```
    int status;
    char * arg;         /* Pointer to the argument string */


    time_t theTime;     /* A long integer giving the datetime in secs */


    for (;;) {
        status = read(soc, command, COMMAND_SIZE);
        if (status<=0) {
            if (TRACE) printf("read returned %i, errno=%i\n", status, errno);
            return status;      /* Return and close file */
        }
        command[status]=0;      /* terminate the string */
#ifdef NOT_ASCII
        {
            char * p;
            for (p=command; *p; p++) {
                *p = FROMASCII(*p);
                if (*p == '\n') *p = ' ';
            }
        }
#endif

/*      Log the call:
*/
        if (LogFile) {
            time(&theTime);
            fprintf(LogFile, "%24.24s %s %s",
                ctime(&theTime), HTClientHost, command);
            fflush(LogFile);    /* Actually update it on disk */
            if (TRACE) printf("Log: %24.24s %s %s",
                ctime(&theTime), HTClientHost, command);
        }

        arg=strchr(command,' ');
        if (arg) {
            *arg++ = 0;                         /* Terminate command
        */
            arg = strip(arg);                   /* Strip space
        */

            if (0==strcmp("GET", command)) {    /* Get a file
        */
                keywords=strchr(arg, '?');
                if (keywords) *keywords++=0;    /* Split into two */
                return HTRetrieve(arg, keywords, soc);
            }

        } /* Space found supplied */

        HTWriteASCII(soc, "<h1>HTTP Server v1.02</h1><h2>Test response</h2>\n");
        HTWriteASCII(soc, "\n\nThe (unrecognised) command used was `");
        HTWriteASCII(soc, command);
        HTWriteASCII(soc, "'.\n\n");

        if (TRACE) printf("Return test string written back'\n");
        return 0;               /* End of file - please close socket */
    } /* for */
```

```
} /* handle */


/*      Handle incomming messages
server_loop()
**      -------------------------
**
** On entry:
**
**      timeout         -1 for infinite, 0 for poll, else in units of 10ms
**
** On exit,
**      returns         The status of the operation, <0 if failed.
**
*/
#ifdef __STDC__
PRIVATE int server_loop(void)
#else
PRIVATE int server_loop()
#endif
{
    int tcp_status;             /* <0 if error, in general */
    int timeout = -1;           /* No timeout required but code exists */
    for(;;) {

/*  If it's a master socket, then find a slave:
*/
        if (role == master) {
#ifdef SELECT
            fd_set              read_chans;
            fd_set              write_chans;
            fd_set              except_chans;
            int                 nfound;     /* Number of ready channels */
            struct timeval      max_wait;   /* timeout in form for select() */


            FD_ZERO(&write_chans);          /* Clear the write mask */
            FD_ZERO(&except_chans);         /* Clear the exception mask */

/*  If timeout is required, the timeout structure is set up. Otherwise
**  (timeout<0) a zero is passed instead of a pointer to the struct timeval.
*/
            if (timeout>=0) {
                max_wait.tv_sec = timeout/100;
                max_wait.tv_usec = (timeout%100)*10000;
            }


            for (com_soc=-1; com_soc<0;) {      /* Loop while connections keep coming


/*  The read mask expresses interest in the master channel for incomming
**  connections) or any slave channel (for incomming messages).
*/

/*  Wait for incoming connection or message
*/
```

```
                read_chans = open_sockets;       /* Read on all active channels */
                if (TRACE) printf(
 "Daemon: Waiting for connection or message. (Mask=%x hex, max=%x hex).\n",

                        *(int *)(&read_chans), num_sockets);
                nfound=select(num_sockets, &read_chans,
                    &write_chans, &except_chans,
                    timeout >= 0 ? &max_wait : 0);

                if (nfound<0) return HTInetStatus("select()");
                if (nfound==0) return 0;        /* Timeout */

/*      We give priority to existing connected customers. When there are
**      no outstanding commands from them, we look for new customers.
*/
/*      If a message has arrived on one of the channels, take that channel:
*/
                {
                    int i;
                    for(i=0; i<num_sockets; i++)
                        if (i != master_soc)
                            if (FD_ISSET(i, &read_chans)) {
                            if (TRACE) printf(
                                "Message waiting on socket %i\n", i);
                            com_soc = i;                /* Got one! */
                            break;
                        }
                    if (com_soc>=0) break; /* Found input socket */


                } /* block */

/*  If an incomming connection has arrived, accept the new socket:
*/
                if (FD_ISSET(master_soc, &read_chans)) {


                        CTRACE(tfp, "Daemon: New incomming connection:\n");
                        tcp_status = accept(master_soc,
                                        (struct sockaddr *)&soc_address,
                                        &soc_addrlen);
                        if (tcp_status<0)
                            return HTInetStatus("accept");
                        CTRACE(tfp, "Daemon: Accepted new socket %d\n",
                            tcp_status);
                        FD_SET(tcp_status, &open_sockets);
                        if ((tcp_status+1) > num_sockets)
                                num_sockets=tcp_status+1;


                } /* end if new connection */



            } /* loop on event */

#else   /* SELECT not supported */
```

```
            if (com_soc<0) { /* No slaves: must accept */
                    CTRACE(tfp,

                    "Daemon: Waiting for incomming connection...\n");
                    tcp_status = accept(master_soc,
                                        &rsoc->mdp.soc_tcp.soc_address,
                                        &rsoc->mdp.soc_tcp.soc_addrlen);
                    if (tcp_status<0)
                        return HTInetStatus("accept");
                    com_soc = tcp_status;       /* socket number */
                    CTRACE(tfp, "Daemon: Accepted socket %d\n", tcp_status);
            } /* end if no slaves */


#endif

        }  /* end if master */


/* com_soc is now valid for read */

        {
            struct sockaddr_in addr;
            int namelen = sizeof(addr);
            getpeername(com_soc, (struct sockaddr*)&addr, &namelen);
            strncpy(HTClientHost,
                 inet_ntoa(addr.sin_addr), sizeof(HTClientHost));
        }

/*  Read the message now on whatever channel there is:
*/
        CTRACE(tfp,"Daemon: Reading socket %d from host %s\n",
                com_soc, HTClientHost);

        tcp_status=HTHandle(com_soc);

        if(tcp_status<=0) {                             /* EOF or error */
            if (tcp_status<0) {                         /* error */
                CTRACE(tfp,
                "Daemon: Error %i handling incomming message (errno=%i).\n",
                        tcp_status, errno);
                /* DONT return HTInetStatus("netread");  error */
            } else {
                CTRACE(tfp, "Daemon: Socket %d disconnected by peer\n",
                    com_soc);
            }
            if (role==master) {
                NETCLOSE(com_soc);
                FD_CLR(com_soc, &open_sockets);
            } else {  /* Not multiclient mode */
#ifdef VM
                return -69;
#else
                return -ECONNRESET;
#endif
            }
        } /* end if handler got EOF or error */


    }; /* for loop */
```

```
/*NOTREACHED*/
} /* end server_loop */


/*              Main program
**              ------------
**
**      Options:
**      -v              verify: turn trace output on to stdout

**      -a addr         Use different tcp port number and style
**      -l file         Log requests in ths file
**      -r file         Take rules from this file
**      -R file         Clear rules and take rules from file.
*/
int main(int argc, char*argv[])
{
    int status;
#ifdef RULES
    int rulefiles = 0;          /* Count number loaded */
#endif
    char * addr = "";           /* default address */
    WWW_TraceFlag = 0;                  /* diagnostics off by default */
#ifdef RULES
    HTClearRules();
#endif
    {
        int a;


        for (a=1; a<argc; a++) {


            if (0==strcmp(argv[a], "-v")) {
                WWW_TraceFlag = 1;

            } else if (0==strcmp(argv[a], "-a")) {
                if (++a<argc) addr = argv[a];

#ifdef RULES
            } else if (0==strcmp(argv[a], "-r")) {
                if (++a<argc) {

                    if (HTLoadRules(argv[a]) < 0) exit(-1);
                    rulefiles++;
                }
            } else if (0==strcmp(argv[a], "-R")) {
                rulefiles++;            /* Inhibit rule file load */
#endif
            } else if (0==strcmp(argv[a], "-l")) {
                if (++a<argc) {
                    LogFile = fopen(argv[a], "a");
                    logfile = LogFile;          /* name for WAIS */
                    log_file_name = argv[a];    /* For WAIS code */
                }
                if (!LogFile) {
                    fprintf(stderr,
                        "Can't open log file %s\n", argv[a]);
                    logfile = LogFile = stderr;
                }
```

```
/* Remove because an inet daemon process gets started every request.
                fprintf(LogFile, "\nhttpd: HTTP Daemon %s started.\n",
                        VERSION);
*/
            } /*ifs */
        } /* for */
    } /* scope of a */


#ifdef RULES
    if (rulefiles==0) {          /* No mention */
        if (HTLoadRules(RULE_FILE) < 0) exit(-1);
    }
#endif




    status = do_bind(addr);
    if (status<0) {
        fprintf(stderr, "Daemon: Bad setup: Can't bind and listen on port.\n");
        fprintf(stderr, "     (Possibly server already running, for example).\n");
        exit(status);
    }


    status = server_loop();

    if (status<0) {
        /* printf("Error in server loop.\n");  not error if inetd-started*/
        exit(status);
    }


    exit(0);
    return 0;           /* For gcc */
}
```

```
/*                      Generic Communication Code              HTTCP.c
**                      ==========================
**
**      This code is in common between client and server sides.
*/


#include "HTUtils.h"
#include "tcp.h"                /* Defines SHORT_NAMES if necessary */
#ifdef SHORT_NAMES
#define HTInetStatus            HTInStat
#define HTInetString            HTInStri
#endif


/*      Module-Wide variables
*/

PRIVATE char *hostname=0;               /* The name of this host */


/*      PUBLIC VARIABLES
*/

PUBLIC struct sockaddr_in HTHostAddress;        /* The internet address of the host *
                                        /* Valid after call to HTHostName() */

/*      Encode INET status (as in sys/errno.h)                  inet_status()
**      ------------------
**
** On entry,
**      where           gives a description of what caused the error
**      global errno    gives the error number in the unix way.
**
** On return,
**      returns         a negative status in the unix way.
*/
#ifdef vms
extern int uerrno;      /* Deposit of error info (as perr errno.h) */
extern int vmserrno;    /* Deposit of VMS error info */
extern volatile noshare int errno;  /* noshare to avoid PSECT conflict */
#else
#ifndef errno
extern int errno;
#endif
#endif

#ifndef VM
#ifndef vms
#ifndef NeXT
extern char *sys_errlist[];             /* see man perror on cernvax */
extern int sys_nerr;
#endif
#endif
#endif


/*      Report Internet Error
**      ---------------------
*/
```

```
#ifdef __STDC__
PUBLIC int HTInetStatus(char *where)
#else
PUBLIC int HTInetStatus(where)
    char    *where;
#endif
{
    CTRACE(tfp, "TCP: Error %d in `errno' after call to %s() failed.\n\t%s\n",
            errno,  where,
#ifdef VM
            "(Error number not translated)");    /* What Is the VM equiv? */
#define ER_NO_TRANS_DONE
#endif
#ifdef vms
            "(Error number not translated)");
#define ER_NO_TRANS_DONE
#endif
#ifdef NeXT
            strerror(errno));
#define ER_NO_TRANS_DONE
#endif

#ifndef ER_NO_TRANS_DONE
            errno < sys_nerr ? sys_errlist[errno] : "Unknown error" );
#endif


#ifdef vms
    CTRACE(tfp, "         Unix error number (uerrno) = %ld dec\n", uerrno);
    CTRACE(tfp, "         VMS error (vmserrno)       = %lx hex\n", vmserrno);
#endif
    return -errno;
}


/*      Parse a cardinal value                                  parse_cardinal()
**      ----------------------
**
** On entry,
**      *pp         points to first character to be interpreted, terminated by
**                  non 0:9 character.
**      *pstatus    points to status already valid
**      maxvalue    gives the largest allowable value.
**
** On exit,
**      *pp         points to first unread character
**      *pstatus    points to status updated iff bad
*/
#ifdef __STDC__
PUBLIC unsigned int HTCardinal(int *pstatus,
        char            **pp,
        unsigned int    max_value)
#else
PUBLIC unsigned int HTCardinal(pstatus, pp, max_value)
   int                  *pstatus;
   char                 **pp;
   unsigned int         max_value;
#endif
{
    int   n;
```

```
    if ( (**pp<'0') || (**pp>'9')) {        /* Null string is error */
        *pstatus = -3;  /* No number where one expeceted */
        return 0;
    }

    n=0;
    while ((**pp>='0') && (**pp<='9')) n = n*10 + *((*pp)++) - '0';

    if (n>max_value) {
        *pstatus = -4;  /* Cardinal outside range */
        return 0;
    }

    return n;
}


/*      Produce a string for an inernet address
**      ---------------------------------------
**
** On exit,
**      returns a pointer to a static string which must be copied if
**              it is to be kept.
*/
#ifdef __STDC__
PUBLIC const char * HTInetString(struct sockaddr_in* sin)
#else
PUBLIC char * HTInetString(sin)
    struct sockaddr_in *sin;
#endif
{
    static char string[16];
    sprintf(string, "%d.%d.%d.%d",
            (int)*((unsigned char *)(&HTHostAddress.sin_addr)+0),
            (int)*((unsigned char *)(&HTHostAddress.sin_addr)+1),
            (int)*((unsigned char *)(&HTHostAddress.sin_addr)+2),
            (int)*((unsigned char *)(&HTHostAddress.sin_addr)+3));
    return string;
}


/*      Parse an internet node address and port
**      ---------------------------------------
**
** On entry,
**      str     points to a string with a node name or number,
**              with optional trailing colon and port number.
**      sin     points to the binary internet address field.
**
** On exit,
**      *sin    is filled in. If no port is specified in str, that
**              field is left unchanged in *sin.
*/
#ifdef __STDC__
PUBLIC int HTParseInet(struct sockaddr_in* sin, const char *str)
#else
PUBLIC int HTParseInet(sin, str)
    struct sockaddr_in *sin;
    char *str;
#endif
```

```
{
    char *port;
    char host[256];
    struct hostent  *phost;     /* Pointer to host - See netdb.h */
    strcpy(host, str);          /* Take a copy we can mutilate */


/*      Parse port number if present
*/
    if (port=strchr(host, ':')) {
        *port++ = 0;            /* Chop off port */
        if (port[0]>='0' && port[0]<='9') {
            sin->sin_port = htons(strtol(port, (char**)0 , 10));
        } else {
#ifdef SUPPRESS         /* 1. crashes!?!.  2. Not recommended */
            struct servent * serv = getservbyname(port, (char*)0);
