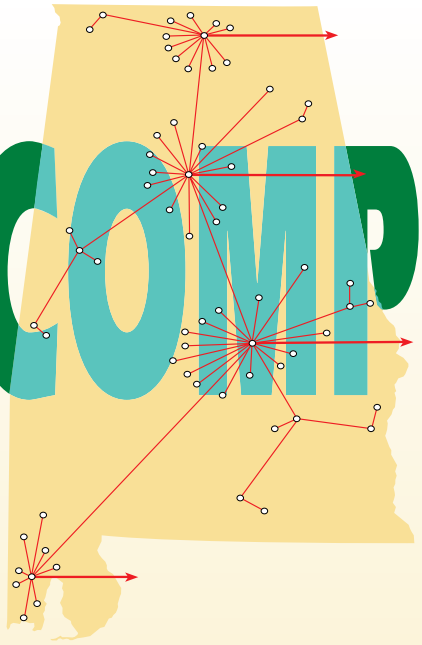
ALABAMA SUPERCOMPUTER CENTER



User Manual

Fifth Edition

Alabama Supercomputer Authority
686 Discovery Drive
Huntsville, Alabama 35806

ASCM.MISC

The Alabama Supercomputer Center

User Manual

Fifth Edition

Alabama Supercomputer Center
686 Discovery Drive
Huntsville, Alabama
35806

Publication	Date	Description
1st Edition	February 1988	Original printing.
Revision A	September 1988	Minor typographical and editorial corrections.
2nd Edition	June 1990	Updates and modifications of 1st Edition.
3rd Edition	June 1993	Complete rewrite.
Revision A	October 1993	New procedures and locations.
4th Edition	January 1994	Update for Cray C90 and editorial corrections.
5th Edition	March 1997	Updates, modifications, and new format.

Overview

This manual is provided for the users of the Alabama Supercomputer Center (ASC) as the primary reference for use of the Alabama Research and Education Network (AREN) and the Alabama Supercomputer Center facilities. The manual covers the physical structure of the network, available software and hardware, access methods, and user support.

Suggestions for additions or corrections to this manual should be directed to an ASC applications analyst or to:

Operations - User Manual
Alabama Supercomputer Center
686 Discovery Drive
Huntsville, AL 35806

This manual is supplemented by a set of policies which cover various aspects of the Alabama Research and Education Network. AREN policies are available from ASC applications analysts and are available online using **asninfo** on the CRAY.

Facilities and Services

Account Administration

Accessing the Network

Obtaining Assistance

Processing on the CRAY C90

Documentation

ASC Internet Access

Figures

The Alabama Supercomputer Center

The Alabama Supercomputer Center (ASC) is established to provide state-of-the-art high performance computing capabilities for researchers and educators at universities and colleges; for K-12 teachers and students in the State of Alabama; and to enhance industrial development for the State. ASC develops and maintains the Alabama Research and Education Network, a statewide network that connects clients to the Internet and to ASC's computing resources via high speed telecommunications links.

The Alabama Supercomputer Center (ASC) provides high performance computing resources to state academic users, state government agencies, national industrial users, and federal government agencies. Networking services are also provided, including access to ASC high performance computing resources, Internet access, World Wide Web services, and training. ASC resources, including a Cray Research C90 supercomputer, are accessed through the Alabama Research and Education Network (AREN), a statewide high speed network installed and maintained by ASC.

ASC is operated by the Alabama Supercomputer Authority (ASA). ASA is a public state nonprofit corporation that develops, maintains, and operates the Alabama Supercomputer Center and the Alabama Research and Education Network. Technical services are provided through professional services and facilities management contractor Nichols Research Corporation (NRC).

The basic configuration of the Alabama Supercomputer Center is shown in Figure 1. The centerpiece of ASC is a Cray Research C90 parallel vector processing supercomputer with a tightly coupled I/O subsystem. It has two CPUs, 64 million words of central memory, and a 32 million word solid state storage device (SSD). The Cray C90 interfaces to the remainder of the network through a fiber distributed data interface (FDDI) ring in the Alabama Supercomputer Center (see Figure 1).

Other high performance computing resources at the ASC available through the network include an SGI 4D/480 server and an IBM RS/6000. The SGI 4D/480 provides 30 GB of NFS mounted scratch disk space for the Cray C90 and is the host of the ASC World Wide Web server. A Sun Sparcstation 5 functions as a mail server for commercial users and the ASC staff.

The growing network comprises nodes connected through routers, leased DS1 (T1) (1.544Mbs), DS0 (56Kb), and ISDN lines.

