, flag)

ttdvscn     ds40.c      ds40outp    ttdvscn(atp);
            hp45.c      hp45outp    ttdvscn(atp);
            tec.c       tecoutput   ttdvscn(atp);
            tex.c       texoutput   ttdvscn(atp);
            tty.c       ttdvscn     ttdvscn(atp)
            vt61.c      vt61output  ttdvscn(atp);

ttioccomm   dh.c        dhioctl     else if(ttioccomm(cmd, tp, addr, dev)) {
            dj.c        djioctl     else if (ttioccomm(cmd, tp, addr, dev) == 0) {
            dz.c        dziоctl     } else if(ttioccomm(cmd, tp, addr, dev)) {
            ioctl.c     ttioccomm   ttioccomm(com, tp, addr, dev)
            kl.c        klioctl     } else if (ttioccomm(cmd, tp, addr, dev) == 0)

ttread      conf.70.c   ttread      extern ttread(), ttyinput(), ttwrite(), ttxint(), ttyioctl(), ttydst();
                                    &ttread,    &ttyinput,      &ttwrite,       &ttxint,
            tty.c                   ttread(tp)

ttrstrt     dh.c        dhrstrt     ttrstrt(tp);
            dj.c        djxint      extern ttrstrt();
```

```
                                    timeout(ttrstrt, tp, (c&0177)+1);
            dz.c          dzrstrt   ttrstrt(tp);
            hf.c          hfdst     extern ttrstrt();
                                    timeout(ttrstrt,tp,0);
                                    timeout(ttrstrt,tp,0);
                          hfxint    extern hftrnxd(),ttrstrt();
                                    timeout(ttrstrt,tp,PAUSE);
                                    timeout(ttrstrt,tp,PAUSE);
                                    timeout(ttrstrt,tp,PAUSE);
            rop.c         rodst     extern ttrstrt();
                                    timeout(ttrstrt,tp,0);
                                    ttrstrt(tp);
            ttdma.c       ttxdma    extern ttrstrt();
                                    timeout(ttrstrt, tp,
            tty.c         ttrstrt   ttrstrt(tp)
                          ttstart   timeout(ttrstrt, tp, c&0177);

ttstart     hp45.c        hp45inpu  ttstart(tp);
            ioctl.c       ttioccomm ttstart(tp);
                                    ttstart(tp);
            kl.c          klxint    ttstart(tp);
            trans.c       trswrite  ttstart(tp);
                                    ttstart(tp);
                                    ttstart(tp);
            tty.c         canon     ttstart(tp);
                          lsgtty    ttstart(tp);
                          ttrstrt   ttstart(tp);
                          ttstart   ttstart(tp)
                          ttwrite   ttstart(tp);
                                    ttstart(tp);
                          ttxint    ttstart(spytp);
                          ttyinput  ttstart(tp);
                                    ttstart(tp);
                                    ttstart(tp);
            vs.c          vsdst     ttstart(tp);
                          vswrite   ttstart(tp);
                                    ttstart(tp);
            vt100.c       vt100ioctl ttstart(tp);

ttuvscn     ds40.c        ds40outp  ttuvscn(atp);
            hp45.c        hp45outp  ttuvscn(atp);
            tec.c         tecoutput ttuvscn(atp);
            tex.c         texoutput ttuvscn(atp);
            tty.c         ttuvscn   ttuvscn(atp)
            vt61.c        vt61output ttuvscn(atp);

ttwrite     conf.70.c     ttread    extern ttread(), ttyinput(), ttwrite(), ttxint(), ttyioctl(), ttydst();
                                    &ttread,    &ttyinput,      &ttwrite,       &ttxint,
            hf.c          hfwrite   ttwrite(atp);
            tty.c         ttwrite   ttwrite(tp)

ttxdma      conf.70.c     ttread    extern ttyopen(), ttyclose(), ttxdma();
                                    &ttxdma,
                                    &ttxdma,
            ttdma.c       ttxdma    ttxdma(atp, adp)
```

