Locally Developed Utilities Simplify Cray User Interface

Donald Moore, Hughes STX for the NASA Center for Computational Sciences, NASA/GSFC, Greenbelt, Maryland

ABSTRACT: *The NASA Center for Computational Sciences (NCCS) at Goddard Space Flight Center (GSFC) has developed a suite of Cray-resident utilities for use on their Cray C98. The purpose is to (1) simplify the display of information and (2) facilitate communication between the Cray and other NCCS platforms, namely an IBM 9021 and a Convex mounted UniTree mass data storage system. Software design emphasis centered on portability, maintainability and ease of use. The utilities fall under the general categories of information display and data transfer. The interplatform utilities in the suite are either FTP or Cray Station Software (CSS) based. Most of the utilities are written in Bourne or Korn shell script and are written to resemble the conventional UNIX command structure. In most cases they are executable in either batch or interactive mode.*

Who We Are

The NCCS supports NASA funded space and earth sciences research by providing its user community with three major computational and data storage platforms: a Cray C98 Supercomputer, a Convex-based UniTree mass data storage facility and an IBM 9021 mainframe. The Technical Assistance Group (TAG), along with other NCCS personnel have developed a suite of utilities to be used by the user community.

The Development Philosophy

The NCCS user utilities adhere to a development philosophy that ensures their usefulness over an extended period of time. That is, that locally developed utilities should share the following characteristics: (1) consistency, (2) portability--where applicable, (3) UNIX-like execution, (4) interactive and batch execution capabilities, (5) maintainability.

Consistency--Consistency is achieved by keeping command line options consistent (as much as practical) throughout the various utilities. In other words, a -u or -i option would perform identical (or similar) functions from one utility to the next. Also, many data transfer utilities record their activities in a log file and optionally utilize a parameter defaults file (much like a .netrc file). The log file and defaults file are common to most of the NCCS FTP-based utilities.

Portability--Virtually all of the UNIX-resident TAG utilities are written in Bourne shell script. Those that interface with non-UNIX platforms (i.e. the IBM mainframe) are usually a combination of Bourne shell script and the foreign machine's system specific language(s). For FTP-based utilities the Bourne shell scripting language allows them to be easily ported to other UNIX platforms. The NCCS Bourne shell scripts are written primarily for machines which are System V based (as is the Cray) but can usually be easily modified to run on Berkeley-based platforms. Some early utilities were written in C (before the ANSI standard was accepted) and proved to be frustratingly non-portable. Hence the adoption of the practice of coding in the Bourne shell.

UNIX-like execution--Where utilities contain command line options, they follow the conventions of other UNIX commands. Command line options are generally one character preceded by a dash. Options that do not take arguments can be grouped together following a single dash. Lower case characters are preferred over upper case. If a user enters an incorrect option or enters no options, the utilities will typically display the syntax and a list of all available options. Also included in the display is an explanation of how each option is used. In this manner they diverge from UNIX de facto standards as a typical UNIX command does not produce a usage display when no options are entered nor does it give explanations of the purpose of each option (see figure 1). Nearly all of the local utility names are entirely lower case which is also consistent with the UNIX de facto standards.

Interactive/Batch execution--Most of the utilities can be executed either in batch mode or interactively. Even though many were developed for execution in batch jobs, the interactive option provides a means for access to the same functionality without the need to memorize a string of options. If a user

selects the interactive option he/she is prompted to enter all of the parameters required to execute the utility. The interface to all existing utilities is a character prompting and entry mechanism. None, as yet, contain a Graphical User Interface (GUI).

Maintainability--The TAG utilities are usually maintained by the utilities' authors. Every effort is made to make certain that no values such as pathnames, field numbers or timing values are buried deep into the code. All such values are stored in variables in an initialization section at the beginning of the code for easy modification. All code contains in-line comments. Code pieces which perform specific, self-contained actions are broken into Bourne shell functions. This, in truth, is more often the case for the later utilities than for those that were written early. The unpleasant experience of attempting to modify code that was not cleanly structured from the start has served as a powerful source of inspiration.

A Short History

The configuration of our facility has created the need for several specialized user utilities. Our three major platforms present an environment with two distinctly different operating systems: UNIX and IBM MVS. UniTree, our mass data storage system, resides on a Convex 3830 running a Berkeley version of the UNIX operating system and is accessible only via FTP. Our Cray C98 running the UNICOS operating system (based on UNIX System V) also supports the Cray Data Migration Facility (DMF) silo tape system.