Except for periods of scheduled maintenance, the network is available 24 hours per day, every day of the year. The center is staffed 24 hours per day, every day except Christmas and New Year's Day.

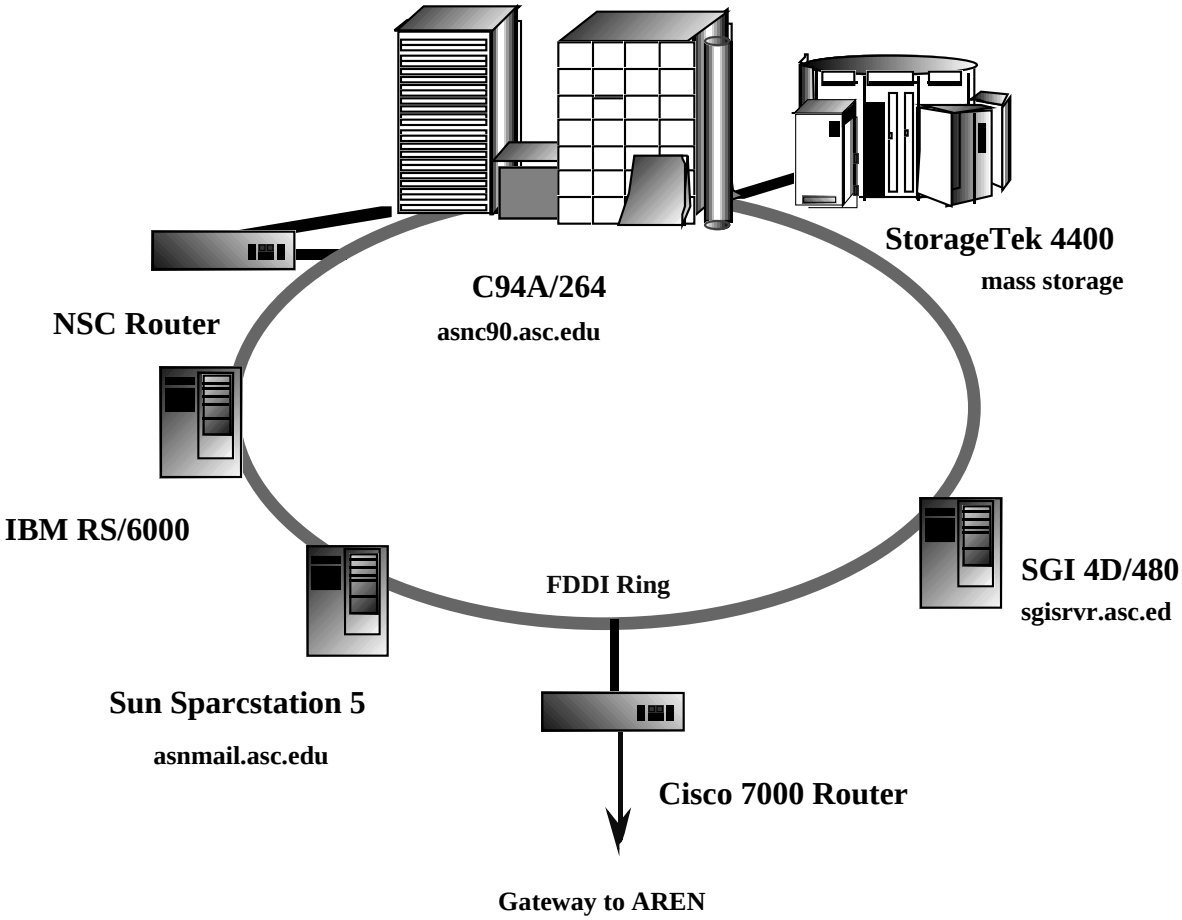


Figure 1. Alabama Supercomputer Center Primary Equipment

Alabama Supercomputer Center User Support

An automated problem management system is operated at the Alabama Supercomputer Center to provide tracking and accountability for problem resolution.

There are five basic elements to the ASC support program:

- *On-Site Applications Analysts*
- *Help Desk Service*

  1-800-338-8320

  Email: helpdesk@asc.edu
- *Problem Management System*
- *Documentation*
- *Training*

Support for ASC users combines central site (ASC) support with campus based applications analysts and is supplemented with training programs. This support allows the user to process productively as rapidly as possible. Ongoing support to overcome problem areas and in mapping high performance computing technology into the researcher's specific area of study is provided.