```
ttxint      conf.70.c   ttread      extern ttread(), ttyinput(), ttwrite(), ttxint(), ttyioctl(), ttydst();
                                    &ttread,    &ttyinput,      &ttwrite,       &ttxint,
            tty.c       ttxint      ttxint(tp, flag)

ttyclose    conf.70.c   ttread      extern ttyopen(), ttyclose(), ttxdma();
                                    &ttyioctl,  &ttydst,        &ttyopen,       &ttyclose,
                                    vsioctl,    &vsdst,         &vsopen,        &ttyclose,
            tty.c       ttyclose    ttyclose(dev, tp)

ttyctl      hp45.c      hp45inpu    ttyctl(c, tp);
            tty.c       hqrelse     ttyctl(LCA, tp, colsave, rowsave);
                        ttdvscn     ttyctl(VHOME, tp);
                                    ttyctl(IL, tp);
                        ttuvscn     ttyctl(VHOME, tp);
                                    ttyctl(DL, tp);
                                    ttyctl(LCA, tp, 0, tp->t_lrow);
                        ttwrite     ttyctl(c, tp, col, row);
                                    ttyctl(LCA, tp, col, row);
                        ttyctl      ttyctl(ac, tp, acol, arow)
                        ttyoutl     ttyctl(UVSCN, tp);
                                    ttyctl(NL, tp);
                                    ttyctl(VHOME, tp);
                                    ttyctl(DL, tp);
                                    ttyctl(LCA, tp, 79, tp->t_lrow-1);
                                    ttyctl(LCA, tp, 0, tp->t_row+1);

ttydst      conf.70.c   ttread      extern ttread(), ttyinput(), ttwrite(), ttxint(), ttyioctl(), ttydst();
                                    &ttyioctl,  &ttydst,        &ttyopen,       &ttyclose,
            tty.c       ttydst      ttydst(tp, acsr, stat)

ttyinput    conf.70.c   ttread      extern ttread(), ttyinput(), ttwrite(), ttxint(), ttyioctl(), ttydst();
                                    &ttread,    &ttyinput,      &ttwrite,       &ttxint,
            hf.c        hfrint      ttyinput(achar,tp);
            tty.c       ttyinput    ttyinput(ac, tp)

ttyioctl    conf.70.c   ttread      extern ttread(), ttyinput(), ttwrite(), ttxint(), ttyioctl(), ttydst();
                                    &ttyioctl,  &ttydst,        &ttyopen,       &ttyclose,
            tty.c       ttyioctl    ttyioctl(com, tp, addr, flag)

ttyopen     conf.70.c   ttread      extern ttyopen(), ttyclose(), ttxdma();
                                    &ttyioctl,  &ttydst,        &ttyopen,       &ttyclose,
            rop.c       roopen      ttyopen(tp);
            tty.c       ttyopen     ttyopen(tp)

ttyoutl     tty.c       hqrelse     ttyoutl(c, tp);
                        ttwrite     ttyoutl(c, tp);
                        ttyoutl     ttyoutl(ac, tp)
                                    ttyoutl(' ', tp);
                                    ttyoutl('\\', tp);
                                    ttyoutl('\r', tp);
                                    ttyoutl('\r', tp);
                        ttyoutput   ttyoutl(c, tp);

ttyoutput   ds40.c      ds40inpu    ttyoutput('\n',atp);
            tty.c       ttyinput    ttyoutput(c, tp);
                                    ttyoutput(c, tp);
```

```
                    ttyoutput   ttyoutput(ac, tp)

tunlock     syscb.c lock        if((u.u_lock&PROCLOCK) || tunlock() == 0)
                    punlock     tunlock();
                    tunlock     tunlock()

uchar       acct.c  sysacct     extern uchar();
                                if ((ip = namei(&uchar, 0))==NULL)
            fio.c   owner       extern uchar();
                                if ((ip = namei(uchar, 0)) == NULL)
            maus.c  maus        extern uchar();
                                if((ip = namei(&uchar, 0)) == NULL)
            mx1.c   mpxchan     extern mxopen(), mcread(), uchar();
                                ip = namei(uchar,1);
            mx2.c   mpxname     c = uchar();
            nami.c  uchar       uchar()
            sys1.c  gethead     extern int uchar();
                                if ((ip = namei(uchar, 0)) == NULL)
            sys2.c  creat       extern uchar;
                                ip = namei(&uchar, 1);
                    link        extern uchar;
                                ip = namei(&uchar, 0);
                                xp = namei(&uchar, 1);
                    mknod       extern uchar;
                                ip = namei(&uchar, 1);
                    open        extern uchar;
                                ip = namei(&uchar, 0);
                    saccess     extern uchar;
                                ip = namei(&uchar, 0);
            sys3.c  getmdev     extern uchar;
                                ip = namei(&uchar, 0);
                    smount      extern uchar;
                                ip = namei(&uchar, 0);
                    stat        extern uchar;
                                ip = namei(&uchar, 0);
            sys4.c  chdir       extern uchar;
                                ip = namei(&uchar, 0);
                    chroot      extern uchar;
                                ip = namei(&uchar, 0);
                    unlink      extern uchar;
                                pp = namei(&uchar, 2);
            syscb.c syscb       while((c = uchar()) != -1) {