            if (serv) sin->sin_port = serv->s_port;
            else if (TRACE) printf("TCP: Unknown service %s\n", port);
#endif
        }
    }

/*      Parse host number if present
*/
    if (*host>='0' && *host<='9') {   /* Numeric node address: */
        sin->sin_addr.s_addr = inet_addr(host); /* See arpa/inet.h */

    } else {                /* Alphanumeric node name: */
#ifdef MVS      /* Oustanding problem with crsh in MVS gethostbyname */
        if(TRACE)printf("HTTCP: Calling gethostbyname(%s)\n", host);
#endif
        phost=gethostbyname(host);      /* See netdb.h */
#ifdef MVS
        if(TRACE)printf("HTTCP: gethostbyname() returned %d\n", phost);
#endif
        if (!phost) {
            if (TRACE) printf(
                    "HTTPAccess: Can't find internet node name `%s'.\n",host);
            return -1;  /* Fail? */
        }
        memcpy(&sin->sin_addr, phost->h_addr, phost->h_length);
    }

    if (TRACE) printf(
        "TCP: Parsed address as port %d, IP address %d.%d.%d.%d\n",
                (unsigned int)ntohs(sin->sin_port),
                (int)*((unsigned char *)(&sin->sin_addr)+0),
                (int)*((unsigned char *)(&sin->sin_addr)+1),
                (int)*((unsigned char *)(&sin->sin_addr)+2),
                (int)*((unsigned char *)(&sin->sin_addr)+3));

    return 0;   /* OK */
}


/*      Derive the name of the host on which we are
**      -------------------------------------------
**
```

```
*/
#ifdef __STDC__
PRIVATE void get_host_details(void)
#else
PRIVATE void get_host_details()
#endif

#ifndef MAXHOSTNAMELEN
#define MAXHOSTNAMELEN 64               /* Arbitrary limit */
#endif

{
    char name[MAXHOSTNAMELEN+1];        /* The name of this host */
    struct hostent * phost;             /* Pointer to host -- See netdb.h */
    int namelength = sizeof(name);

    if (hostname) return;               /* Already done */
    gethostname(name, namelength);      /* Without domain */
    CTRACE(tfp, "TCP: Local host name is %s\n", name);
    phost=gethostbyname(name);          /* See netdb.h */
    if (!phost) {
        if (TRACE) printf(
                "TCP: Can't find my own internet node address for `%s'!!\n",
                name);
        return;  /* Fail! */
    }
    StrAllocCopy(hostname, phost->h_name);
    memcpy(&HTHostAddress, &phost->h_addr, phost->h_length);
    if (TRACE) printf("     Name server says that is `%s' = %s\n",
            hostname, HTInetString(&HTHostAddress));
}

#ifdef __STDC__
PUBLIC const char * HTHostName(void)
#else
PUBLIC char * HTHostName()
#endif
{
    get_host_details();
    return hostname;
}
```

```
/*      Macros for general use                                          HTUtils.h
*/
/*      Allocate a new copy of a string:
*/
#define StrAllocCopy(new,old) \
 if (new) free(new); (new) = (char *)malloc(strlen(old)+1); strcpy((new),(old))
/* extern void *malloc(size_t size); */

/*      Debug message control.
*/
#ifdef DEBUG
#define TRACE (WWW_TraceFlag)
#define PROGRESS(str) printf(str)
extern int WWW_TraceFlag;
#else
#define TRACE 0
#define PROGRESS(str) /* nothing for now */
#endif
#define CTRACE if(TRACE)fprintf
#define tfp stdout

/*      Standard C library for malloc() etc
*/
#ifdef vax
#ifdef unix
#define ultrix  /* Assume vax+unix=ultrix */
#endif
#endif

#ifndef VMS
#ifndef ultrix
#ifdef NeXT
#include <libc.h>       /* NeXT */
#endif
#include <stdlib.h>     /* ANSI */
#else
#include <malloc.h>     /* ultrix */
#include <memory.h>
#endif

#else                           /* VMS */
#include <stdio.h>
#include <ctype.h>
#endif

#define PUBLIC                  /* Accessible outside this module     */
#define PRIVATE static          /* Accessible only within this module */

#ifdef __STDC__
#define CONST const             /* "const" only exists in STDC */
#else
#define CONST
#endif

/* Note: GOOD and BAD are already defined (differently) on RS6000 aix */
/* #define GOOD(status) ((status)&1)     VMS style status: test bit 0         */
/* #define BAD(status)  (!GOOD(status))  Bit 0 set if OK, otherwise clear   */

#ifndef BOOLEAN_DEFINED
typedef char    BOOLEAN;                /* Logical value */
```

```
#ifndef TRUE
#define TRUE    (BOOLEAN)1
#define FALSE   (BOOLEAN)0
#endif
#define BOOLEAN_DEFINED
#endif

#ifndef BOOL
#define BOOL BOOLEAN
#endif
#ifndef YES
#define YES (BOOLEAN)1
#define NO (BOOLEAN)0
#endif

#define TCP_PORT 2784           /* Arbitrary value -- should be allocated */

/*      Is character c white space? */

#ifndef NOT_ASCII
#define WHITE(c) (((unsigned char)(c))<=' ')    /* Assumes ASCII but faster */
#else
#define WHITE(c) ( ((c)==' ') || ((c)=='\t') || ((c)=='\n') || ((c)=='\r') )
#endif
```

```
/*                      Generic Communication Code                HTTCP.h
**                      ==========================
**
**      Requires tcp.h to be included first.
*/
#ifdef SHORT_NAMES
#define HTInetStatus            HTInStat
#define HTInetString            HTInStri
#define HTParseInet             HTPaInet
#endif

/*      Produce a string for an internet address
**      ----------------------------------------
**
** On exit,
**      returns a pointer to a static string which must be copied if
**              it is to be kept.
*/
#ifdef __STDC__
extern const char * HTInetString(struct sockaddr_in* sin);
#else
extern char * HTInetString();
#endif


/*      Encode INET status (as in sys/errno.h)                  inet_status()
**      ------------------
**
** On entry,
**      where           gives a description of what caused the error
**      global errno    gives the error number in the unix way.
**
** On return,
**      returns         a negative status in the unix way.
*/
#ifdef __STDC__
extern int HTInetStatus(char *where);
#else
extern int HTInetStatus();
#endif

/*      Publicly accesible variables
*/
extern struct sockaddr_in HTHostAddress;/* The internet address of the host */
                                        /* Valid after call to HTHostName() */

/*      Parse a cardinal value                                  parse_cardinal()
**      ----------------------
**
** On entry,
**      *pp         points to first character to be interpreted, terminated by
**                  non 0:9 character.
**      *pstatus    points to status already valid
**      maxvalue    gives the largest allowable value.
**
** On exit,
**      *pp         points to first unread character
**      *pstatus    points to status updated iff bad
*/
#ifdef __STDC__
```

```
extern unsigned int HTCardinal(int *pstatus,
        char            **pp,
        unsigned int    max_value);
#else
extern unsigned int HTCardinal();
#endif

/*      Parse an internet node address and port
**      ---------------------------------------
**
** On entry,
**      str     points to a string with a node name or number,
**              with optional trailing colon and port number.
**      sin     points to the binary internet address field.
**
** On exit,
**      *sin    is filled in. If no port is specified in str, that
**              field is left unchanged in *sin.
*/
#ifdef __STDC__
extern int HTParseInet(struct sockaddr_in* sin, const char *str);
#else
extern int HTParseInet();
#endif

/*      Get Name of This Machine
**      ------------------------
**
*/
#ifdef __STDC__
extern const char * HTHostName(void);
#else
extern char * HTHostName();
#endif
```

```
DMSLIO201W The following names are undefined:
IBMBLIIA
```

```
/* THIS EXEC FILE COMPILES AND LINKS A C PROGRAM */
/* arg name */
"GLOBAL TXTLIB EDCBASE IBMLIB CMSLIB"
"CMOD HTDAEMON FINDGATE "
/* "EXEC TCPLOAD" name "1400CERN C" */
/*MOD" name */
```

```
  /*  Txtlib's get redefined somehow by the CC exec so call CC  */
  /* then reglobal the necessary txtlibs. chee laug91           */

  address COMMAND

    preflibs = 'NAPL HANDYPAK MK2 M2LIB BTM FORTSLAC VSF2FORT'
    extralib = 'IBMLIB EDCBASE'
    cdisk    = 0

    parse upper arg cname .
    if cname = '' | cname = '?' then call ERR 1

  /* is C370 disk up? */
   'EXEC CMSSTACK LIFO QUERY DISK'
    do queued()
       parse upper pull label vaddr mode rwstat .
       if label = 'IBM-C' then cdisk = 1
    end
    if cdisk = 0 then 'EXEC GIME C370'
   'EXEC CC 'cname
    if rc ^= 0 then call ERR 2
   'GLOBAL TXTLIB EDCBASE IBMLIB'
   'GLOBAL LOADLIB EDCLINK'
   'EXEC CPLINK 'cname
    if rc ^= 0 then call ERR 3
   'EXEC CMOD' cname
    if rc ^= 0 then call ERR 4
   'GLOBAL TXTLIB 'preflibs extralib

 exit
 /* ------------------- EOProgram ---------------------- */

  ERR:
      parse upper arg errno

      select
        when errno = 1 then do
          'EXEC HELP CC'
        end
        when errno = 2 then do
          say 'Your C file failed to compile'
        end
        when errno = 3 then do
          say 'CPLINK failed '
        end
        when errno = 4 then do
          say 'CMOD failed'
        end
        otherwise nop
      end /* of select */
exit
```

```
/* */
ADDRESS command  /* PAD 3 before each PIPE STACK is a "fix" for the */
arg iinnpp       /* problem with ?? being taken as EOF by IBM C     */
Say '   NICFOUN1 Start->'iinnpp
parse var iinnpp fid '('dispmode'/'glued'/'versions')'
versions=translate(versions,' ','+')
dispmode=translate(dispmode,' ','+')
Queue '<SEEINDEX HREF=/FIND>'
If fid='' Then do; queue 'No input whatsoever'; call get_out; end
parse var fid group '.'diskname'.'ft'.'fn .
Say '   mode='dispmode 'glued='glued' versions='versions
SELECT /*_____________________________________________________________*/
 WHEN datatype(dispmode,'X') THEN DO
   CALL GET_MODE group'.'diskname 0;m_=result
   queue '<PLAINTEXT>'
   parse var dispmode _1 5 _2 7
   _1=x2d(_1)-1; _2=x2d(_2)
  'PIPE < 'fn ft m_' | drop '_1' | take '_2' | PAD 3 | STACK'
 END /* dispmode is a heXadecimal number --> linenumbers='X' */
 WHEN dispmode='X' THEN DO
   Call GET_MODE group'.'diskname 0;m_=result
   If glued='A' Then do
     pft='PRO:'||substr(ft,5)
     CALL GET_MODE group'.P_'diskname 0;m_p=result
   End; Else do;m_p=m_;pft=ft;end
  'EXECIO * DISKR 'fn pft m_p' (STEM _. MAR 9 * FIND /.cm EXP:/'
   If rc>0 Then,
    'EXECIO * DISKR 'fn pft m_p' (STEM _. MAR 9 * FIND /.CM EXP:/'
   queue '<title>'_.1'</title>'
   queue '<PLAINTEXT>'
   If ft='HELPPROG' Then 'PIPE < 'fn ft m_,
                         '| NFIND . | SPEC 12-90 1 | PAD 3 | STACK'
   Else 'PIPE < 'fn ft m_' | NFIND . | PAD 3 | STACK'
END /* of dispmode='X' */
WHEN dispmode='W' THEN DO
  pft='PRO:'||substr(ft,5)
  CALL GET_MODE group'.P_'diskname 0;m_p=result
 'EXECIO * DISKR 'fn pft m_p' (STEM _. MAR 9 * FIND /.cm EXP:/'
  If rc>0 Then,
   'EXECIO * DISKR 'fn pft m_p' (STEM _. MAR 9 * FIND /.CM EXP:/'
  queue '<title>'_.1'</title>'
 'EXECIO * DISKR 'fn pft m_p' (STEM _. MAR 9 * FIND /.cm TYP:/'
  If rc>0 Then types=''; else types=_.1
  Select
   When find(versions,'H') >0 Then do
     queue '<PLAINTEXT>'
     Call GET_MODE group'.'diskname 0;m_=result
    'PIPE < 'fn ft m_' | PAD 3 | STACK'
   End
   When find(versions,'S') >0 Then do
     parse var types 'SOURCE='sft . .
     If sft^='' Then do
      Call GET_MODE group'.S_'diskname 0;m_=result
     'PIPE < 'fn sft m_' | PAD 3 | STACK'
     End
     Else queue 'Something wrong with source'
   End
   Otherwise queue 'No displayable version available'
  End
END /* of dispmode='W' */
```

```
  WHEN dispmode='H' THEN DO
    Call GET_MODE group'.'diskname 0;m_=result
    If glued='A' Then do
      pft='PRO:'||substr(ft,5)
      CALL GET_MODE group'.P_'diskname 0;m_p=result
    End; Else do;m_p=m_;pft=ft;end
   'EXECIO * DISKR 'fn pft m_p' (STEM _. MAR 9 * FIND /.cm EXP:/'
    If rc>0 Then,
     'EXECIO * DISKR 'fn pft m_p' (STEM _. MAR 9 * FIND /.CM EXP:/'
    queue '<title>'_.1'</title>'
    queue '<PLAINTEXT>'
    If ft='HELPPROG' Then 'PIPE < 'fn ft m_,
                          ' | NFIND . | SPEC 12-90 1 | PAD 3 | STACK'
    Else 'PIPE < 'fn ft m_' | NFIND . | PAD 3 | STACK'
  END /* of dispmode='H' */
  OTHERWISE queue 'Something is wrong, tell BERND@CERNVM'
END /*_____________________________________________________________________*/
/*  'PIPE Stack | > bla bla a | console' */
CALL GET_OUT
RETURN queued()
/*********************************************************************/
GET_MODE:  procedure; arg disk flag
'GLOBALV SELECT :XFIND_0 GET ACC_DISKS '
 parse var acc_disks acc_disks ',' acc_links ',' acc_modes
 nn_n=find(acc_disks,disk)
 if nn_n>0 Then RETURN subword(acc_modes,nn_n,1)
'GLOBALV SELECT :XFIND_1 GET 'disk
 parse value value(disk) with id addr uaddr umode rpw .
 if id='' then do
    Queue 'The disk 'disk' is not defined'
    Call Get_out
 end
 If length(umode)=1 Then flag=1  /* disk accessed for all modes       */
 If flag & umode ^='.' & umode ^='' then RETURN umode
 else do
    l_=qdisk('?','ADDRESS')
   'EXECIO 1 CP (SKIP STRING LINK TO 'id addr 'AS' l_ 'RR' rpw
    If rc>2 Then Do
      Queue 'Linking problem with disk 'disk': TELL BERND@CERNVM'
      Call Get_out
    End
    zz=typemode('HT');m_=QDISK('?','MODE','-Z');'ACCESS 'l_ m_;zz=typemode(zz)
    If rc>0 Then Do
      Queue 'Accessing problem with disk 'disk': TELL BERND@CERNVM'
      Call Get_out
    End
    acc_disks=acc_disks' 'disk
    acc_links=acc_links' 'l_
    acc_modes=acc_modes' 'm_
   'GLOBALV SELECT :XFIND_0 SETL ACC_DISKS 'acc_disks','acc_links','acc_modes
 end
RETURN m_
/*--------------------------------------------------------------------*/
GET_OUT:
'GLOBALV SELECT :XFIND_0 GET ACC_DISKS '
 parse var acc_disks acc_disks ',' acc_links ',' acc_modes
'SET CMSTYPE HT'
 do k=1 to words(acc_links)
  'RELEASE 'WORD(acc_modes,k)
  'EXECIO 1 CP (SKIP STRING DETACH 'WORD(acc_links,k)
```

```
 end
'SET CMSTYPE RT'
'GLOBALV SELECT :XFIND_0 PURGE'
Say '    NICFOUN1 End  -> stacksize='queued()
EXIT queued()
```

```
-compile
htdaemon c
httcp c
findgate c

-CC
include ebdlib
cc name "(def ....)"

-header files
HTUtils.h
tcp.h
HTTCP.h

-cw
gime watcnew
cw fn.c (prm debug t prm vm t

-tcp/ip
gime tcpappl
files: manifest h


-c370
gime c370
cc fn
help file: cc helpcms
cc fn (xref list
cc fn (langlvl (ansi)

-duplication
string h          comment out strncpy
edcbase txtlib    no different from IBM-C's version

-error with HTDaemon c
bsdtypes h    (time_t was commented out in bsdtypes.h, put it in)
strncpy       comment out strncpy from string.h
logfile       change logfile to logfiles in HTDaemon.c
              real fix: change LogFile to logfile

-error with HTTCP c
sin_addr      HTDaemon (structure sockaddr_in defined by in.h by tcp.h
fix           cc fn (langlvl (ansi) in cc2 exec
              real fix: find newer version, WWW/Imple../HTTCP.c
port/index    fix: define index in string.h
              real fix: use strchr() instead of index

-vm tools
disklist     who has what
qdiskid      sho user addr
q txtlib     query current global txtlib
txtlib map fn generate map for a txtlib
```

```
/* REXX please     - Run Daemon */
address command
Say 'Access the disks for TCP/IP '
'EXEC GIME MAINT  5E5          (MODE -P'
'EXEC GIME TCPMNT 2B0 * READIT (MODE -P'  /* TCP/IP source               */
'EXEC GIME TCPMNT 410 * DACUR  (MODE -P'  /* TCP/IP base                 */
'EXEC GIME TCPMNT 412 * DACUR  (MODE -P'  /* TCP/IP text                 */
'EXEC GIME TCPMNT 441 * DACUR  (MODE -P'  /* TCP/IP last updates fixes   */

Say 'Indicate to CMS which libraries to search'

'GLOBAL TXTLIB EDCBASE IBMLIB CMSLIB'

Say 'Access the disk with all the indices of XFIND'

'EXEC DROP Z'
'EXEC GIME TESTSERV 201 Z'

Say 'Start the DAEMON'

'HTDAEMON -v -a *:8000 -l findlog'
```

```
/**/
   'EXEC GLOBADD LOADLIB EDCLINK'
/* 'htdaemon -v -a *:2784' */
   'htdaemon -v -a *:2784'
```

```
/* SET UP TCP ACCESS */
/* "GIME TCPMNT 582" tyh */
"GIME TCPAPPL "
"GLOBAL TXTLIB IBMLIB CMSLIB EDCBASE COMMTXT VSPASCAL AMPLANG"
```