Before 1990 the NCCS was primarily IBM-based and additionally supported a CYBER 205 supercomputer front-ended by an IBM mainframe. The arrival of a Cray Y-MP in 1990 resulted in a large pool of users who were not UNIX literate but who needed to do work on the new UNIX-based supercomputer. The facility also was moving away from hard-wired IBM terminals toward UNIX workstations. Thus, early utilities were geared toward a user community to whom UNIX was an unfamiliar environment. Also new to many of the users was the concept of FTP data transfer. The early utilities sought to address these issues. As the user community became more UNIX literate and UNIX-based workstations continued to proliferate, the needs for software tools also changed. As users learned more about the UNICOS environment, they wanted more information. As the NCCS added more hardware and software, the users needed convenient ways to access these new facilities and to gather information about them.

Addressing The Needs

NON-UNIX USERS

The first major utility attempted to provide tools that would perform much of the UNICOS detail work for the users. It was designed to interactively solicit a user for the parameters necessary to build a batch script and submit a Cray job. It was written with a combination of C code and Bourne shell script. Unfortunately, the time and complexity involved in developing this utility was greater than the learning curve of most of the user community and therefore, was never implemented. It did, however, serve as an instructional device for those users who wanted to see examples of certain types of UNIX functionality in a batch script such as job chaining.

STATUS CHECKING

Users whose batch work depends on resources such as DMF or files on remote FTP nodes need to determine if those resources are available before beginning processing. Therefore, the NCCS developed three "up or down" type status utilities for use in batch scripts: utstat, dmfstat, testnode.

Often, users will have a number of large files stored in the UniTree system which must be retrieved before running a batch job on the Cray. Likewise, large output files created during a job run often need to be stored to the UniTree system. The utility, utstat, functions by executing a noop (no operation) command to the UniTree system and checking for a character string response which indicates whether or not the noop was successful. Issuing a noop does not require a password and hence lessens any security risk concerns. If an error is returned from the noop or the system fails to respond within two minutes, a non-zero code is returned. If the noop is successful, a zero is returned. The user's code must test the return code value and proceed accordingly.

The NCCS Cray C98 DMF provides a short term storage area for users where files may be stored for up to an NCCS imposed limit of 35 days. In order to actively migrate files from the /silo disk cache to tape or to recall the files from tape to disk, the DMF subsystem must be operational. The dmfstat utility was developed to check the health of the DMF daemon. It returns a zero value if the DMF daemon appears as an active system process indicating that DMF processing is alive and well. It returns a non-zero value if the DMF daemon cannot be found. Used in a batch job script a user can then determine whether or not DMF services are available at any given time and put their job in a sleep state periodically checking for DMF availability if so desired.

When files must be sent to or retrieved from remote FTP nodes as part of a batch job, it is prudent to first check and see if the node is reachable. Verifying that a critical remote node is up prevents jobs from hanging on an unreachable resource. The testnode utility performs a simple ping to the remote node and looks for a response in a string of characters that indicates that the remote node is up. A successful ping then causes testnode to produce a zero return code. An unsuccessful ping results in a non-zero return code. The testnode utility operates only in batch mode since interactive pings are easy to do.

UNITREE INTERFACE

The UniTree mass data storage system can be accessed only via FTP and supports only a subset of UNIX commands. Several utilities have been developed in order to make user access more powerful. In particular, users can only change the access permissions of UniTree files one at a time. The UniTree command set does not provide a mechanism for producing a

comprehensive list of files under any given subdirectory or the amount of space they occupy. It is equally difficult to delete a range of files once a list is compiled as UniTree only provides a mechanism for deleting files one at a time. Therefore, users often find it difficult to keep their UniTree storage area clean. Lastly, users need an easy way to determine whether or not their data files have been successfully transferred to UniTree.

The utch utility allows users to change the permissions for all of the files under any UniTree subdirectory. It can also change the group to which a subtree of files belong. The utch utility provides this capability by recursively traversing a user specified UniTree subtree, building the UniTree equivalent of a chmod command for every file in the tree. This same methodology applies for changing group ownership of UniTree files.

The netrdir and utlist utilities provide a comprehensive listing of all files under a user specified subdirectory. The utlist utility works only on UniTree files where netrdir provides a comprehensive listing for any UNIX-based FTP site and also reports the total amount of storage occupied under each subdirectory visited. A grand total of all file space occupied appears at the end of the subtree traversal. The utremove utility is a companion to utlist and enables mass deletions from UniTree. Users place a mark ($) before each file entry in the output file generated by utlist and run the utremove utility. It will then generate the FTP commands to delete the marked files. See figure 2.

The utfcheck utility determines whether or not a user's file was successfully transferred to the UniTree system. It compares file sizes on the Cray to those transferred to UniTree. If the sizes do not match (or the filenames do not appear), an error message is generated and a non-zero value returned.

(The utstat utility has been covered under status checking utilities.)