ufalloc     fio.c   falloc      if ((i = ufalloc(0)) < 0)
                    ufalloc     ufalloc(i)
            mx1.c   mxfalloc    i = ufalloc(0);
            sys3.c  dup         if (i > NOFILE || (i = ufalloc(i)) < 0)
                    fcntl       if (i < 0 || i > NOFILE || (i = ufalloc(i)) < 0)

ulimit      sys4.c  ulimit      ulimit()
            sysent.c nullsys    extern ulimit(), setpgrp(), tell(), dup(), pipe(), times(), profil();
                                2, &ulimit,                     /* 38 = ulimit */
            sysent.tu.c         extern ulimit(), setpgrp(), tell(), dup(), pipe(), times(), profil();
                                2, ulimit,                      /* 38 = ulimit */
            sysent.util.        extern ulimit(), setpgrp(), tell(), dup(), pipe(), times(), profil();
                                2, ulimit,                      /* 38 = ulimit */
```

```
umask       sys4.c      umask       umask()
            sysent.c    nullsys     extern syscb(), chroot(), umask(), event();
                                    1, &umask,                  /* 60 = umask */
            sysent.tu.c             extern syscb(), chroot(), umask(), event();
                                    1, umask,                   /* 60 = umask */
            sysent.util.            extern syscb(), chroot(), umask(), event();
                                    1, umask,                   /* 60 = umask */

unlink      sys4.c      unlink      unlink()
            sysent.c    nullsys     extern creat(), link(), unlink(), exec(), chdir(), gtime(), mknod(), chmod();
                                    1, &unlink,                 /* 10 = unlink */
            sysent.tu.c             extern creat(), link(), unlink(), exec(), chdir(), gtime(), mknod(), chmod();
                                    1, unlink,                  /* 10 = unlink */
            sysent.util.            extern creat(), link(), unlink(), exec(), chdir(), gtime(), mknod(), chmod();
                                    1, unlink,                  /* 10 = unlink */

update      alloc.c     getfs       update();
                        update      update()
            prf.c       panic       update();
            sys3.c      smount      update();
                        sumount     update();
            sys4.c      sync        update();

utime       sys4.c      utime       utime()
            sysent.c    nullsys     extern utime(), stty(), gtty(), saccess(), nice(), sync(), kill();
                                    2, &utime,                  /* 30 = utime */
            sysent.tu.c             extern utime(), stty(), gtty(), saccess(), nice(), sync(), kill();
                                    2, utime,                   /* 30 = utime */
            sysent.util.            extern utime(), stty(), gtty(), saccess(), nice(), sync(), kill();
                                    2, utime,                   /* 30 = utime */

utssys      sysent.c    nullsys     extern utssys();
                                    1, &utssys,                 /* 57 = utssys */
            utssys.c    utssys      utssys()

vpclose     conf.70.c   _           &vpopen,    &vpclose,       &nodev,         &vpwrite,       /*lp*/
                        nodev       extern vpopen(), vpclose(), vpwrite(), vpioctl();
            vp.c        vpclose     vpclose(dev)

vpintr      vp.c        vpintr      vpintr(dev)

vpioctl     conf.70.c   _           &vpioctl,   &nulldev,       0,
                        nodev       extern vpopen(), vpclose(), vpwrite(), vpioctl();
            vp.c        vpioctl     vpioctl(dev, cmd, addr, flag)

vpopen      conf.70.c   _           &vpopen,    &vpclose,       &nodev,         &vpwrite,       /*lp*/
                        nodev       extern vpopen(), vpclose(), vpwrite(), vpioctl();
            vp.c        vpopen      vpopen(dev)

vpready     vp.c        vpclose     vpready(dev);
                        vpioctl     vpready(dev);
                                    vpready(dev);
                        vpopen      if (vpready(dev)<0) {
                        vpready     vpready(dev)
                        vpwrite     if (vpready(dev)<0) {
```

```
vpwrite     conf.70.c   _nodev      &vpopen,    &vpclose,       &nodev,         &vpwrite,       /*lp*/
                                    extern vpopen(), vpclose(), vpwrite(), vpioctl();
            vp.c        vpwrite     vpwrite(dev)

vsdst       conf.70.c   ttread      extern vsread(), vsinput(), vswrite(), vsxint(), vsioctl(), vsdst(), vsopen();
                                    vsioctl,    &vsdst,         &vsopen,        &ttyclose,
            vs.c        vsdst       vsdst(tp, csr, lsr)

vsinput     conf.70.c   ttread      extern vsread(), vsinput(), vswrite(), vsxint(), vsioctl(), vsdst(), vsopen();
                                    &vsread,    &vsinput,       &vswrite,       &vsxint,
            vs.c        vsinput     vsinput(c, tp)