RECOVER

```
Searching ...
*** Note: SPICELL is not now a valid VM userid ***

There are 16 copies of SPICELL 192 on backup tapes.
They are:

1 ) SPICELL 0192 dumped to file 257 of tape RL2748 on 10/03/95 at 19:45:10
2 ) SPICELL 0192 dumped to file 214 of tape RL1816 on 07/04/95 at 20:04:19
3 ) SPICELL 0192 dumped to file 81 of tape RL1698 on 04/04/95 at 22:01:55
4 ) SPICELL 0192 dumped to file 216 of tape RL2993 on 01/03/95 at 22:30:20
5 ) SPICELL 0192 dumped to file 136 of tape RL3154 on 10/04/94 at 20:37:49
6 ) SPICELL 0192 dumped to file 248 of tape RL0457 on 07/06/94 at 20:51:40
7 ) SPICELL 0192 dumped to file 168 of tape RL3326 on 04/06/94 at 03:28:17
8 ) SPICELL 0192 dumped to file 185 of tape RL3139 on 01/06/94 at 04:25:52
9 ) SPICELL 0192 dumped to file 27 of tape RL2980 on 10/06/93 at 03:06:46
10 ) SPICELL 0192 dumped to file 10 of tape RL2741 on 07/07/93 at 04:12:40
11 ) SPICELL 0192 dumped to file 136 of tape RL2217 on 04/07/93 at 01:42:14
12 ) SPICELL 0192 dumped to file 139 of tape RL2544 on 01/06/93 at 01:32:56
13 ) SPICELL 0192 dumped to file 176 of tape RL2060 on 10/07/92 at 09:11:56
14 ) SPICELL 0192 dumped to file 170 of tape RL1559 on 07/08/92 at 04:14:00
15 ) SPICELL 0192 dumped to file 30 of tape RL1798 on 04/08/92 at 02:39:53
16 ) SPICELL 0192 dumped to file 51 of tape RL1414 on 01/08/92 at 01:02:19

Which one of the above do you wish to restore?
( 1 - 16 , <CR>=none )
```