Applications analysts are available at selected nodes to provide training and support to the local user community. Their function is described more fully in the Technical Support section.

Specialized local training is available to enhance user effectiveness.

A manned Help Desk is available 24 hours a day to assist with problem solving and to answer user questions about the status of the ASC systems and AREN. The Help Desk is accessible through a toll-free number, 1-800-338-8320, as well as through electronic mail (helpdesk@asc.edu) on the network.

An automated problem management system is operated at the central site to provide tracking and accountability for problem resolution.

A full set of documentation for ASC resources is available at selected nodes and the central site. Users may also purchase vendor manuals if desired. Electronic versions of ASC resource documentation is available online.

CRAY C90 High Performance Vector Processor

*Each Cray C90 CPU can produce results at a rate of **960 Million Floating Point Operations per second** (MFLOPS). A rate of 1905 MFLOPS can be achieved by the use of both CPUs in parallel.*

The ASC Cray C90 (see Figure 2) consists of two processing units (CPUs), a 64 million word main memory, an I/O subsystem with 50 billion bytes of disk storage, and a 32 million word solid state disk (SSD).

CPU

The two CPUs are identical and can process in scalar or vector mode.

Clock period	4.2 nanoseconds
Instruction stack/buffers	256words
Functional units	15
Registers	8 x 128 vector (64 bit) 8 scalar (64 bit) 8 address (32 bit)
Maximum result rate	960 MFLOPS

Memory

Word length	64 bits
Cycle time	68.0 nanoseconds
Memory banks	32
Address space	64 million words
Max bandwidth/channel	150 million words / second

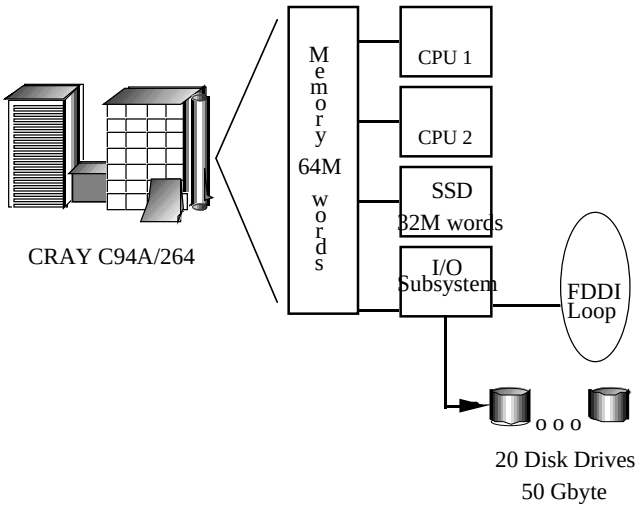


Figure 2. Cray C90 Architecture

Functional Units

Functional units receive operands from the registers, perform the operation, and send the results to a register. Each unit performs its operation in a fixed amount of time, called the functional unit time. Functional unit time is the time between the arrival of the input operands and the completion of the operation, measured in 8.5 nanosecond clock periods. There are 15 functional units:

3 integer add (1 vector)
1 integer multiply
3 shift (1 vector)
3 logical (2 vector)
1 floating point add (scalar/vector)
1 floating point multiply
1 reciprocal approximation (scalar/vector)
2 population count

SSD

The Solid State Storage Device (SSD) has a capacity of 32 million words, or 256 megabytes. The transfer speed is 1,000 megabytes per second to/from the memory. Access time is 40 microseconds. The SSD is used as a high speed disk cache.

Vector Registers

The Cray C90 has eight vector registers, V0-V7. Each vector register is 64 elements of 64 bits. It takes seven clock periods to load the first word of a vector register. The remaining words are loaded one per clock period thereafter if there are no memory bank conflicts. Vector registers participate in four types of instructions:

- operation on two vector operands giving a vector result
- operation on one vector and one scalar operand giving a vector result
- load a vector register from memory
- store a vector register to memory.

Software

Several software packages and libraries are available on ASC systems, including the IMSL libraries and application packages such as UNICHEM, ABAQUS, ANSYS, MSC/NASTRAN, GAUSSIAN 94, and others.

ABAQUS
Design Analysis
Cray
General purpose finite element code for analysis of structures subjected to static or time varying loads.

AMBER
Biotech/Computational Chemistry
SGI
A suite of programs for performing a variety of molecular mechanics-based simulations.

AMPAC
Biotech/Computational Chemistry
Cray
A suite of semi-empirical quantum mechanics codes. Able to do single point energies, geometry optimizations, transition state searches, and other calculations analogous to Gaussian.