IBM INTERFACE

Many UNIX users interact with the IBM MVS system on a very limited basis using it only as a storage medium or to read in or scan tapes. Many of these users are not IBM MVS literate. In order to alleviate, as much as possible, the detailed knowledge needed to work with the IBM MVS system, NCCS staff have created a suite of Cray/IBM interface utilities.

--Handling IBM Tapes

The IBM MVS system enjoys a reputation for excellent tape handling capabilities. Also, much data stored on tape media in the NCCS facility was generated on IBM machines often in IBM standard label format. Users often need to format IBM MVS tapes in preparation for storing data. They also need to store data to tape, read data from tape or determine what information currently resides on a tape and determine the tape's format.

In order to address these requirements, NCCS personnel developed the initT, anlyzT, nettape, fet and dispo utilities which provide an interface to the IBM MVS tape system from the Cray C98. All but the nettape utility are based on Cray Station Software (CSS). The fet and dispo utilities simplify user interaction with the fetch and dispose CSS commands to send data to or retrieve it from tape. If run interactively, users are prompted for the parameters necessary to describe the tape, the files, their positions on the tape, etc. and builds the required MVS text string within the dispose or fetch command. If run via command line execution, these values are either left to default or are entered on the command line.

The nettape utility is used only for MVS tape file retrieval. It submits a job to the IBM MVS system via FTP which initiates a tape copy using the IBM system program IEBGENER. Once the files are copied to the respective data sets, an FTP job is initiated on the IBM side which in turn sends the files back to the Cray to a user-specified directory.

The initT and anlyzT utilities perform tape initialization and tape scanning operations. Also, CSS-based, they solicit the user (if run interactively) for the parameters which identify the tape and the processing to be performed and building the JCL which performs either a tape labeling or a tape scanning job. The dispose command then submits the job to the IBM's job input queue. When processing is complete a second job is spawned on the IBM which notifies the user via electronic mail, explaining how and where to retrieve the output. See figures 3 and 4.

--IBM Disk Data Transfer

The dispo and fet utilities mentioned above also provide CSS-based access to IBM disk. When users wish to dispose or FTP data to the IBM, they must first determine the amount of disk space to allocate. The dispo utility automatically figures the size of the file(s) in bytes to be stored and requests the appropriate size in tracks from the IBM system. The netpush, netpull and netjump utilities will also handle disk data transfers to and from the IBM MVS system. The netpush utility will, like the dispo utility, automatically calculate the track size and allocate it on the IBM MVS system. These three utilities are explained in greater detail below.

--Other IBM Interaction

Other kinds of interaction with the IBM include obtaining a listing of IBM files for a given user's ID (lsmvs), deleting IBM files (rmmvs), remotely creating an IBM MVS partitioned data set (makepds), printing Cray files on the high speed IBM laser printer (prt3800), changing IBM file format characteristics (zapback), recalling archived data and listing archived data (lshsmm, lshsmb and rclmvs). NCCS staff have created these utilities to handle all of these kinds of interactions remotely. They are all either FTP-based or CSS-based. See the complete listing of all NCCS utilities at the end of this document.

BATCH DATA TRANSFER

The NCCS staff have developed three major utilities: netpush, netpull and netjump designed as batch (and interactive) FTP interfaces. The netpush and netpull utilities were designed to send/retrieve data from other FTP reachable platforms. Other than the batch (command line) capabilities they

provide an interactive interface which prompts users for parameters necessary to identify the remote file(s) to send or retrieve. If the user has defined a defaults file (much like the commonly available .netrc file), they will display a menu of predefined nodes, user ID's and directory paths which the user can select. A defaults file also allows users to perform routine transfers without the need to enter all information on the command line or interactively. Options are also provided to handle IBM MVS specifics such as EBCDIC/ASCII character data translation, record format interpretation, record control word preservation and automatic preallocation of IBM MVS file space.

The netjump utility uses the netpull and netpush utilities as its building blocks. It simulated proxy FTP before true proxy FTP was widely available. This was accomplished by invoking netpull to stage the data across the "home" node from the first remote node and using netpush to the send the data to the second remote node. A user defined defaults file is mandatory for netjump execution in simulated proxy FTP mode. The netjump utility now includes an option to invoke true proxy FTP where it is available.

Any utilities that are named with the net prefix, log FTP session information to a special file created in the user's home directory. Because these utilities were designed with portability in mind they can often be ported to a user's workstation. This way they can be run without incurring the expense (or the system impact) of running them on the Cray.

DISPLAY SYSTEM INFORMATION

Some of the Cray system commands produce output that is hard to interpret or visually confusing. In other cases, users want information concerning the status of their accounts or of system resources. The NCCS provides a number of utilities to address these issues.

Two examples of utilities that reformat Cray system command displays are showqs and dspace. The showqs utility takes the output from the qstat -a command and organizes it so that the display is easier to read and understand (see figure 5). The dspace utility reformats the output from the Cray quota command so that users can see their remaining disk space in bytes rather than blocks (see figure 6).