SLACVM SPIRES HEP Preprint Database

Search
: Perform search using standard SPIRES terms.

Help
: Get help for SPIRES

You can search this index. Type the keyword(s) you want to search for:

SLAC SPIRES HEP Preprint database search

Use standard SPIRES search terms such as...

```
find author Perl, M
find title tau and date 1980
```

```
                    #pragma nomargins nosequence
#ifndef __string
  #define __string 1

/*****************************************************************
 *  <string.H> header file                                       *
 *                                                               *
 *  IBM C/370 Library                                            *
 *                                                               *
 *  Program Number: 5688-039                                     *
 *  Version: 1.0                                                 *
 *  Release: 2.0                                                 *
 *  Modification: 0.0                                            *
 *                                                               *
 *  Restricted Materials of IBM                                  *
 *                                                               *
 *  Copyright:                                                   *
 *  International Business Machines Corporation 1988, 1989       *
 *****************************************************************/

  #ifndef __size_t
    typedef unsigned int size_t;
    #define __size_t 1
  #endif
  #ifndef NULL
    #define NULL ((void *)0)
  #endif

  void * memchr (const void *, int, size_t);
  #define memcpy(x,y,z) __memcpy(x,y,z)
  #pragma linkage(__memcpy,builtin)
  void * (__memcpy) (void *, const void *, size_t);
  #define memcmp(x,y,z) __memcmp(x,y,z)
  #pragma linkage(__memcmp,builtin)
  int (__memcmp) (const void *, const void *, size_t);
  #define memset(x,y,z) __memset(x,y,z)
  #pragma linkage(__memset,builtin)
  void * (__memset) (void *, int, size_t);
  void * (memcpy) (void *, const void *, size_t);
  int    (memcmp) (const void *, const void *, size_t);
  void * (memset) (void *, int, size_t);
  void * memmove (void *, const void *, size_t);
  char * strchr (const char *, int);
  int    strcmp (const char *, const char *);
  int    strcoll (const char *, const char *);
  #define strcpy(x,y) __strcpy(x,y)
  #pragma linkage(__strcpy,builtin)
  char * (__strcpy) (char *, const char *);
  char * (strcpy) (char *, const char *);
  #define strcat(x,y) __strcat(x,y)
  #pragma linkage(__strcat,builtin)
  char * (__strcat) (char *, const char *);
  char * (strcat) (char *, const char *);
/*char * (strncpy) (char *, const char *, size_t); terryh*/
  size_t strcspn (const char *, const char *);
  char * strerror (int);
  #define strlen(x) __strlen(x)
  #pragma linkage(__strlen,builtin)
  size_t (__strlen) (const char *);
  size_t (strlen) (const char *);
```

```
   #define strcmp(x,y) __strcmp(x,y)
   #pragma linkage(__strcmp,builtin)
   int (__strcmp) (const char *, const char *);
   int   (strcmp) (const char *, const char *);
   char * strncat (char *, const char *, size_t);
   int    strncmp (const char *, const char *,
                   size_t);
   char * strpbrk (const char *, const char *);
   char * strrchr (const char *, int);
   size_t strspn (const char *, const char *);
   char * strstr (const char *, const char *);
   char * strtok (char *, const char *);
   size_t strxfrm (char *, const char *, size_t);
/*  extern char *index();    * terryh */
#endif
```

```
/*      System-system differences for TCP include files and macros     tcp.h
**      ============================================================
**
**
**      This file includes for each system, the files necessary for
**      network and file I/O
**
**  History:
**      22 Feb 91       Written (TBL) as part of the WWW project.
*/

#define NETCLOSE close      /* Routine to close a TCP-IP socket          */
#define NETREAD  read       /* Routine to read from a TCP-IP socket      */
#define NETWRITE write      /* Routine to write to a TCP-IP socket       */

/*      On the NeXT, there's a little package of include files.
*/
#ifdef NeXT
#include <libc.h>                   /* NeXT has all this packaged up */
#define ntohs(x) (x)
#define htons(x) (x)
#include <sys/errno.h>              /* Get ECONNRESET etc */
#define SELECT                      /* Is supported ok */
#define INCLUDES_DONE

#else                               /* Not NeXT */
#include <stdio.h>

/*      MVS is compiled as for VM. MVS has no unix-style I/O
**      The command line compile options seem to come across in
**      lower case.
**
**      See aslo lots of VM stuff lower down.
*/
#ifdef mvs
#define MVS
#endif

#ifdef MVS
#define VM
#endif

/*      VM doesn't have a built-in predefined token, so we cheat: */
#ifdef __STDIO__
#define VM
#else
#include <string.h>                 /* For bzero etc - not NeXT or VM */
#endif
#define SELECT                      /* Handle >1 channel if we can.      */
#endif                              /* Not NeXT */


/*      Under VMS, there are many versions of TCP-IP. Define one if you
**      do not use Digital's UCX product:
**
**              UCX             DEC's "Ultrix connection" (default)
**              WIN_TCP         From Wollongong, now GEC software.
**              MULTINET        From SRI, now from TGV Inv.
**
**      The second two do not interfere with the unix i/o library, and so they
```

```
**      need special calls to read, write and close sockets.  In these cases the
**      socket number is a VMS channel number, so we make the HORRIBLE
**      assumption that a channel number will be greater than 10 but a
**      unix file descriptor less than 10.
*/
#ifdef vms
#ifdef WIN_TCP
#undef NETREAD
#undef NETWRITE
#undef NETCLOSE
#define NETREAD(s,b,l)  ((s)>10 ? netread((s),(b),(l)) : read((s),(b),(l)))
#define NETWRITE(s,b,l) ((s)>10 ? netwrite((s),(b),(l)) : write((s),(b),(l)))
#define NETCLOSE(s)     ((s)>10 ? netclose(s) : close(s))
#endif

#ifdef MULTINET
#undef NETCLOSE
#undef NETREAD
#undef NETWRITE
#define NETREAD(s,b,l)  ((s)>10 ? socket_read((s),(b),(l)) : read((s),(b),(l)))
#define NETWRITE(s,b,l) ((s)>10 ? socket_write((s),(b),(l)) : \
                                  write((s),(b),(l)))
#define NETCLOSE(s)     ((s)>10 ? socket_close(s) : close(s))
#endif

/*      Certainly this works for UCX:   @@@
*/
#include types
#include errno
#include time
#include string
#include stdio
#include file
#include unixio

#define INCLUDES_DONE

#include socket
#include in
#include inet
#include netdb
#define TCP_INCLUDES_DONE

#endif  /* vms */


/*      IBM VM/CMS or MVS
**      -----------------
**
**      Note:   All files must have lines <= 80 characters
**
**      Under VM, compile with "DEF=VM"
**
**      Under MVS, compile with "NOMAR DEF(MVS)" to get rid of 72 char margin
**        System include files TCPIP and COMMMAC neeed line number removal(!)
*/

#ifdef VM                       /* or MVS -- see above. */
#define NOT_ASCII               /* char type is not ASCII */
#define NO_UNIX_IO              /* Unix I/O routines are not supported */
```

```
#define SHORT_NAMES             /* 8 character uniqueness for globals */
#include <manifest.h>
#include <bsdtypes.h>
#include <stdefs.h>
#include <socket.h>
#include <in.h>
#include <netdb.h>
#include <errno.h>          /* independent */
extern char asciitoebcdic[], ebcdictoascii[];
#define TOASCII(c)   (c=='\n' ?  10  : ebcdictoascii[c])
#define FROMASCII(c) (c== 10  ? '\n' : asciitoebcdic[c])
#include <bsdtime.h>
#include <string.h>
#define INCLUDES_DONE
#define TCP_INCLUDES_DONE
#endif

/*      Regular BSD unix versions: */

#ifndef INCLUDES_DONE
#include <sys/types.h>
/* #include <streams/streams.h>                not ultrix */
#include <string.h>
#include <stdio.h>
#include <errno.h>          /* independent */
#include <sys/time.h>       /* independent */
#include <sys/stat.h>
#include <sys/param.h>
#include <sys/file.h>       /* For open() etc */
#endif  /* Normal includes */


/*      Default include files for TCP
*/
#ifndef TCP_INCLUDES_DONE
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>      /* Must be after netinet/in.h */
#include <netdb.h>
#endif  /* TCP includes */


/*      Default macros for manipulating masks for select()
*/
#ifndef FD_SET
typedef unsigned int fd_set;
#define FD_SET(fd,pmask) (*(pmask)) |=  (1<<(fd))
#define FD_CLR(fd,pmask) (*(pmask)) &= ~(1<<(fd))
#define FD_ZERO(pmask)   (*(pmask))=0
#define FD_ISSET(fd,pmask) (*(pmask) & (1<<(fd)))
#endif


/*      Default macros for converting characters
**
*/
#ifndef TOASCII
#define TOASCII(c) (c)
#define FROMASCII(c) (c)
#endif
```

```
*******************************************************************************
*                                                                             *
* Start execution of TYH239 for TERRYH, class S.                              *
* Execution begins at 18:40:52 on 14 Nov 1991 in BATCH07A at SLACVM.          *
* Submitted at        18:40:43 on 14 Nov 1991 for TERRYH at SLACVM.           *
* Running CURCMS, mode 370, storage = 4096K, time limit = 2 minutes.          *
* Max print = 10K lines, max punch = 100K cards.                              *
*                                                                             *
*******************************************************************************

-----> Execute "R$COV193"

** RECOVER TERRYH 193 to TDISK **
DASD 0555 DEFINED
TAPE 0F13 ATTACHED TO BATCH07A 0181
TAPE 0181 ON DEV  0F13 R/W
MNTREQ100I 18:40:53 Mounting volume RL0357 on device 181, SL NoRing
MNTVER101I 18:41:18 Volume RL0357 on device 181 has been verified
Rewind complete
RESTORING TYH193
DATA DUMPED   10/31/91 AT 04.35.56  GMT FROM TYH193 RESTORED  TO SCRATCH
INPUT CYLINDER EXTENTS      OUTPUT CYLINDER EXTENTS
        START     STOP      START     STOP
        0000      0000      0000      0000
END OF RESTORE
END OF JOB
18:42:59  * MSG FROM VMSETUPA: VMSCAN001I VMSETUP REQUEST FOR TERRYH TYH239 WILL BE
TAPE 0181 DETACHED BY VMSETUPA, TAPE VOLUME(S) UNLOADED
DMSCAR002E FILE(S) NOT FOUND.
FILENAME FILETYPE FM FORMAT LRECL       RECS     BLOCKS   DATE      TIME     LABEL
A        C        Z1 F          80          5          1 10/16/91 18:13:27 TYH193
A        ERRS     Z1 V          49          1          1 10/16/91 18:15:35 TYH193
A        TEXT     Z1 F          80          5          1 10/16/91 18:15:35 TYH193
BSDTYPES H        Z1 V          70        120          1 10/15/91 18:01:30 TYH193
CC_ALL   EXEC     Z1 V          81         13          1 10/14/91 15:29:29 TYH193
CC2      EXEC     Z1 V          58          8          1 10/30/91 14:18:05 TYH193
CMOD     EXEC     Z2 V          62        201          2 10/30/91 15:38:28 TYH193
CMS      EXEC     Z1 F          80         13          1 10/19/91 10:50:58 TYH193
CMS      PRESERVE Z1 V          65         30          1 10/30/91 16:15:36 TYH193
CPOBJ    RMAP     Z1 F          80          3          1 10/16/91 18:46:59 TYH193
DLOAD    EXEC     Z1 V          62          6          1 10/30/91 15:35:54 TYH193
EDCXV    C        Z1 F          80          5          1 10/16/91 18:08:43 TYH193
FGET     EXEC     Z2 V          78        129          2  9/26/91 11:30:24 TYH193
FINDGATE C        Z1 V          78        245          2  9/26/91  9:58:03 TYH193
FINDGATE C2       Z1 V          78        245          2 10/14/91 15:28:41 TYH193
FINDGATE TEXT     Z1 F          80        113          3 10/30/91 14:18:46 TYH193
FSEARCH  EXEC     Z1 V          76        120          2  9/26/91 10:43:32 TYH193
HTDAEMON C        Z1 V          85        639          5 10/15/91 18:10:19 TYH193
HTDAEMON C2       Z1 V          85        639          5 10/15/91 15:45:38 TYH193
HTDAEMON MODULE   Z1 V       65535          5         32 10/30/91 16:15:39 TYH193
HTDAEMON TEXT     Z1 F          80        183          4 10/30/91 14:18:19 TYH193
HTTCP    C        Z1 V          79        282          2 10/16/91 17:04:54 TYH193
HTTCP    H        Z1 V          78         94          1 10/14/91 15:47:38 TYH193
HTTCP    TEXT     Z1 F          80         77          2 10/30/91 14:18:32 TYH193
HTUTILS  H        Z1 V          79         85          1 10/14/91 15:47:21 TYH193
LOAD     MAP      Z1 F         100        392         10 10/30/91 15:34:23 TYH193
MAKE     EXEC     Z1 V          51          6          1  9/19/91 14:06:07 TYH193
MYCC     EXEC     Z1 V          63         51          1 10/16/91 18:28:29 TYH193
NICFOUN1 EXEC     Z2 V          78        125          2  9/24/91 11:01:53 TYH193
PROFILE  EXEC     Z1 V          72         28          1  9/26/91 10:44:28 TYH193
```

```
Q        Q        Z1 F          80          52          2 10/16/91 17:22:58 TYH193
RUN      EXEC     Z1 V          74          21          1  9/19/91 14:19:57 TYH193
SETUP    EXEC     Z1 V          62           4          1 10/15/91 17:47:15 TYH193
STRING   H        Z2 F          80          75          2 10/16/91 17:07:43 TYH193
TCP      H        Z1 V          91         180          2 10/14/91 15:47:28 TYH193
-----> Batch return code = 0.

-----> System cleanup $BCLEAN.

*****************************************************************************
*                                                                           *
* End execution of TYH239 for TERRYH at 18:42:59 on 14/11/91.               *
*                                                                           *
* Actual CPU time (mm:ss[.th]):      Virtual = 0:00.37 , Total = 0:00.64    *
* I/O counts:  SIO=000531 RDR-000000 PRT-000100 PCH-000071                  *
*                                                                           *
* Space used on A-disk at job end:  48K                                     *
*                                                                           *
* I/O counts on units still available                                       *
* Spool units                                                               *
* 0009:    87    000C:     0    000D:     2    000E:     0                  *
* 00AE:     0                                                               *
* Dasd units                                                                *
* 0190(S):    25    0191(A):    76    019E(Y):    45    0555(Z):    22     *
* 059E(T):    15    059F(U):     5    05A0(V):    32                       *
*****************************************************************************
```