vsioctl     conf.70.c   ttread      extern vsread(), vsinput(), vswrite(), vsxint(), vsioctl(), vsdst(), vsopen();
                                    vsioctl,    &vsdst,         &vsopen,        &ttyclose,
            vs.c        vsioctl     vsioctl(com, tp, addr, flag)

vsopen      conf.70.c   ttread      extern vsread(), vsinput(), vswrite(), vsxint(), vsioctl(), vsdst(), vsopen();
                                    vsioctl,    &vsdst,         &vsopen,        &ttyclose,
            vs.c        vsopen      vsopen(atp)

vsread      conf.70.c   ttread      extern vsread(), vsinput(), vswrite(), vsxint(), vsioctl(), vsdst(), vsopen();
                                    &vsread,    &vsinput,       &vswrite,       &vsxint,
            vs.c        vsread      vsread(tp)

vswrite     conf.70.c   ttread      extern vsread(), vsinput(), vswrite(), vsxint(), vsioctl(), vsdst(), vsopen();
                                    &vsread,    &vsinput,       &vswrite,       &vsxint,
            vs.c        vswrite     vswrite(tp)

vsxint      conf.70.c   ttread      extern vsread(), vsinput(), vswrite(), vsxint(), vsioctl(), vsdst(), vsopen();
                                    &vsread,    &vsinput,       &vswrite,       &vsxint,
            vs.c        vsxint      vsxint(tp, flag)

vt100input  conf.70.c   hp45inpu    extern vt100input(), vt100output(), vt100ioctl();
                                    &vt100input,        &vt100output,   &vt100ioctl,    /*VT100*/
            vt100.c     vt100input  vt100input(ac, atp)

vt100ioctl  conf.70.c   hp45inpu    extern vt100input(), vt100output(), vt100ioctl();
                                    &vt100input,        &vt100output,   &vt100ioctl,    /*VT100*/
            vt100.c     vt100ioctl  vt100ioctl(tp, flag, nvrow)

vt100output conf.70.c   hp45inpu    extern vt100input(), vt100output(), vt100ioctl();
                                    &vt100input,        &vt100output,   &vt100ioctl,    /*VT100*/
            vt100.c     vt100ioctl  vt100output(SVSCN, tp);
                        vt100output vt100output(ac, atp)

vt61input   vt61.c      vt61input   vt61input(ac, atp)

vt61ioctl   vt61.c      vt61ioctl   vt61ioctl(tp, flag, nvrow)

vt61output  vt61.c      vt61output  vt61output(ac, atp)

vtaddpr     vtmon.c     vtaddpr     vtaddpr(pcp)
                        vtprocent   vtaddpr(pcp);
                                    vtaddpr(pcp);
```

```
vtccopy     vtdbg.c     vtdbgst     vtccopy(nulls,msp->ms_name,7);
                                    vtccopy(nulls,msp->ms_val ,7);
                        vtmiscent   if(name) vtccopy(name,msp->ms_name,6);
            vtmon.c     vtinfinit   vtccopy(" procs",inp->in_lproc,7);
                                    vtccopy(" texts",inp->in_ltext,7);
                        vtprinit    vtccopy(
                        vtprocent   vtccopy(nulls,pcp->pr_state,73);
                                    vtccopy(vtflag(prp->p_flag),pcp->pr_flag,2);
                                    vtccopy(pr_states[prp->p_stat],pcp->pr_state,1);
                                    vtccopy(vtflag(prp->p_flag),pcp->pr_flag,2);
                                    vtccopy(vtlstpath(value,valu2),pcp->pr_name,15);
                                    vtccopy(vtflag(prp->p_flag),pcp->pr_flag,2);
                                    vtccopy(pr_states[prp->p_stat],pcp->pr_state,1);
                                    vtccopy(pr_states[prp->p_stat],pcp->pr_state,1);
                                    vtccopy(pr_states[SZOMB],pcp->pr_state,1);
                                    vtccopy(vtflag(prp->p_flag),pcp->pr_flag,2);
                                    vtccopy(pr_frms[value-proc].pr_name, pcp->pr_name, 15);
                                    vtccopy(pr_states[prp->p_stat],pcp->pr_state,1);
                                    vtccopy(vtflag(prp->p_flag),pcp->pr_flag,2);
                        vtstrt      vtccopy("CB-UNIX Release 2.1", txp->hd_unix, sizeof(txp->hd_unix));
                                    vtccopy("0000:00:00", txp->hd_time, sizeof(txp->hd_time));
            vtutil.c    vtccopy     vtccopy(from,to,count)  char *from,*to;  {