The dmcounts utility displays the number of dmf recall and migration requests waiting in queue. This information helps users to determine the load on the DMF thereby enabling them to identify a potential processing bottleneck and scheduling their DMF interaction for low usage times (see figure 7). The cusage utility displays the amount of Computing Units used for a given account. This helps users determine the efficiency of their resource utilization and helps them estimate the amount of computing resources required to perform their tasks.

KEEPING USERS INFORMED

It is important to keep users abreast of any recent developments which could affect their batch or interactive work. The NCCS staff employs two utilities, motdup and statup, to the keep user community informed of the current system status, scheduled system outages, new online documents and the like. These utilities run on Sun workstations and, via FTP, update files on the Cray as well as on other NCCS platforms. The motdup utility modifies the /etc/motd file. Any user logging on to an NCCS platform will see the latest system information. The statup utility modifies a file which requires users to issue a special command, statinfo, to display it. During regular business hours, the information displayed by statinfo provides up-to-date status information for all NCCS systems. See figure 8 for an example of the statinfo display.

Current Trends/Future Plans

As more standards are developed and better and more diverse methods of implementation become available the greater the opportunities for utility writers to find a good match between implementation and the desired results. The establishment of the C language ANSI standard and the wide availability of the PERL script-like programming language have added two powerful resources for building utilities. Also, some special needs are too complex to be handled by one homogeneous, self-contained implementation. At the NCCS some of the latest utilities are now being written in PERL. PERL offers the pattern matching capabilities of awk with many more features, faster execution and greater capabilities. Execution of PERL code also tends to be more consistent than awk across various UNIX platforms. The utrefam utility is an example of a PERL written NCCS utility. It is an interface to the UniTree mass data storage system which allows users to change file families and to optionally make file copies within a UniTree file subtree.

The most dramatic example of a complex and powerful utility is the tapex/ TapeImport utility. The tapex utility exports files to 4mm DAT tape, 8mm Exobyte, 3480/90 cartridge tape from UniTree, the Cray C98 or the IBM 9021. There is also a TapeImport interface to tapex which allows data to be imported from 4mm DAT, 8mm Exobyte, 9-track reel style tape, and 3480/90 cartridges to the Cray C98, or UniTree. There is a tapex interactive interface on the Cray, the IBM and the Convex (where UniTree is mounted). The tapex software consists of four separate shell scripts (Bourne and C shell) and a C interactive interface on the Cray and the Convex. The IBM tapex interface is written in REXX. There is no TapeImport capability on the IBM 9021. Because of the complexity and the unique nature of the software it is not portable. The tapex/TapeImport utility fulfills a critical need for users who need to move their data files out of or into the UniTree, Cray or IBM systems via non-IBM tape media.

In November of 1994 the Cray Station Software was removed from the NCCS Cray C98. This was done in anticipation of the removal of facility's IBM 9021 at the end of 1996. The removal of the CSS eliminates the need for some CSS-based utilities. Others such as initT and anlyzT have already been converted to run on the FTP-based protocol.

The remaining non-CSS-based utilities will be enhanced, modified or obsolesced as the facility continues to evolve. Some will be converted to PERL or C.

Future utilities are likely to be developed in PERL and C code except where a short, simple Bourne or Korn shell script will suffice. Every effort will be made to keep future utilities portable, maintainable and functionally consistent with existing utilities. NCCS staff is currently experimenting with World Wide Web applications which can read image files directly from the UniTree mass data storage system. A Web-based application is also being explored for simple UniTree file retrieval in place of the traditional user interface with FTP.

Lessons Learned

- The lessons learned from user utilities development over an extended period of time can be summed up as follows:
- Users will develop their own utilities or establish methods for handling problems if a utility is not forthcoming to address the problem in a timely fashion.
- Utilities should be easy to use. If you have made a difficult interface more difficult, a confusing display more confusing or a complex situation more complex, you have not achieved anything. Go for simplicity in display and function.
- Stay consistent with your options and naming conventions so as not to confuse your users.
- If possible, make the code for your utility portable. It allows users to offload mundane processing from the Cray. This benefits all Cray users.
- Utilities need not be complex to be useful. A simple script of only a few lines can be just as valuable as a complex piece of code if it makes someone's job easier.
- Take the extra time to make your utilities maintainable. It will save you much time and effort in the long run. Do not write anything you do not want to maintain.
- Recognize when users are having problems with your utility. Appreciate their objectivity. Pay attention to user feedback and change functionality when it seems prudent to do so. Follow your user's suggestions and advice. Offer guidance to users on how to best use your utilities and suggest specific implementations for including your utilities that will make their jobs easier.
- Do not make your utilities your pets. When better methods become available, show your users how to use them and let your utilities die.
- Make sure that you have good online documentation and provide plenty of examples. Make sure you have clear, concise man pages for each utility which include at least one good example. Examples are the best way to demonstrate how a utility can and should be used.
- Publicize your utilities. Users cannot use utilities if they do not know they exist. Make sure that other colleagues who are in contact with users know about your utilities and how to use them. Give demos to your fellow workers and to users when and where possible. If you have a mechanism for announcing system changes, make sure modifications to utilities or newly developed utilities are included in the announcements.
- Recognize when a utility can improve a situation and write one.
- Share information with your users and your co-workers. It is a reciprocal thing. Make your code accessible to the user community so they can learn from your code or use all or a part of it if they wish. You may get your code back with an improvement that you can use.