UNIX Command Summary with an Extract of Output

Commands used to copy files RECOVERed from CMS Backup to WINTERS 294 (a SPACE TEMPDISK), from that SLACVM minidisk on to UNIX AFS WWW archive space:

/afs/slac.stanford.edu/www/archive/1992/SLACVM/spicell/192/rl1414

The commands (almost always) and output were cut and pasted from the xterm (or aixterm) window where the work was executed. Text in square brackets ([ ]) is commentary.

```
pwd
/afs/slac.stanford.edu/www/archive/1992/SLACVM/spicell/192/rl1414

date
Wed Jul 22 02:18:54 PDT 1998

vmnfs reaccess winters.294
[SLACVM.SLAC.STANFORD.EDU]
Disk WINTERS.294 refreshed.

cp -p /nfs/slacvm/ascii/winters.294/* .
695 cassandra /192/rl1414> chmod a-x *
696 cassandra /192/rl1414> mv readme.rl1414 README

ls -l | grep "Jul 21"
-r--r--r--   1 winters  sf          9877 Jul 21 19:30 README
-r--r--r--   1 winters  sf          1337 Jul 21 19:31 cms.preserve
-r--r--r--   1 winters  sf          1463 Jul 21 18:58 spice192.backup
-r--r--r--   1 winters  sf          5400 Jul 21 19:31 winters.filelist

ls -l >winters.ls
701 cassandra /192/rl1414> chmod u-w winters.ls

wc winters.ls
     53     470    3395 winters.ls
703 cassandra /192/rl1414> wc winters.filelist
     50     650    5400 winters.filelist
...
cp -p /nfs/slacvm/ascii/winters.294/cms.preserve .

ls -l cms.preserve
-r-xr-xr-x   1 winters  sf          1136 Nov  6 1991  cms.preserve*
712 cassandra /192/rl1414> chmod a-x cms.preserve
713 cassandra /192/rl1414> ls -l cms.preserve
-r--r--r--   1 winters  sf          1136 Nov  6 1991  cms.preserve

 cp -p /nfs/slacvm/ascii/winters.294/winters.filelist .
overwrite ./winters.filelist? y
cp: ./winters.filelist: The file access permissions do not allow the specified actio
715 cassandra /192/rl1414> rm winters.filelist
rm: Remove winters.filelist? y
716 cassandra /192/rl1414> cp -p /nfs/slacvm/ascii/winters.294/winters.filelist .
717 cassandra /192/rl1414> chmod a-x winters.filelist
718 cassandra /192/rl1414> ls -l winters.filelist
-r--r--r--   1 winters  sf          5584 Jul 21 19:53 winters.filelist

rm winters.ls
```

```
rm: Remove winters.ls? y
721 cassandra /192/rl1414> ls -l >winters.ls
722 cassandra /192/rl1414> chmod u-w winters.ls

vmnfs refresh winters.294
[SLACVM.SLAC.STANFORD.EDU]
Disk WINTERS.294 refreshed.
727 cassandra /192/rl1414> rm winters.ls
rm: Remove winters.ls? y
728 cassandra /192/rl1414> rm winters.filelist
rm: Remove winters.filelist? y

 vmnfs refresh winters.294
[SLACVM.SLAC.STANFORD.EDU]
Disk WINTERS.294 refreshed.
733 cassandra /192/rl1414> cp -p /nfs/slacvm/ascii/winters.294/winters.filelist .
734 cassandra /192/rl1414> chmod a-x winters.filelist
735 cassandra /192/rl1414> ls -l winters.filelist
-r--r--r--   1 winters  sf          5508 Jul 21 19:57 winters.filelist
736 cassandra /192/rl1414> ls -l cms.preserve

-r--r--r--   1 winters  sf          1136 Nov  6 1991  cms.preserve
737 cassandra /192/rl1414>
737 cassandra /192/rl1414> ls -l > winters.ls
738 cassandra /192/rl1414> chmod u-w winters.ls


 wc winters.ls
     54     479    3463 winters.ls
740 cassandra /192/rl1414> wc winters.filelist
     51     663    5508 winters.filelist
741 cassandra /192/rl1414> grep -i filelist winters.filelist
      WINTERS  FILELIST A0 V        107        54          2  7/22/98  2:53:36 SPC2
742 cassandra /192/rl1414> grep -i filelist winters.ls
-r--r--r--   1 winters  sf          5508 Jul 21 19:57 winters.filelist
743 cassandra /192/rl1414> grep -i total winters.ls
total 728
744 cassandra /192/rl1414> grep -i winters.ls winters.ls
-rw-r--r--   1 winters  sf             0 Jul 22 03:04 winters.ls
745 cassandra /192/rl1414> grep -i unix-commands  winters.ls

date
Wed Jul 22 03:07:32 PDT 1998



22 Jul 1998                                                Joan M. Winters


22


-rw-r--r--   1 winters  sf          2096 Jul 22 02:56 unix-commands
```