vtclose     conf.70.c   _           /* 5*/      &vtopen,        &vtclose,       &nodev,         &vtwrite,       /*dp*/
                        nodev       extern vtopen(), vtclose(), vtwrite();
            vt11.c      vtclose     vtclose(dev)

vtcopy      vt11.c      vtcopy      vtcopy(to,frame)
                        vtwrite     if(vtcopy((caddr_t)((lfrmp=nfrmp)->v_addr),frame) < 0) {

vtdbgin     vtdbg.c     vtdbgin     vtdbgin(framen)  {
            vtmon.c     vtinit      vtdbgin(vframen++);         /* insert debug lines */

vtdbgst     vtdbg.c     vtdbgst     vtdbgst(aypos) {
            vtmon.c     vtstrt      ypos = vtdbgst(ypos-2*VTCHRHD);

vtflag      vtmon.c     vtflag      vtflag(flag)
                        vtprocent   vtccopy(vtflag(prp->p_flag),pcp->pr_flag,2);
                                    vtccopy(vtflag(prp->p_flag),pcp->pr_flag,2);
                                    vtccopy(vtflag(prp->p_flag),pcp->pr_flag,2);
                                    vtccopy(vtflag(prp->p_flag),pcp->pr_flag,2);
                                    vtccopy(vtflag(prp->p_flag),pcp->pr_flag,2);

vtfree      vt11.c      vtfree      vtfree(vfp)
                        vtint       vtfree(vp->v_frmpc);

vtfreelin   vtmon.c     vtaddpr     if(dsp=vtfreelin()) {
                        vtfreelin   vtfreelin()

vtfrmfree   vt11.c      vtclose     vtfrmfree(frp);
                        vtfrmfree   vtfrmfree(frp)
                        vtmapc      vtfrmfree(vfp);
                        vtreplace   vtfrmfree(vt11h[frame].vfd_frm);
                        vtwrite     vtfrmfree(strndx);
                                    vtfrmfree(strndx);
```

```
vtinfinit   vtmon.c     vtinfinit   vtinfinit(ypos)
                        vtstrt      ypos = vtinfinit(ypos);

vtinit      vtll.c      vtopen      vtinit();
            vtmon.c     vtinit      vtinit() {

vtint       vtll.c      vtint       vtint()

vtlpclose   vtlp.c      vtlpclose   vtlpclose(dev) {

vtlpint     vtlp.c      vtlpint     vtlpint() {

vtlpopen    fakevtlp.c  _           vtlpopen(dev, flag) {}
            vtll.c      vtopen      vtlpopen(dev,flag);
            vtlp.c      vtlpclose   vtlpopen(dev,2);
                        vtlpopen    vtlpopen(dev,flag)

vtlpqnt     vtlp.c      vtlpint     vtlpqnt(LPTRKON);
                        vtlpqnt     vtlpqnt(flag) {
                        vtpint      vtlpqnt(LPTRKOF);

vtlpread    vtlp.c      vtlpread    vtlpread() {

vtlstpath   vtmon.c     vtprocent   vtccopy(vtlstpath(value,valu2),pcp->pr_name,15);
            vtutil.c    vtlstpath   vtlstpath(stringp,nchars)  char *stringp;  {

vtmiscent   vtdbg.c     vtmiscent   vtmiscent(flagn,name,value)  char *name;  {

vtnxpc      vtll.c      vtnxpc      vtnxpc(frmp,frame)
                        vtwrite     if((nfrmp = vtnxpc(lfrmp,frame)) == 0) {

vtopen      conf.70.c   _           /* 5*/      &vtopen,        &vtclose,       &nodev,         &vtwrite,       /*dp*/
                        nodev       extern vtopen(), vtclose(), vtwrite();
            vtll.c      vtclose     vtopen(dev,2);
                        vtopen      vtopen(dev,flag)

vtpint      fakevtlp.c  _           vtpint() {}
            vtll.c      vtint       vtpint();
            vtlp.c      vtpint      vtpint()

vtprinit    vtmon.c     vtprinit    vtprinit(ypos)
                        vtstrt      ypos = vtprinit(ypos);

vtprocent   vtmon.c     vtprinit    vtprocent(proc+(pcp-pr_frms),PR_CLEAR);
                        vtprocent   vtprocent(procp,field,value,valu2)  struct proc *procp,*value;  {

vtread      vtll.c      vtread      vtread()

vtreplace   vtll.c      vtreplace   vtreplace(frp,frame)
                        vtwrite     vtreplace(stradr,frame);

vtrmvpr     vtmon.c     vtprocent   vtrmvpr(pcp);
                                    vtrmvpr(pcp);
                        vtrmvpr     vtrmvpr(pcp)