More Information About Our Facility

Online information about our facility is available via gopher or the World Wide Web:

World Wide Web URL: http://nccsinfo.gsfc.nasa.gov/NCCS
Gopher Address: nccsinfo.gsfc.nasa.gov

```
100: FAROUT >utlist

 Usage: utlist [-loedupnf] [DMY] | [T]

    -- options with no arguments --
          -l -- disable auto logging to .netutil.log
          -o -- output to stdout as well as to file
          -e -- echo ftp commands to stdout

    -- options that require arguments --
          -d -- UniTree directory
          -u -- UniTree ID and node name (e.g. usrid@dirac, usrid@dirac-h
          -p -- UniTree password
          -n -- defaults file index number (default: 1)
          -f -- output listing name (default: UT.removelist)
          -D -- generate a list of UniTree files dated > n days
          -M -- generate a list of UniTree files dated > n months
          -Y -- generate a list of UniTree files dated > n years
          -T -- generate a list of UniTree files predating a date entered
                as Mmm/dd/yy (e.g. Aug/8/93) or mm/dd/yy (e.g. 8/8/93)

      Note: Option groups DMY and group T are mutually exclusive

101: FAROUT >
```

Figure 1: Example of a local utility's usage display.

```
:ime of request: Mar 09 18:28   1995

.les not accessed since Mar 9, 1995


.rac -- UniTree
:y: "/u1/xrdbm/testdir"

 1 xrdbm   xstat  AR  common       415 May  7  1
 1 xrdbm   xstat  AR  common  39305413 May 31  1
 1 xrdbm   xstat  AR  common   4655000 May 20  1
 1 xrdbm   xstat  AR  common   4866048 May 20  1
 1 xrdbm   xstat  AR  common      1136 Aug 25  1
 1 xrdbm   xstat  AR  common        37 Aug  6  1
 1 xrdbm   xstat  AR  common      1442 Feb 24  1
 1 xrdbm   xstat  AR  common       768 Feb 24  1

= = = = = = = = = = = = = = = = = = = = = = = =

.rac -- UniTree
:y: "/u1/xrdbm/testdir/subtst1"

  1 xrdbm  xstat  AR  common       165 May 27  1
  1 xrdbm  xstat  AR  common         4 May 27  1
  1 xrdbm  xstat  AR  common       145 May 27  1

= = = = = = = = = = = = = = = = = = = = = = = =

.rac -- UniTree
:y: "/u1/xrdbm/testdir/subtst1/awkdir7"

  1 xrdbm  xstat  AR  common      1593 Aug  9  1
  1 xrdbm  xstat  AR  common      1789 Aug  9  1
```

Figure 2: UniTree files marked with "$" for deletion.

```
55: CRAZED >anlyzT -i

 Tape to analyze is a Reel or Cartridge tape ?  (r or c) > c

 Enter the Volume ID of the tape > dbm006

 Long or Short output ? (l or s    default = "s") > s

 Do you want to run the job high priority, Class N ?  (y or n) > n

 Print tape analysis output on the IBM 3800 printer ?  (y or n) > y

56: CRAZED >
```

Figure 3: Interactive prompting to analyze a tape.

```
Message 44/56  From XRDBMMS@GIBBS.GSFC.NASA.GOV        Sep 12, 94 11:55:00 am EDT

Return-Path: <XRDBMMS@GIBBS.GSFC.NASA.GOV>
Date:    Mon, 12 Sep 94 11:55 EDT
To: XRDBM@CHARNEY.GSFC.NASA.GOV
Subject: TAPE ANALYSIS REQUEST


 Analysis complete for IBM/MVS tape volume -- BAKER1
 Output stored to IBM/MVS file on scratch disk
 Filename: XRDBM.TS114939.OUTLIST

 -- use "fet" utility to retrieve output to local file

  Enter: "fet <local_name> XRDBM.TS114939.OUTLIST c"
  or Enter: "fet -i" for interactive prompting

 For help contact the Technical Assistance Group:
   (301) 286-9120   NASA/GSFC, CODE 931

Command ('i' to return to index):
```