WINTERS	FILELIST	A0	V	107	54	2	7/22/98	2:53:36	SPC2
README	RL1414	A1	V	107	173	3	7/22/98	2:49:19	SPC2
WINTERS	JCW120	A1	F	132	193	7	7/22/98	2:11:19	SPC2
SPICE192	BACKUP	A1	V	75	27	1	7/22/98	1:58:34	SPC2
SPIRES	INDEX	A1	V	43	12	1	12/12/91	15:59:13	SPC2
FSEARCH	EXEC	A1	V	40	7	1	12/09/91	18:10:32	SPC2
SPIRES	HTML	A1	V	74	9	1	12/09/91	17:20:32	SPC2
DEFAULT	HTML	A1	V	81	13	1	12/06/91	21:15:48	SPC2
BINLIST	HTML	A1	V	73	13	1	12/06/91	21:11:53	SPC2
BINLIST	INDEX	A1	F	80	8	1	12/06/91	21:00:22	SPC2
FGET	EXEC	A2	V	60	16	1	12/06/91	20:45:45	SPC2
HTDAEMON	MODULE	A1	V	65535	5	32	12/05/91	16:03:14	SPC2
EDCXV	TEXT	A1	F	80	9	1	12/05/91	16:03:06	SPC2
FINDGATE	TEXT	A1	F	80	113	3	12/05/91	16:02:49	SPC2
LOAD	MAP	A1	F	100	2	1	12/05/91	15:55:46	SPC2
TERRYH	TYH239	A1	F	132	87	3	11/14/91	18:43:33	SPC2
FGET	ORIG	A2	V	78	132	2	11/06/91	17:08:04	SPC2
CMS	PRESERVE	A1	V	65	30	1	11/06/91	17:06:02	SPC2
FINDGATE	C	A1	V	78	245	2	11/06/91	17:05:49	SPC2
RUNDAEMO	EXEC	A1	V	33	5	1	11/06/91	17:03:09	SPC2
FSEARCH	SAVE	A1	V	76	123	2	11/06/91	14:06:50	SPC2
FINDGATE	SLACVM	A1	V	78	245	2	11/06/91	13:57:17	SPC2
FEARCH	ORIG	A1	V	76	125	2	11/06/91	12:04:44	SPC2
SETUP	EXEC	A1	V	62	4	1	11/05/91	14:57:21	SPC2
HTTCP	TEXT	A1	F	80	77	2	11/05/91	14:38:12	SPC2
HTDAEMON	TEXT	A1	F	80	183	4	11/05/91	14:37:48	SPC2
CMOD	EXEC	A2	V	68	203	2	11/04/91	11:35:12	SPC2
FINDGATE	ORIG	A1	V	78	245	2	11/04/91	11:30:12	SPC2
DLOAD	EXEC	A1	V	64	4	1	11/01/91	15:15:36	SPC2
A	EXEC	A1	V	68	10	1	11/01/91	15:13:32	SPC2
CC2	EXEC	A1	V	58	8	1	10/30/91	14:18:05	SPC2
CPOBJ	RMAP	A1	F	80	3	1	10/16/91	18:46:59	SPC2
MYCC	EXEC	A1	V	63	51	1	10/16/91	18:28:29	SPC2
A	ERRS	A1	V	49	1	1	10/16/91	18:15:35	SPC2
A	TEXT	A1	F	80	5	1	10/16/91	18:15:35	SPC2
A	C	A1	F	80	5	1	10/16/91	18:13:27	SPC2
EDCXV	C	A1	F	80	5	1	10/16/91	18:08:43	SPC2
Q	Q	A1	F	80	52	2	10/16/91	17:22:58	SPC2
STRING	H	A2	F	80	75	2	10/16/91	17:07:43	SPC2
HTTCP	C	A1	V	79	282	2	10/16/91	17:04:54	SPC2
HTDAEMON	C	A1	V	85	639	5	10/15/91	18:10:19	SPC2
BSDTYPES	H	A1	V	70	120	1	10/15/91	18:01:30	SPC2
HTDAEMON	C2	A1	V	85	639	5	10/15/91	15:45:38	SPC2
HTTCP	H	A1	V	78	94	1	10/14/91	15:47:38	SPC2
TCP	H	A1	V	91	180	2	10/14/91	15:47:28	SPC2
HTUTILS	H	A1	V	79	85	1	10/14/91	15:47:21	SPC2
CC_ALL	EXEC	A1	V	81	13	1	10/14/91	15:29:29	SPC2
FINDGATE	C2	A1	V	78	245	2	10/14/91	15:28:41	SPC2
NICFOUN1	EXEC	A2	V	78	125	2	9/24/91	11:01:53	SPC2
RUN	EXEC	A1	V	74	21	1	9/19/91	14:19:57	SPC2
MAKE	EXEC	A1	V	51	6	1	9/19/91	14:06:07	SPC2

```
*********************************************************************
*                                                                   *
* Start execution of JCW120 for WINTERS, class S.                   *
* Execution begins at 01:59:53 on 22 Jul 1998 in BATCH08A at  SLACVM.*
* Submitted at        01:59:48 on 22 Jul 1998 for WINTERS at  SLACVM.*
* Running CURCMS, mode XA, storage = 4096K, time limit = 2 minutes. *
* Max print = 10K lines, max punch = 100K cards.                    *
*                                                                   *
*********************************************************************

-----> Execute "R$COV192"

** RECOVER SPICELL 192 to TDISK **
DASD 0555 DEFINED
TAPE 0F13 ATTACHED TO BATCH08A 0181
TAPE 0181 ON DEV  0F13 3490 R/W SUBCH = 000D
MNTREQ100I 01:59:55 Mounting volume RL1414 on device 181, SL NoRing
MNTVER107I 02:00:25 Remounting volume RL1414 on device 181; intervention required
MNTVER101I 02:00:52 Volume RL1414 on device 181 has been verified
Rewind complete
RESTORING WWW
DATA DUMPED   01/08/92 AT 01.02.10  GMT FROM WWW    RESTORED TO SCRATCH
INPUT CYLINDER EXTENTS      OUTPUT CYLINDER EXTENTS
        START     STOP      START     STOP
        0000      0004      0000      0004
END OF RESTORE
END OF JOB
02:01:23  * MSG FROM VMSETUPA: VMSCAN001I VMSETUP REQUEST FOR WINTERS JCW120 WILL BE
TAPE 0181 DETACHED BY VMSETUPA
CARD DUMP A C Z1
PUN FILE 0636 SENT TO   WINTERS  RDR AS  7305 RECS 0003 CPY  001 Y NOHOLD NOKEEP
CARD DUMP A ERRS Z1
PUN FILE 0637 SENT TO   WINTERS  RDR AS  7306 RECS 0003 CPY  001 Y NOHOLD NOKEEP
CARD DUMP A EXEC Z1
PUN FILE 0638 SENT TO   WINTERS  RDR AS  7307 RECS 0005 CPY  001 Y NOHOLD NOKEEP
CARD DUMP A TEXT Z1
PUN FILE 0639 SENT TO   WINTERS  RDR AS  7308 RECS 0005 CPY  001 Y NOHOLD NOKEEP
CARD DUMP BINLIST HTML Z1
PUN FILE 0640 SENT TO   WINTERS  RDR AS  7309 RECS 0006 CPY  001 Y NOHOLD NOKEEP
CARD DUMP BINLIST INDEX Z1
PUN FILE 0641 SENT TO   WINTERS  RDR AS  7310 RECS 0004 CPY  001 Y NOHOLD NOKEEP
CARD DUMP BSDTYPES H Z1
PUN FILE 0642 SENT TO   WINTERS  RDR AS  7311 RECS 0044 CPY  001 Y NOHOLD NOKEEP
CARD DUMP CC_ALL EXEC Z1
PUN FILE 0643 SENT TO   WINTERS  RDR AS  7312 RECS 0005 CPY  001 Y NOHOLD NOKEEP
CARD DUMP CC2 EXEC Z1
PUN FILE 0644 SENT TO   WINTERS  RDR AS  7313 RECS 0006 CPY  001 Y NOHOLD NOKEEP
CARD DUMP CMOD EXEC Z2
PUN FILE 0645 SENT TO   WINTERS  RDR AS  7314 RECS 0057 CPY  001 Y NOHOLD NOKEEP
CARD DUMP CMS PRESERVE Z1
PUN FILE 0646 SENT TO   WINTERS  RDR AS  7315 RECS 0014 CPY  001 Y NOHOLD NOKEEP
CARD DUMP CPOBJ RMAP Z1
PUN FILE 0647 SENT TO   WINTERS  RDR AS  7316 RECS 0004 CPY  001 Y NOHOLD NOKEEP
CARD DUMP DEFAULT HTML Z1
PUN FILE 0648 SENT TO   WINTERS  RDR AS  7317 RECS 0007 CPY  001 Y NOHOLD NOKEEP
CARD DUMP DLOAD EXEC Z1
PUN FILE 0649 SENT TO   WINTERS  RDR AS  7318 RECS 0004 CPY  001 Y NOHOLD NOKEEP
CARD DUMP EDCXV C Z1
PUN FILE 0650 SENT TO   WINTERS  RDR AS  7319 RECS 0003 CPY  001 Y NOHOLD NOKEEP
CARD DUMP EDCXV TEXT Z1
```

```
PUN FILE 0651 SENT TO   WINTERS  RDR AS  7320 RECS 0007 CPY   001 Y NOHOLD NOKEEP
CARD DUMP FEARCH ORIG Z1
PUN FILE 0652 SENT TO   WINTERS  RDR AS  7321 RECS 0057 CPY   001 Y NOHOLD NOKEEP
CARD DUMP FGET EXEC Z2
PUN FILE 0653 SENT TO   WINTERS  RDR AS  7322 RECS 0007 CPY   001 Y NOHOLD NOKEEP
CARD DUMP FGET ORIG Z2
PUN FILE 0654 SENT TO   WINTERS  RDR AS  7323 RECS 0065 CPY   001 Y NOHOLD NOKEEP
CARD DUMP FINDGATE C Z1
PUN FILE 0655 SENT TO   WINTERS  RDR AS  7324 RECS 0091 CPY   001 Y NOHOLD NOKEEP
CARD DUMP FINDGATE C2 Z1
PUN FILE 0656 SENT TO   WINTERS  RDR AS  7325 RECS 0091 CPY   001 Y NOHOLD NOKEEP
CARD DUMP FINDGATE ORIG Z1
PUN FILE 0657 SENT TO   WINTERS  RDR AS  7326 RECS 0091 CPY   001 Y NOHOLD NOKEEP
CARD DUMP FINDGATE SLACVM Z1
PUN FILE 0658 SENT TO   WINTERS  RDR AS  7327 RECS 0091 CPY   001 Y NOHOLD NOKEEP
CARD DUMP FINDGATE TEXT Z1
PUN FILE 0659 SENT TO   WINTERS  RDR AS  7328 RECS 0100 CPY   001 Y NOHOLD NOKEEP
CARD DUMP FSEARCH EXEC Z1
PUN FILE 0660 SENT TO   WINTERS  RDR AS  7329 RECS 0004 CPY   001 Y NOHOLD NOKEEP
CARD DUMP FSEARCH SAVE Z1
PUN FILE 0661 SENT TO   WINTERS  RDR AS  7330 RECS 0057 CPY   001 Y NOHOLD NOKEEP
CARD DUMP HTDAEMON C Z1
PUN FILE 0662 SENT TO   WINTERS  RDR AS  7331 RECS 0211 CPY   001 Y NOHOLD NOKEEP
CARD DUMP HTDAEMON C2 Z1
PUN FILE 0663 SENT TO   WINTERS  RDR AS  7332 RECS 0211 CPY   001 Y NOHOLD NOKEEP
CARD DUMP HTDAEMON MODULE Z1
PUN FILE 0664 SENT TO   WINTERS  RDR AS  7333 RECS 1535 CPY   001 Y NOHOLD NOKEEP
CARD DUMP HTDAEMON TEXT Z1
PUN FILE 0665 SENT TO   WINTERS  RDR AS  7334 RECS 0139 CPY   001 Y NOHOLD NOKEEP
CARD DUMP HTTCP C Z1
PUN FILE 0666 SENT TO   WINTERS  RDR AS  7335 RECS 0091 CPY   001 Y NOHOLD NOKEEP
CARD DUMP HTTCP H Z1
PUN FILE 0667 SENT TO   WINTERS  RDR AS  7336 RECS 0032 CPY   001 Y NOHOLD NOKEEP
CARD DUMP HTTCP TEXT Z1
PUN FILE 0668 SENT TO   WINTERS  RDR AS  7337 RECS 0056 CPY   001 Y NOHOLD NOKEEP
CARD DUMP HTUTILS H Z1
PUN FILE 0669 SENT TO   WINTERS  RDR AS  7338 RECS 0027 CPY   001 Y NOHOLD NOKEEP
CARD DUMP LOAD MAP Z1
PUN FILE 0670 SENT TO   WINTERS  RDR AS  7339 RECS 0003 CPY   001 Y NOHOLD NOKEEP
CARD DUMP MAKE EXEC Z1
PUN FILE 0671 SENT TO   WINTERS  RDR AS  7340 RECS 0005 CPY   001 Y NOHOLD NOKEEP
CARD DUMP MYCC EXEC Z1
PUN FILE 0672 SENT TO   WINTERS  RDR AS  7341 RECS 0017 CPY   001 Y NOHOLD NOKEEP
CARD DUMP NICFOUN1 EXEC Z2
PUN FILE 0673 SENT TO   WINTERS  RDR AS  7342 RECS 0061 CPY   001 Y NOHOLD NOKEEP
CARD DUMP Q Q Z1
PUN FILE 0674 SENT TO   WINTERS  RDR AS  7343 RECS 0016 CPY   001 Y NOHOLD NOKEEP
CARD DUMP RUN EXEC Z1
PUN FILE 0675 SENT TO   WINTERS  RDR AS  7344 RECS 0010 CPY   001 Y NOHOLD NOKEEP
CARD DUMP RUNDAEMO EXEC Z1
PUN FILE 0676 SENT TO   WINTERS  RDR AS  7345 RECS 0004 CPY   001 Y NOHOLD NOKEEP
CARD DUMP SETUP EXEC Z1
PUN FILE 0677 SENT TO   WINTERS  RDR AS  7346 RECS 0004 CPY   001 Y NOHOLD NOKEEP
CARD DUMP SPIRES HTML Z1
PUN FILE 0678 SENT TO   WINTERS  RDR AS  7347 RECS 0006 CPY   001 Y NOHOLD NOKEEP
CARD DUMP SPIRES INDEX Z1
PUN FILE 0679 SENT TO   WINTERS  RDR AS  7348 RECS 0005 CPY   001 Y NOHOLD NOKEEP
CARD DUMP STRING H Z2
PUN FILE 0680 SENT TO   WINTERS  RDR AS  7349 RECS 0033 CPY   001 Y NOHOLD NOKEEP
CARD DUMP TCP H Z1
```

```
PUN FILE 0681 SENT TO   WINTERS  RDR AS  7350 RECS 0063 CPY   001 Y NOHOLD NOKEEP
CARD DUMP TERRYH TYH239 Z1
PUN FILE 0682 SENT TO   WINTERS  RDR AS  7351 RECS 0051 CPY   001 Y NOHOLD NOKEEP
FILENAME FILETYPE FM FORMAT LRECL       RECS     BLOCKS   DATE     TIME     LABEL
A        C        Z1 F          80          5          1 10/16/91 18:13:27 WWW
A        ERRS     Z1 V          49          1          1 10/16/91 18:15:35 WWW
A        EXEC     Z1 V          68         10          1 11/01/91 15:13:32 WWW
A        TEXT     Z1 F          80          5          1 10/16/91 18:15:35 WWW
BINLIST  HTML     Z1 V          73         13          1 12/06/91 21:11:53 WWW
BINLIST  INDEX    Z1 F          80          8          1 12/06/91 21:00:22 WWW
BSDTYPES H        Z1 V          70        120          1 10/15/91 18:01:30 WWW
CC_ALL   EXEC     Z1 V          81         13          1 10/14/91 15:29:29 WWW
CC2      EXEC     Z1 V          58          8          1 10/30/91 14:18:05 WWW
CMOD     EXEC     Z2 V          68        203          2 11/04/91 11:35:12 WWW
CMS      PRESERVE Z1 V          65         30          1 11/06/91 17:06:02 WWW
CPOBJ    RMAP     Z1 F          80          3          1 10/16/91 18:46:59 WWW
DEFAULT  HTML     Z1 V          81         13          1 12/06/91 21:15:48 WWW
DLOAD    EXEC     Z1 V          64          4          1 11/01/91 15:15:36 WWW
EDCXV    C        Z1 F          80          5          1 10/16/91 18:08:43 WWW
EDCXV    TEXT     Z1 F          80          9          1 12/05/91 16:03:06 WWW
FEARCH   ORIG     Z1 V          76        125          2 11/06/91 12:04:44 WWW
FGET     EXEC     Z2 V          60         16          1 12/06/91 20:45:45 WWW
FGET     ORIG     Z2 V          78        132          2 11/06/91 17:08:04 WWW
FINDGATE C        Z1 V          78        245          2 11/06/91 17:05:49 WWW
FINDGATE C2       Z1 V          78        245          2 10/14/91 15:28:41 WWW
FINDGATE ORIG     Z1 V          78        245          2 11/04/91 11:30:12 WWW
FINDGATE SLACVM   Z1 V          78        245          2 11/06/91 13:57:17 WWW
FINDGATE TEXT     Z1 F          80        113          3 12/05/91 16:02:49 WWW
FSEARCH  EXEC     Z1 V          40          7          1 12/09/91 18:10:32 WWW
FSEARCH  SAVE     Z1 V          76        123          2 11/06/91 14:06:50 WWW
HTDAEMON C        Z1 V          85        639          5 10/15/91 18:10:19 WWW
HTDAEMON C2       Z1 V          85        639          5 10/15/91 15:45:38 WWW
HTDAEMON MODULE   Z1 V       65535          5         32 12/05/91 16:03:14 WWW
HTDAEMON TEXT     Z1 F          80        183          4 11/05/91 14:37:48 WWW
HTTCP    C        Z1 V          79        282          2 10/16/91 17:04:54 WWW
HTTCP    H        Z1 V          78         94          1 10/14/91 15:47:38 WWW
HTTCP    TEXT     Z1 F          80         77          2 11/05/91 14:38:12 WWW
HTUTILS  H        Z1 V          79         85          1 10/14/91 15:47:21 WWW
LOAD     MAP      Z1 F         100          2          1 12/05/91 15:55:46 WWW
MAKE     EXEC     Z1 V          51          6          1  9/19/91 14:06:07 WWW
MYCC     EXEC     Z1 V          63         51          1 10/16/91 18:28:29 WWW
NICFOUN1 EXEC     Z2 V          78        125          2  9/24/91 11:01:53 WWW
Q        Q        Z1 F          80         52          2 10/16/91 17:22:58 WWW
RUN      EXEC     Z1 V          74         21          1  9/19/91 14:19:57 WWW
RUNDAEMO EXEC     Z1 V          33          5          1 11/06/91 17:03:09 WWW
SETUP    EXEC     Z1 V          62          4          1 11/05/91 14:57:21 WWW
SPIRES   HTML     Z1 V          74          9          1 12/09/91 17:20:32 WWW
SPIRES   INDEX    Z1 V          43         12          1 12/12/91 15:59:13 WWW
STRING   H        Z2 F          80         75          2 10/16/91 17:07:43 WWW
TCP      H        Z1 V          91        180          2 10/14/91 15:47:28 WWW
TERRYH   TYH239   Z1 F         132         87          3 11/14/91 18:43:33 WWW
-----> Batch return code = 0.

-----> System cleanup $BCLEAN.

*************************************************************************
*                                                                       *
* End execution of JCW120 for WINTERS at 02:01:28 on 22/07/98.          *
*                                                                       *
* Actual CPU time (mm:ss[.th]):      Virtual = 0:00.48 , Total = 0:01.05  *
```

```
* I/O counts:  SIO=000623 RDR-000000 PRT-000206 PCH-003489                    *
*                                                                             *
* Space used on A-disk at job end:  48K                                       *
*                                                                             *
* I/O counts on units still available                                         *
* Spool units                                                                 *
* 0009:   146    000C:     0    000D:    71    000E:     0                    *
* 00AE:     0                                                                 *
* Dasd units                                                                  *
* 0190(S):    34    0191(A):    98    019E(Y):    37    0555(Z):   172       *
* 059E(T):    12    059F(U):     6    05A0(V):    27    05A1(R):    18       *
*******************************************************************************
```

```
total 728
-r--r--r--   1 winters  sf       9877 Jul 21 19:30 README
-r--r--r--   1 winters  sf        405 Oct 16 1991  a.c
-r--r--r--   1 winters  sf         50 Oct 16 1991  a.errs
-r--r--r--   1 winters  sf        242 Nov  1 1991  a.exec
-r--r--r--   1 winters  sf        405 Oct 16 1991  a.text
-r--r--r--   1 winters  sf        319 Dec  6 1991  binlist.html
-r--r--r--   1 winters  sf        648 Dec  6 1991  binlist.index
-r--r--r--   1 winters  sf       3321 Oct 15 1991  bsdtypes.h
-r--r--r--   1 winters  sf        253 Oct 30 1991  cc2.exec
-r--r--r--   1 winters  sf        208 Oct 14 1991  cc_all.exec
-r--r--r--   1 winters  sf       4434 Nov  4 1991  cmod.exec
-r--r--r--   1 winters  sf       1136 Nov  6 1991  cms.preserve
-r--r--r--   1 winters  sf        243 Oct 16 1991  cpobj.rmap
-r--r--r--   1 winters  sf        347 Dec  6 1991  default.html
-r--r--r--   1 winters  sf        150 Nov  1 1991  dload.exec
-r--r--r--   1 winters  sf        405 Oct 16 1991  edcxv.c
-r--r--r--   1 winters  sf        729 Dec  5 1991  edcxv.text
-r--r--r--   1 winters  sf       4663 Nov  6 1991  fearch.orig
-r--r--r--   1 winters  sf        399 Dec  6 1991  fget.exec
-r--r--r--   1 winters  sf       5079 Nov  6 1991  fget.orig
-r--r--r--   1 winters  sf       7057 Nov  6 1991  findgate.c
-r--r--r--   1 winters  sf       7052 Oct 14 1991  findgate.c2
-r--r--r--   1 winters  sf       7057 Nov  4 1991  findgate.orig
-r--r--r--   1 winters  sf       7014 Nov  6 1991  findgate.slacvm
-r--r--r--   1 winters  sf       9153 Dec  5 1991  findgate.text
-r--r--r--   1 winters  sf        142 Dec  9 1991  fsearch.exec
-r--r--r--   1 winters  sf       4624 Nov  6 1991  fsearch.save
-r--r--r--   1 winters  sf      16495 Oct 15 1991  htdaemon.c
-r--r--r--   1 winters  sf      16492 Oct 15 1991  htdaemon.c2
-r--r--r--   1 winters  sf     130653 Dec  5 1991  htdaemon.module
-r--r--r--   1 winters  sf      14823 Nov  5 1991  htdaemon.text
-r--r--r--   1 winters  sf       7056 Oct 16 1991  httcp.c
-r--r--r--   1 winters  sf       2294 Oct 14 1991  httcp.h
-r--r--r--   1 winters  sf       6237 Nov  5 1991  httcp.text
-r--r--r--   1 winters  sf       1882 Oct 14 1991  htutils.h
-r--r--r--   1 winters  sf        202 Dec  5 1991  load.map
-r--r--r--   1 winters  sf        185 Sep 19 1991  make.exec
-r--r--r--   1 winters  sf       1224 Oct 16 1991  mycc.exec
-r--r--r--   1 winters  sf       4806 Sep 24 1991  nicfoun1.exec
-r--r--r--   1 winters  sf       4212 Oct 16 1991  q.q
-r--r--r--   1 winters  sf        675 Sep 19 1991  run.exec
-r--r--r--   1 winters  sf         98 Nov  6 1991  rundaemo.exec
-r--r--r--   1 winters  sf        131 Nov  5 1991  setup.exec
-r--r--r--   1 winters  sf       1463 Jul 21 18:58 spice192.backup
-r--r--r--   1 winters  sf        319 Dec  9 1991  spires.html
-r--r--r--   1 winters  sf        199 Dec 12 1991  spires.index
-r--r--r--   1 winters  sf       6075 Oct 16 1991  string.h
-r--r--r--   1 winters  sf       5125 Oct 14 1991  tcp.h
-rw-r--r--   1 winters  sf      11571 Nov 14 1991  terryh.tyh239
-r--r--r--   1 winters  sf       2096 Jul 22 02:56 unix-commands
-r--r--r--   1 winters  sf       5508 Jul 21 19:57 winters.filelist
-rw-r--r--   1 winters  sf      25669 Jul 21 19:11 winters.jcw120
-rw-r--r--   1 winters  sf          0 Jul 22 03:04 winters.ls
```