vtsconv     vtmon.c     vtprocent   vtsconv(prp->p_pri,pcp->pr_pri,4,10);
```

```
                                    vtsconv(prp->p_pri,pcp->pr_pri,4,10);
          vtutil.c      vtsconv     vtsconv(val,to,ndig,base)  char *to;  {

vtstrt    vtmon.c       vtinit      vtstrt();
                        vtstrt      vtstrt()

vttinc    vtmon.c       vttinc      vttinc()

vttrans   vt100.c       vt100output vttrans(atp, cp->tpindx, cp->outchar);
                        vttrans     vttrans(atp, ac, aoc)
                                    vttrans(atp, c & 077, 0);

vtvconv   vtdbg.c       vtmiscent   vtvconv(value,msp->ms_val,6,8);
          vtmon.c       vtinfinit   vtvconv(NPROC,inp->in_nproc,3,10);
                                    vtvconv(NTEXT,inp->in_ntext,3,10);
                        vtprocent   vtvconv(prp->p_wchan,pcp->pr_wchan,6,8);
                                    vtvconv(fsig(prp),pcp->pr_sig,2,10);
                                    vtvconv(prp->p_time,pcp->pr_ptim,3,10);
                                    vtvconv(prp->p_ctime,pcp->pr_ctim,3,10);
                                    vtvconv(pcp->p_clktim,pcp->pr_clock,5,10);
                                    vtvconv(prp->p_pgrp,pcp->pr_group,6,10);
                                    vtvconv(prp->p_pid,pcp->pr_pid,5,10);
                                    vtvconv(prp->p_ppid,pcp->pr_ppd,5,10);
                                    vtvconv(prp->p_wchan,pcp->pr_wchan,6,8);
                                    vtvconv(prp->p_ctime,pcp->pr_ctim,3,10);
                                    vtvconv(prp->p_wchan,pcp->pr_wchan,6,8);
                                    vtvconv(prp->p_ctime,pcp->pr_ctim,3,10);
                                    vtvconv(prp->p_pid,pcp->pr_pid,5,10);
                                    vtvconv(prp->p_ppid,pcp->pr_ppd,5,10);
                                    vtvconv(prp->p_time,pcp->pr_ptim,3,10);
                                    vtvconv(prp->p_size,pcp->pr_size,4,8);
                                    vtvconv(prp->p_pgrp,pcp->pr_group,6,10);
                                    vtvconv(fsig(prp),pcp->pr_sig,2,10);
                                    vtvconv(prp->p_wchan,pcp->pr_wchan,6,8);
                                    vtvconv(prp->p_time,pcp->pr_ptim,3,10);
                                    vtvconv(prp->p_size,pcp->pr_size,4,8);
          vtutil.c      vtvconv     vtvconv(val,to,ndig,base)  char *to;  {

vtwrite   conf.70.c     _           /* 5*/      &vtopen,        &vtclose,       &nodev,         &vtwrite,       /*dp*/
                        nodev       extern vtopen(), vtclose(), vtwrite();
          vtll.c        vtwrite     vtwrite(dev)
          vtdbg.c       vtdbgin     vtwrite();
          vtmon.c       vtinit      vtwrite();
                                          vtwrite();
                                    vtwrite();

wait      sys1.c        wait        wait()
          sysent.c      nullsys     extern rexit(), fork(), read(), write(), open(), close(), wait();
                                    0, &wait,                       /*  7 = wait */
          sysent.tu.c               extern rexit(), fork(), read(), write(), open(), close(), wait();
                                    0, wait,                        /*  7 = wait */
          sysent.util.              extern rexit(), fork(), read(), write(), open(), close(), wait();
                                    0, wait,                        /*  7 = wait */

waitloc   clock.c       _           extern char waitloc;
                        clock       if(pc == &waitloc)
```

```
                    sincupc     if (pc == &waitloc) {
             mch.70.s   _waitloc    _waitloc:

wakeup       alloc.c    alloc       wakeup(&fp->s_flock);
                        free        wakeup(&fp->s_flock);
                        getfs       wakeup(p);
                        ialloc      wakeup(&fp->s_ilock);
             bio.c      brelse      wakeup((caddr_t)bp);
                                    wakeup((caddr_t)&bfreelist);
                        hrelse      wakeup((caddr_t)&hfreelist);
                        iodone      wakeup(bp);
                        physio      wakeup(&runin);
                        swap        wakeup((caddr_t)bp);
             clock.c    clock       wakeup(&lbolt);
                                    wakeup(&runin);
             dmc11.c    dmcclr      wakeup(dmp);
                        dmciint     wakeup(dmp);
                        dmcinit     wakeup(dmp);
             dn.c       dnint       wakeup(dnadr);
                        dnwrite     extern wakeup();
                                    timeout(wakeup,dnadr,SEC2);
                                    timeout(wakeup,dnadr,DELAY);
                                    timeout(wakeup,dnadr,SEC1);
                                    wakeup(&(dnadr->dn_datum));
             errlog.c   puterec     wakeup(&err.e_org);
             fio.c      closef      wakeup((caddr_t)ip+1);
                                    wakeup((caddr_t)ip+2);
             hf.c       hfdst       wakeup(tp);
                                    wakeup(tp);
                                    wakeup(tp);
                        hfioctl     wakeup(tp);
                        hfread      wakeup(&tp->t_rawq);
                        hfrint      wakeup(&tp->t_rawq);
                        hfxint      wakeup(&tp->t_outq);
             ht.c       hcommand    wakeup(bp);
             jy.c       jyint       wakeup(jyq);
             kl.c       klxint      wakeup(&tp->t_outq);
             lfh.c      lfrlse      wakeup(&lffflag);
             lp.c       lpint       wakeup(&lp11);
             lpm.c                  wakeup(lp);
             malloc.c   mfree       wakeup(&runin);              /* Wake scheduler when freeing core */
             messag.c   msgfree     wakeup(&msgbase);
                        msgremov    wakeup(&qp->mq_flag);
                        msgsend     wakeup(qp);
             mx1.c      mpxchan     wakeup((caddr_t)cp);
             mx2.c      chwake      wakeup(cp);
                                    wakeup(p); wakeup(++p); wakeup(++p);
                                    wakeup(p); wakeup(++p); wakeup(++p);
                        mcstart     wakeup(q);
                        mcwrite     wakeup((caddr_t)q);
                        msread      wakeup((caddr_t)cp);
                                    wakeup((caddr_t)q);
                        mswrite     else                         wakeup((caddr_t)q);
                                    else                         wakeup((caddr_t)q);
                        mxrstrt     wakeup((caddr_t)q+1); else
                                    wakeup((caddr_t)q+2);
```

		mxwcontrol	wakeup(&cp->cy.datq);
		sdata	wakeup((caddr_t)&ngp->g_datq);
	nmpipe.c	npioctl	wakeup((caddr_t)ip+1);
			wakeup((caddr_t)ip+2);
		npopen	wakeup((caddr_t)npp);
	pipe.c	prele	wakeup((caddr_t)ip);
		readp	wakeup((caddr_t)ip+1);
		writep	wakeup((caddr_t)ip+2);
	rh.c	mapfree	wakeup(ubmap);
	rop.c	rodst	wakeup(tp);
			wakeup(tp);
	sig.c	procxmt	wakeup(&ipc);
		ptrace	wakeup(&ipc);
		stop	wakeup(pp);
	slp.c	HASH	wakeup((caddr_t)&runin);
		setrun	wakeup(v);
			wakeup((caddr_t)&runout);
		shuffle	wakeup(&runlock);
		wakeup	wakeup(chan)
	sys1.c	exit	wakeup(&runout);
	sys5.c	event	wakeup(&semi[s]);
			wakeup(&semi[s]);
			wakeup(&semi[s]);
		semafree	wakeup(&semi[i]);
	syscb.c	lock	wakeup(&runout);
	text.c	xswap	wakeup(&runout);
		xunlock	wakeup(xp);
	tm.c	tcommand	wakeup(bp);
	trans.c	trsinput	wakeup(&tp->ts_rawq);
		trswake	wakeup(&tp->ts_rawq);
		trsxint	wakeup((caddr_t)&tp->ts_outq);
	tty.c	flushtty	wakeup(&tp->t_rawq);
			wakeup(&tp->t_outq);
		ttwrite	wakeup(tp);
		ttxint	wakeup((caddr_t)&tp->t_outq);
		ttydst	wakeup(tp);
		ttyinput	wakeup(&tp->t_rawq);
			wakeup(&tp->t_rawq);
	vp.c	vpintr	wakeup(&vp_vp[dev]);
	vs.c	vsdst	wakeup(tp);
		vsinput	wakeup(&tp->t_rawq);
		vsxint	wakeup(&tp->t_outq);
	vt11.c	vtint	wakeup(&vp->v_fmap[c]);
			wakeup(vp);
	vtlp.c	vtlpqnt	wakeup(vp11);
wdir	iget.c	maknode	wdir(ip);
		wdir	wdir(ip)
	sys2.c	link	wdir(ip);
wflush	mx2.c	chdrain	wflush(cp,&cp->cx.datq);
		wflush	wflush(cp,q)
wflushtty	ioctl.c	ttioccomm	wflushtty(tp);
	tty.c	lsgtty	wflushtty(tp);
		ttyclose	wflushtty(tp);