Figure 4: Utility anlyzT sends email to user.

```
NQS 80.45 BATCH REQUEST SUMMARY
------------------------------
IDENTIFIER     NAME     USER     QUEUE                JID  PRTY REQMEM REQTIM ST
-------------- -------- -------- -------------------- ---- ---- ------ ------ ---
95743.charney x0054A  w3mga    c_120@charney        1306   28  43048   3830 R07
95765.charney e1004xan wlkwe    c_120@charney             ---  92160  21600 Ho
p
96048.charney e0003xan y310c    c_120@charney             ---  61440  21600 Ho
p
96173.charney x0407d  w3ygb    c_120@charney             ---  38912  21600 Hop
96194.charney complink zcjjm    bug60@charney        5472   21   1575     87 R0
3
96110.charney phx.jul w3wes    sh_15@charney        4169   28    502    294 R04
96112.charney phx.aug w3wes    sh_15@charney        4191   28    619    224 R03
96121.charney phx.sep w3wes    sh_15@charney        4381   28    502    294 R04
96124.charney phx.oct w3wes    sh_15@charney        4517   28    502    294 R04
96133.charney qgetem  wljaw    sh_15@charney        4567   28    644    595 R04
96156.charney jmain   detmd    sh_15@charney        5096   28   8158     14 R03
96098.charney diagu_9307_ts w3mga    sh_15@charney        5301   28   2862     5
53 R04
96182.charney phxavg  zwrxy    sh_15@charney        5346   28    911    593 R07
96186.charney obs_count wldvl    sh_15@charney             ---  15360    600 H
op
96143.charney craythetaeinlik wsajl    sh_30@charney        5459   28  18445
 570 R03
96130.charney g861208 wljaw    sh_30@charney             ---  24576    600 Hop
96132.charney g861210 wljaw    sh_30@charney             ---  24576    600 Hop
96131.charney g861209 wljaw    sh_30@charney             ---  24576    600 Hop
96084.charney xrtjsno xrtjs    m_15@charney         4082   28   9418    735 R04
96117.charney front_case yfc0c    m_15@charney         4681   28    808   1412
R04
96146.charney rans80_1fall.co zmlll    m_15@charney         5126   28  10774
1531 R03
96167.charney e201x   wlfxc    m_15@charney         5200   28   5691   2131 R03
96176.charney job     wts0n    m_15@charney              ---   8192   2400 Hop
```

```
ID     PROGRAM          USER   QUEUE      JID    PRTY  REQMEM  REQTIM  ST
-----  ---------------  -----  ---------  -----  ----  ------  ------  ---
95743  x0054A           w3mga  c_120       1306    28   43038    5107  R07
95765  e1004xan         wlkwe  c_120              ---   92160   21600  Hop
96173  x0407d           w3ygb  c_120              ---   38912   21600  Hop
96110  phx.jul          w3wes  sh_15       4169    28     502     294  R04
96112  phx.aug          w3wes  sh_15       4191    28     502     294  R04
96121  phx.sep          w3wes  sh_15       4381    28     502     294  R04
96124  phx.oct          w3wes  sh_15       4517    28     502     295  R04
96133  qgetem           wljaw  sh_15       4567    28     632     597  R04
96156  jmain            detmd  sh_15       5096    28    8158     416  R03
96094  prog_9307_ts     w3mga  sh_15       5112    28    2862     555  R04
96157  TER              zlgsl  sh_15       5155    28     410     527  R03
96162  writeTOVS2_HDF.  zmj0z  sh_15       5168    28    7335     557  R03
96177  t022             y310d  sh_15              ---    3072     600  Hop
96129  g861207          wljaw  sh_30       5056    28     632     403  R05
96138  svd.sub          ybh0w  sh_30              ---   30720     300  Hop
96143  craythetaeinlik  wsajl  sh_30              ---   19456     600  Hop
96084  xrtjsno          xrtjs  m_15        4082    28    9378    1073  R04
96117  front_case       yfc0c  m_15        4681    28     808    1814  R04
96146  rans80_1fall.co  zmlll  m_15        5126    28    1508    1952  R05
96167  e201x            wlfxc  m_15        5200    28     343    2399  R03
96178  xrtjsre          xrtjs  m_15              ---    4096    2400  Hop
96100  rexp552          yfz0y  m_30        4140    28    8542    1585  R03
96165  enoaal0          zmj0z  m_30              ---   30720    2400  Hop
95723  qgmodg           ybmmr  l_15         190    28    6775   12754  R03
95724  qgmodg           ybmmr  l_15        2791    28    6779   17812  R03
95751  obs_count        wldvl  l_15        4834    28   10722    6513  R03
95755  GSMM_55          y3jen  l_15              ---    4096    4567  Hop
95746  MM5v0            zlgsl  l_30        1194    28   14654    8435  R05
95780  e0462df          w3jmt  l_30              ---   28672    8000  Hop
95876  xdvl057          wldvl  l_60              ---   51200   21600  Hop
96179  MM5TOCEM         zlgsl  bug120      5204    21    6642     599  R03
95500  qgmodc           ybmmr  h_60       38985    28    6755   20503  R03
96033  a.R013.com       www0j  h_60              ---    8192   72000  Hop
-----  ---------------  -----  ---------  -----  ----  ------  ------  ---
```