SCPLUNPRINTS 1992/SLACVM/[illegible]/192/[illegible]

SLAC ARCHIVES COLL [illegible]
SERIES 2 SUBSERIES 3
BOX 7 FOLDER 4

Index of /archive/1992/SLACVM/spicell/192/rl1559

Name	Last modified	Size	Description
Parent Directory	22-Jul-1998 03:12	-	
README	22-Jul-1998 03:37	11k	
a.c	16-Oct-1991 11:13	1k	
a.errs	16-Oct-1991 11:15	1k	
a.exec	01-Nov-1991 07:13	1k	
a.text	16-Oct-1991 11:15	1k	
binlist.@html	27-Feb-1992 06:36	1k	
binlist.@index	27-Feb-1992 06:36	1k	
binlist.help	06-Mar-1992 07:47	1k	
binlist.html	09-Mar-1992 03:36	1k	
binlist.index	27-Feb-1992 06:54	1k	
binlist.savindex	27-Feb-1992 06:36	1k	
bsdtypes.h	15-Oct-1991 11:01	3k	
cc2.exec	30-Oct-1991 06:18	1k	
cc_all.exec	14-Oct-1991 08:29	1k	
cmod.exec	04-Nov-1991 03:35	4k	
cms.preserve	06-Nov-1991 09:06	1k	
cpobj.rmap	16-Oct-1991 11:46	1k	
default.@html	06-Dec-1991 13:15	1k	
default.html	18-Mar-1992 08:07	1k	
defaultx.html	18-Mar-1992 08:02	1k	
dload.exec	01-Nov-1991 07:15	1k	
edcxv.c	16-Oct-1991 11:08	1k	
edcxv.text	05-Dec-1991 08:03	1k	
fearch.orig	06-Nov-1991 04:04	5k	
fget.@exec	06-Dec-1991 12:45	1k	
fget.exec	09-Mar-1992 04:38	1k	
fget.orig	06-Nov-1991 09:08	5k	
findgate.c	06-Nov-1991 09:05	7k	
findgate.c2	14-Oct-1991 08:28	7k	

findgate.orig	04-Nov-1991 03:30	7k
findgate.slacvm	06-Nov-1991 05:57	7k
findgate.text	05-Dec-1991 08:02	9k
fsearch.@exec	27-Feb-1992 06:37	1k
fsearch.exec	18-Mar-1992 08:01	2k
fsearch.save	06-Nov-1991 06:06	5k
fsearch.saveexec	27-Feb-1992 05:55	1k
hep.@html	27-Feb-1992 05:59	1k
hep.@index	27-Feb-1992 05:59	1k
hep.help	06-Mar-1992 07:47	1k
hep.html	09-Mar-1992 03:36	1k
hep.index	27-Feb-1992 05:26	1k
htdaemon.c	15-Oct-1991 11:10	16k
htdaemon.c2	15-Oct-1991 08:45	16k
htdaemon.module	05-Dec-1991 08:03	128k
htdaemon.text	05-Nov-1991 06:37	14k
httcp.c	16-Oct-1991 10:04	7k
httcp.h	14-Oct-1991 08:47	2k
httcp.text	05-Nov-1991 06:38	6k
htutils.h	14-Oct-1991 08:47	2k
load.map	05-Dec-1991 07:55	1k
make.exec	19-Sep-1991 07:06	1k
mycc.exec	16-Oct-1991 11:28	1k
nicfoun1.exec	24-Sep-1991 04:01	5k
q.q	16-Oct-1991 10:22	4k
run.exec	19-Sep-1991 07:19	1k
rundaemo.exec	06-Nov-1991 09:03	1k
setup.exec	05-Nov-1991 06:57	1k
spicell192.backup	21-Jul-1998 18:58	1k
spires.html	10-Mar-1992 02:41	1k
spires.index	12-Dec-1991 07:59	1k
stores.help	05-Mar-1992 02:43	21k
stores.html	09-Mar-1992 03:36	1k
stores.index	04-Mar-1992 07:51	1k
string.h	16-Oct-1991 10:07	6k

tcp.h	14-Oct-1991	08:47	5k
terryh.tyh239	14-Nov-1991	10:43	11k
unix-commands	22-Jul-1998	03:43	2k
winters.filelist	21-Jul-1998	20:33	7k
winters.jcw122	21-Jul-1998	20:18	32k
winters.ls	22-Jul-1998	03:39	4k

Apache/1.3.12 Server at www.slac.stanford.edu Port 80

File /afs/slac.stanford.edu/www/archive/1992/SLACVM/spicell/192/rl1559/README

On 22 Jul 1998 restored currently existing backup copy #14 (on tape cartridge RL1559) of userid SPICELL minidisk 192 via the CMS RECOVER command to WINTERS virtual reader, and from there to WINTERS 294 (a SPACE TEMPDISK) using a "RECEIVE / (OLDDATE" command within RDRLIST. The VM backup tape was taken 7/8/92.