```
                    wflushtty   wflushtty(tp)

write       sys2.c      write       write()
            sysent.c    nullsys     extern rexit(), fork(), read(), write(), open(), close(), wait();
                                    2, &write,                      /*  4 = write */
            sysent.tu.c             extern rexit(), fork(), read(), write(), open(), close(), wait();
                                    2, write,                       /*  4 = write */
            sysent.util.            extern rexit(), fork(), read(), write(), open(), close(), wait();
                                    2, write,                       /*  4 = write */

writei      acct.c      acct        writei(ip);
            iget.c      wdir        writei(u.u_pdir);
            pipe.c      writep      writei(ip);
            rdwri.c     writei      writei(ip)
            sig.c       core        writei(ip);
                                    writei(ip);
            sys2.c      rdwr        writei(ip);
            sys4.c      unlink      writei(pp);

writep      nmpipe.c    npwrite     writep(npp->p_wfp);
            pipe.c      writep      writep(fp)
            sys2.c      rdwr        writep(fp);

xalloc      sys1.c      getxfile    xalloc(ip);
                                    xalloc(ip);
            text.c      xalloc      xalloc(ip)

xccdec      slp.c       shuffle     xccdec(xp);
            syscb.c     tunlock     xccdec(xp);
            text.c      xccdec      xccdec(xp)
                        xfree       xccdec(xp);
                        xswap       xccdec(rp->p_textp);

xcp         mx1.c       mpxchan     cp = xcp(gp, vec.m_arg[0]);
                                    cp = xcp(gp, vec.m_arg[0]);
                                    cp = xcp(gp, vec.m_arg[0]);
                                    cp = xcp(gp, vec.m_arg[0]);
                                    cp = xcp(gp, vec.m_arg[0]);
                        xcp         struct chan *xcp();
            mx2.c       mxwrite     cp = xcp(gp, h.index);
                        xcp         struct chan *xcp(),*addch(),*nextcp();
                                    xcp(gp, x)

xexpand     text.c      xalloc      xexpand(xp);
                                    xexpand(xp);
                        xexpand     xexpand(xp)

xflsh       sys3.c      sumount     xflsh(d);
            text.c      xalloc      if(xflsh(NODEV))
                        xflsh       xflsh(dev)

xfree       sys1.c      exit        xfree();
                        getxfile    xfree();
            text.c      xfree       xfree()

xget        alloc.c     alloc       fp->s_nfree = xget(paddr(bp));
```

```
          mch.70.s    _xget       _xget:
          subr.c      bmap        if ((nb = xget(paddr(bp) + 2*i)) == 0) {
                                  rablock = xget(paddr(bp) + 2*(i+1));

xgetc     mch.70.s    _xgetc      _xgetc:
          sys1.c      exece           subyte((caddr_t)ucp++,(c=xgetc(paddr(bp)+(unsigned)cp)));

xgetl     hps.c       hpintr      long        xgetl();
                                  l = xgetl(adr);
          mch.70.s    _xgetl      _xgetl:

xlock     text.c      xalloc      xlock(xp);
                      xccdec      xlock(xp);
                      xfree       xlock(xp);
                      xlock       xlock(axp)
                      xswapin     xlock(xp);

xput      alloc.c     free        xput(paddr(bp), fp->s_nfree);
          mch.70.s    _xput       _xput:
          subr.c      bmap        xput(paddr(bp) + 2*i, nb);

xputc     mch.70.s    _xputc      _xputc:
          sys1.c      exece       xputc(paddr(bp)+(unsigned)cp, c);

xputl     hps.c       hpintr      xputl(adr, l);
          iget.c      iupdat      xputl(addr, time);
                                  xputl(addr, *tm);
          mch.70.s    _xputl      _xputl:

xswap     slp.c       expand      xswap(p, 1, n);
                      newproc     xswap(rpp, 0, 0);
                      sched       xswap(rp, 1, 0);
                      shuffle     xswap(pp,1,0);
          text.c      xexpand     xswap(u.u_procp, 1, 0);
                      xswap       xswap(p, ff, os)

xswapin   slp.c       shuffle     xswapin(xp);
                      swapin      if(xswapin(p->p_textp) == 0)
          text.c      xswapin     xswapin(axp)

xunlock   text.c      xalloc      xunlock(xp);
                      xccdec      xunlock(xp);
                      xexpand     xunlock(xp);
                                  xunlock(xp);
                      xswapin     xunlock(xp);
                                  xunlock(xp);
                      xunlock     xunlock(axp)

zero      mx2.c       mxclose     zero(gp, sizeof (struct group));
                      zero        zero(s, cc)
```