Figure 5: qatat -a compared to showqs utility output.

```
58: CRAZED >quota
File system: /ul
 User: xrdbm, Id: 5228
  ** User warning time: Thu Mar 10 10:31:04 1994
                      File blocks                 Inodes
     User Quota:            10240  ( 95.0%)  Unlimited*
        Warning:             7680  (126.6%)        None*
          Usage:             9724                  1836

59: CRAZED >
```

```
55: CRAZED >dspace -a

 File system: /ul
 User: xrdbm
 Disk space (bytes):    Allotted          Remaining
                        --------          ---------
                        41,943,040        3,264,512

56: CRAZED >
```

Figure 6: Quota compared to dspace utility output.

```
61: CRAZED >dmcounts
                                        Current Maximum
15:03:46 Counts -  backup                 26     186
15:03:46 Counts -  recall                  0     124
15:03:46 Counts - krecall                  1      14
62: CRAZED >
```

Figure 7: Utility dmcounts shows dmf queues.

```
11: CRAZED >statinfo

==========================================================================

The following information is updated during regular business hours.
Call (301) 286-1392 for status information updated 24 hours a day.

02/13/95 16:10  The Versatec plotter is down due to hardware problems
                until further notice.

02/14/95 22:08  One STK silo tape drive experiencing a hardware problem.
                The drive is off-line until a STK CE can look at it
                tomorrow.

==========================================================================

12: CRAZED >
```