Then copied all the RECOVERed files plus four files on the recovery and migration process from SLACVM to UNIX. Used an NFS mount?? in UNIX of the SLACVM WINTERS 294 minidisk, followed by a UNIX copy of all those files to the subdirectory /afs/slac.stanford.edu/www/archive/1992/SLACVM/spicell/192/rl1559. (A summary of UNIX commands is in file unix-commands.)

SPICELL 192 was used in very early development of the experimental WWW system at SLAC. SPICELL 192 is not an "INSTALL" disk though it appears from the at signs (@) in filetypes some files may have been RIPPLEd.

More specifically, from the CMS RECOVER list of backup tapes on 7/22/98 selected:

14) SPICELL 0192 dumped to file 170 of tape RL1559 on 07/08/92 at 04:14:00

See file spice192.backup for the entire list of currently existing backups (created from XSHOW capture of RECOVER command output).

RECOVER submitted VM batch job JCW122 from userid WINTERS on 7/22/98. The job was run shortly afterwards. Its batch console output is in file winters.jcw122.

After the recovered files were RECEIVEd, QUERY DISK A showed:

LABEL	VDEV	M	STAT	CYL	TYPE	BLKSIZE	FILES	BLKS USED-(%)	BLKS LEFT	BLK TOTAL
SPC294	294	A	R/W	20	3390	4096	67	175-05	3425	3600

There were a total of 64 files on the SLACVM SPICELL 192 minidisk of 7/8/92. In inverse chronological order since last update, here are the files (from "FILELIST (STATS" command output*):

Filename	Filetype	Fm	Format	LRECL	Records	Blocks	Date	Time	
DEFAULT	HTML	A1	V	75	10	1	3/18/92	16:07:42	SPC2
DEFAULTX	HTML	A1	V	78	12	1	3/18/92	16:02:28	SPC2
FSEARCH	EXEC	A1	V	69	63	1	3/18/92	16:01:08	SPC2
SPIRES	HTML	A1	V	74	9	1	3/10/92	10:41:55	SPC2
FGET	EXEC	A2	V	75	27	1	3/09/92	12:38:49	SPC2
STORES	HTML	A1	V	76	10	1	3/09/92	11:36:54	SPC2
HEP	HTML	A1	V	73	9	1	3/09/92	11:36:45	SPC2
BINLIST	HTML	A1	V	77	13	1	3/09/92	11:36:29	SPC2
BINLIST	HELP	A1	V	53	4	1	3/06/92	15:47:52	SPC2
HEP	HELP	A1	V	71	27	1	3/06/92	15:47:24	SPC2
STORES	HELP	A1	F	80	267	6	3/05/92	10:43:26	SPC2
STORES	INDEX	A1	F	80	10	1	3/04/92	15:51:24	SPC2
BINLIST	INDEX	A1	F	80	10	1	2/27/92	14:54:47	SPC2
FSEARCH	@EXEC	A0	V	65	13	1	2/27/92	14:37:35	SPC2
BINLIST	@INDEX	A0	F	80	9	1	2/27/92	14:36:49	SPC2

```
BINLIST   SAVINDEX A1 F        80         8          1  2/27/92 14:36:37 SPC2
BINLIST   @HTML    A0 V        73        13          1  2/27/92 14:36:14 SPC2
HEP       @INDEX   A0 V        43        12          1  2/27/92 13:59:39 SPC2
HEP       @HTML    A0 V        74         9          1  2/27/92 13:59:30 SPC2
FSEARCH   SAVEEXEC A1 V        40         7          1  2/27/92 13:55:56 SPC2
HEP       INDEX    A1 V        43        12          1  2/27/92 13:26:11 SPC2
SPIRES    INDEX    A1 V        43        12          1 12/12/91 15:59:13 SPC2
DEFAULT   @HTML    A0 V        81        13          1 12/06/91 21:15:48 SPC2
FGET      @EXEC    A0 V        60        16          1 12/06/91 20:45:45 SPC2
HTDAEMON  MODULE   A1 V     65535         5         32 12/05/91 16:03:14 SPC2
EDCXV     TEXT     A1 F        80         9          1 12/05/91 16:03:06 SPC2
FINDGATE  TEXT     A1 F        80       113          3 12/05/91 16:02:49 SPC2
LOAD      MAP      A1 F       100         2          1 12/05/91 15:55:46 SPC2
TERRYH    TYH239   A1 F       132        87          3 11/14/91 18:43:33 SPC2
FGET      ORIG     A2 V        78       132          2 11/06/91 17:08:04 SPC2
CMS       PRESERVE A1 V        65        30          1 11/06/91 17:06:02 SPC2
FINDGATE  C        A1 V        78       245          2 11/06/91 17:05:49 SPC2
RUNDAEMO  EXEC     A1 V        33         5          1 11/06/91 17:03:09 SPC2
FSEARCH   SAVE     A1 V        76       123          2 11/06/91 14:06:50 SPC2
FINDGATE  SLACVM   A1 V        78       245          2 11/06/91 13:57:17 SPC2
FEARCH    ORIG     A1 V        76       125          2 11/06/91 12:04:44 SPC2
SETUP     EXEC     A1 V        62         4          1 11/05/91 14:57:21 SPC2
HTTCP     TEXT     A1 F        80        77          2 11/05/91 14:38:12 SPC2
HTDAEMON  TEXT     A1 F        80       183          4 11/05/91 14:37:48 SPC2
CMOD      EXEC     A2 V        68       203          2 11/04/91 11:35:12 SPC2
FINDGATE  ORIG     A1 V        78       245          2 11/04/91 11:30:12 SPC2
DLOAD     EXEC     A1 V        64         4          1 11/01/91 15:15:36 SPC2
A         EXEC     A1 V        68        10          1 11/01/91 15:13:32 SPC2
CC2       EXEC     A1 V        58         8          1 10/30/91 14:18:05 SPC2
CPOBJ     RMAP     A1 F        80         3          1 10/16/91 18:46:59 SPC2
MYCC      EXEC     A1 V        63        51          1 10/16/91 18:28:29 SPC2
A         ERRS     A1 V        49         1          1 10/16/91 18:15:35 SPC2
A         TEXT     A1 F        80         5          1 10/16/91 18:15:35 SPC2
A         C        A1 F        80         5          1 10/16/91 18:13:27 SPC2
EDCXV     C        A1 F        80         5          1 10/16/91 18:08:43 SPC2
Q         Q        A1 F        80        52          2 10/16/91 17:22:58 SPC2
STRING    H        A2 F        80        75          2 10/16/91 17:07:43 SPC2
HTTCP     C        A1 V        79       282          2 10/16/91 17:04:54 SPC2
HTDAEMON  C        A1 V        85       639          5 10/15/91 18:10:19 SPC2
BSDTYPES  H        A1 V        70       120          1 10/15/91 18:01:30 SPC2
HTDAEMON  C2       A1 V        85       639          5 10/15/91 15:45:38 SPC2
HTTCP     H        A1 V        78        94          1 10/14/91 15:47:38 SPC2
TCP       H        A1 V        91       180          2 10/14/91 15:47:28 SPC2
HTUTILS   H        A1 V        79        85          1 10/14/91 15:47:21 SPC2
CC_ALL    EXEC     A1 V        81        13          1 10/14/91 15:29:29 SPC2
FINDGATE  C2       A1 V        78       245          2 10/14/91 15:28:41 SPC2
NICFOUN1  EXEC     A2 V        78       125          2  9/24/91 11:01:53 SPC2
RUN       EXEC     A1 V        74        21          1  9/19/91 14:19:57 SPC2
MAKE      EXEC     A1 V        51         6          1  9/19/91 14:06:07 SPC2
```

* As defined in "HELP CMS FILELIST", the STATS option produces the following output:

STAts
lists the following information about the specified files:

- o File identifier or directory identifier
- o Format and logical record length of the file
- o Number of records and number of blocks in the file
- o Date and time of last update.

See the "Examples" for a sample display using the STATS option.

In April 1992 the WWW and WWWTEST 192 minidisks were set up for pages as INSTALL disks. From the dates on the pages and some related files here, the development activity seems to have moved from SPICELL 192 to the WWW[TEST] 192 minidisks.

I haven't found any ownership noted in the files.

For a list of all the files recovered plus three files created related to the recovery see winters.filelist (created by SAVEing the output of a FILELIST command). A list of all the files on the backup tape (which seems to be the same as all the files recovered) is in the batch job output winters.jcw122.

N.B.: CMS PRESERVE is still a record of the enviroment in 11/91 when WWW was being brought up at SLAC, not a record of the migration from SLACVM to UNIX in 1998.

For a list of all the files moved to UNIX (which should include all the files in the winters.filelist file), see winters.ls (which also lists the unix-commands file, plus).

In summary, for more details about the recovery from VM backup and move from SLACVM to SLAC UNIX, see the following files:

README [created in SLACVM about entire migration, named README RL1559]

spicell192.backup [in SLACVM named spice192.backup]
winters.jcw122
winters.filelist

unix-commands [created in UNIX]
winters.ls

See also files in subdirectory /afs/slac/www/archive/SLACVM for generally applicable information.

22 Jul 1998

Joan M. Winters

```
main (argc)
  int argc;
{
  printf("here here");
}
```

```
Warning! line 5: Symbol 'printf' assumed external
```

```
/* REXX */
   trace r

  ADDRESS COMMAND

/* SLAC Additions  */
'EXEC VMSTATUS PRESERVE'
/*'GLOBADD TXTLIB EDCBASE IBMLIB CMSLIB COMMTXT VSPASCAL AMPLANG' */
 exec GLOBADD TXTLIB EDCBASE IBMLIB   /* terryh */
'EXEC GLOBADD LOADLIB EDCLINK'
```

SLACVM BINLIST Telephone and e-mail Directory

Find
: Find NAME last-name, {,first-name}

Help
: Help with BINLIST

You can search this index. Type the keyword(s) you want to search for:

SLAC Phone and E-mail directory search.

Use standard SPIRES search terms such as...

```
find name addis
find name crane
```

```
This is the SLAC directory containing such items as
names phone numbers, vm accounts and office locations
```

SLACVM BINLIST Telephone and e-mail Directory

Find

Find last-name, {,first-name}

Help

Help with BINLIST