Figure 8: Example of statinfo display.

```
Overview of NCCS Cray Local Utilities (9k)                          19% 
                                                    last updated:  02/16/95

NCCS Local Utilities Installed on Unix Systems
==============================================

The following is a list and a brief description of utilities that are
available on NCCS Unix systems.  They are installed on both charney and
dirac unless otherwise noted.  These utilities are intended to make
frequently used functions easier to perform and remove the need to
learn non-Unix operating system specific parameters.  More information
on each of these utilities is available via man pages.

NCCS Developed File Transfer Utilities
--------------------------------------
netpush     Send one or more files, via batch ftp, to another Unix node
            or to the NCCS IBM MVS system from the NCCS Cray or from any
            AT&T system V or Berkeley 4.3 Unix platform.

netpull     The compliment of netpush, for file retrieval.

netjump     Simulates a proxy ftp by first invoking netpull, thereby
            staging one or more files to the home node, then invoking
            netpush to transfer the file(s) to the second remote node.
            The staged files are then deleted from the home node.
            The netjump utiltiy also has the capability to invoke true
            proxy ftp on systems that support it.

tapex       Export files to DAT or 8mm tape.

TapeImport  Import files from DAT, 8mm, 9-track, 6250 or 3480/3490
            cartridge tapes.


NCCS UniTree Specific Utilities
-------------------------------
utstat      A script which checks to see if UniTree (either via ethernet
            or UltraNet) is available.  The utility returns a non-zero
            code if access is not available or a zero if it is available.
            It can be used either in batch or interactive mode.

utfcheck    Check and see if files ftp'd to or from UniTree arrived
            intact.

utlist      Recursively list files on UniTree which are older than a
[Help: ?]  [Exit: u]  [PageDown: Space]
```

```
Overview of NCCS Cray Local Utilities (9k)                          38% 
            specified age or date.

utremove    Remove files from UniTree which are tagged by the user to
            be removed.

utch        Perform chmod or chgrp operations recursively on a UniTree
            file sub-tree.


NCCS MVS Specific Utilities
---------------------------

lshsmm      builds and submits a batch job that runs on Gibbs
            which essentially executes a locally written clist to
            create a file containing a list of the user's datasets
            which are in level 1 migration, or "migrated."  This
            file is then mailed to the user at charney.  The
            userid for the job is taken to be the userid under
            which the lshsmm command is issued.  This utility takes
            no arguments.

lshsmb      builds and submits a batch job that runs on Gibbs
            which essentially executes a locally written clist to
            create a file containing a list of the user's datasets
            which are in level 2 migration, or "backed up."  This
            file is then mailed to the user at charney.  The userid
            for the job is taken to be the userid under which the
            lshsmb command is issued.  This utility takes no arguments.

makepds     Remotely create a partitioned data set (PDS), via batch ftp,
            on the NCCS IBM MVS system.  Available only on the Cray.

subMVS      Remotely submit a job to the NCCS IBM MVS system from the
            NCCS Cray.  Available only on the Cray

rmmvs       Remotely remove files from the NCCS IBM MVS system.
            Available only on the Cray.

lsmvs       List all of the files which belong to the current userid on
            the NCCS MVS system from the NCCS Cray.  Output will be
            mailed back to the user on the Cray.  Available only on the
            Cray.

rclmvs      Recall an MVS file from HSM migration to an MVS disk.
            Available only on the Cray.

[Help: ?]  [Exit: u]  [PageDown: Space]  [PageUp: b]
```

```
Overview of NCCS Cray Local Utilities (9k)                              60%
+-------------------------------------------------------------------------+
anlyzT      Remotely analyze a tape which resides in the NCCS tape
            library.  The NCCS currently supports either cartridge tapes
            or 9-track reel-to-reel style tapes.  This utility operates
            via the Cray Station Software (CSS) and can be invoked only
            from the NCCS Cray.

initT       Remotely initialize (label) a tape which resides in the NCCS
            tape library.  The same restrictions apply to the physical
            style of tape which can be initialized as is true for the
            anlyzT utility.  Tapes may be initialized to the IBM standard
            label (SL) format, ASCII labeled (AL) format or nonlabeled
            (NL) format.  This utility operates via the Cray Station
            Software (CSS) and can be invoked only from the NCCS Cray.

nettape     Remotely retrieve one or more files from a tape which resides
            in the NCCS tape library system (TLS).  The files, once
            copied from tape, are then directed to the requested internet
            node.

zapback     Change the record format of an IBM resident file to a
            different IBM record format.

NCCS Usage Monitoring and Account Information Utilities
-------------------------------------------------------
cusage      Display the amount of CU's that have been used to date on the
            current account (default) or the total CU usage for each user
            in a group.  Available only on the Cray.

divnuse     Display the computer unit (CU) usage of a division to date,
            overall sponsor code and individual usage.  Also, for
            authorized division representatives, provides an option to
            display a detailed breakdown of charges.  Available only on
            the Cray.

dspace      Display the amount of disk space allotted and/or currently
            used up on a user's private disk storage area.  Available
            only on the Cray.

seepct      Display the percentage of storage charges allocated to a
            user's sponsor codes.

sponuse     Display the total CU usage for each of a sponsor's sponsor
            codes.  Displays individual CU utilization sorted by sponsor
            code for all of a sponsor's codes.  This command is for
            authorized NCCS sponsors only.  Available only on the Cray.

+-------------------------------------------------------------------------+
[Help: ?]  [Exit: u]  [PageDown: Space]  [PageUp: b]
```

```
Overview of NCCS Cray Local Utilities (9k)                              81%
+-------------------------------------------------------------------------+
showacct    Display the NCCS sponsor code (account number) of the user
            who is currently logged on.  Available only on the Cray.


Other NCCS Utilities
--------------------
chkpw       Determine when your password will expire on the Cray.  The
            equivalent command on the Convex is pwexp.

dmcounts    A utility which displays the amount of outstanding DMF (silo
            migration and retrieval) requests on the Cray.

netrdir     Perform a recursive directory listing of any UNIX node on the
            internet (including the NCCS UniTree mass data storage
            system) to which a user has access.  The user may specify the
            subdirectory level where the recursive listing should begin
            or default to the $HOME directory level.  If netrdir is
            used to access the NCCS IBM MVS system, a recursive file
            listing will not be performed as the IBM MVS system does not
            maintain a hierarchical file structure.  Only a single level
            data set listing at the depth specified will be given

newf        Determine what files in the current working directory
            (default), or the directory of choice, are time stamped with
            today's date or the date entered as an argument on the
            command line.

prmotd      Display the Message of the Day.  This includes scheduled
            downtimes, announcements of new utilities and general
            information that is of value to the user community.

pwexp       Determine when your password will expire on the Convex.  The
            equivalent command on the Cray is chkpw.

showqs      Display the Cray job queue status.  Available only on the
            Cray.

statinfo    Display the current status of the NCCS systems.


testnode    A script to ping an internet node and return a status code.
            It is primarily intended for use in batch scripts.

dmfstat     A script which tests to see if the DMF daemon is running.
            A running DMF daemon is an indication that /silo file
            disk-to-tape migration and tape-to-disk file recall is
+-------------------------------------------------------------------------+
[Help: ?]  [Exit: u]  [PageDown: Space]  [PageUp: b]
```