AMSOL
Biotech/Computational Chemistry
Cray
Semi-empirical quantum mechanics code which can do calculations which incorporate solvation effects into wavefunction.

ANSYS w/ Expanded Waveform
Design Analysis
Cray
General purpose, finite element program for engineering analysis and includes pre-processing, solution, and post-processing.

ARC2D
Computational Fluid Dynamics
Cray
Solves the two-dimensional Euler and Navier-Stokes equations using an implicit approximate factorization finite difference scheme.

ARC3D

Computational Fluid Dynamics
Cray
Solves the three-dimensional Euler and Navier-Stokes equations using an implicit approximate factorization finite difference scheme.

BCSLIB

General Math and Statistics
Cray
Standard BCS mathematical library.

BCSLIB-EXT

General Math and Statistics
Cray
Extended BCS mathematical library. Highly optimized for the Cray.

BLAST

Biotech/Computational Chemistry
Cray
Heuristic search algorithm employed by the programs blastp, blastn, blastx, and tblastn; these programs ascribe significance to their findings using the statistical methods of Karlin and Altschul (1990). The BLAST programs were tailored for sequence similarity searching, for example, to identify homologs, and are not generally useful for motif-style searching.

CADPAC

Biotech/Computational Chemistry
Cray
Suite of programs which offers calculations of polarizabilities, dipole moment derivatives, and force constants at the MP2 level using fully analytic approaches.

CAL- Cray Assembler

Compilers/Libraries
Cray
Standard Cray Assembler.

CF77- Cray FORTRAN 77

Compilers/Libraries
Cray
Standard Cray FORTRAN 77 autotasking compiling system.

CFT- Cray FORTRAN

Compilers/Libraries
Cray
Standard Cray FORTRAN compiler.

CFT77- Cray FORTRAN 77

Compilers/Libraries
Cray
Standard Cray FORTRAN 77 compiler.

F90 - Cray FORTRAN 90 compiler

Compilers/Libraries
Cray
Cray FORTRAN 90 autotasking compiling system.

FIDAP 7.61

Computational Fluid Dynamics
Cray
Analyzes fluid flow including heat and mass transfer. It includes comprehensive pre- and post-processing capabilities and a flexible and powerful solver.

FMP- Autotasking Preprocessor

Compilers/Libraries
Cray
Standard Cray autotasking preprocessor.

GAMESS

Biotech/Computational Chemistry
Cray
Quantum chemistry code.

GAUSSIAN 92

Biotech/Computational Chemistry

Cray

Ab initio quantum mechanics codes. Capable of handling RHF, ROHF and UHF calculations. Contains both CI and Moller-Plesset routines to handle effects of electron correlation. Can return thermochemical values such as zero point energies and electronic data such as vibrational and electronic transition modes.

GAUSSIAN 92 DFT

Biotech/Computational Chemistry

Cray

Density Functional Theory equivalent of Gaussian 92. Full implementation of Beckley and other functionals.

GAUSSIAN 94

Biotech/Computational Chemistry

Cray

1994 revision of Gaussian 92. This implementation includes the use of redundant variables in geometry optimization which decreases the overall number of steps to reach a minimum.

GNUPLOT

Visualization

Cray

Command driven interactive function plotting program. Gnuplot plots any number of functions built up of C operators, C library functions, and intrinsic operations such as exponentiation.

GRAPE2D

Computational Fluid Dynamics

Cray

Generates two-dimensional computational grids about arbitrary configurations. GRAPE2D is especially useful for generating grids about uncambered or cambered airfoils and projectiles of virtually any shape or design.

GRAPE3D

Computational Fluid Dynamics
Cray
Generates three-dimensional computational grids about arbitrary configurations. GRAPE3D employs a fast iterative Poisson equation solver to generate 3-D grids about arbitrary shapes.

IMSL Parallel Express Grp Lib

Compilers/Libraries
Cray
Comprehensive resource of more than 900 FORTRAN mathematical and statistical subroutines for scientists, engineers, and mathematicians.

INS3D-UP

Computational Fluid Dynamics
Cray
Solves the incompressible Navier-Stokes equations in three-dimensional generalized coordinates for both steady state and time varying flow.

LS-DYNA3D

Design Analysis
Cray
Explicit finite element program used to analyze the nonlinear dynamic response of inelastic structures with fully automated contact capabilities.

ME/NASTRAN

Design Analysis
Cray
Similar to MSC/NASTRAN but provided by Macro Engineering. Source code (FORTRAN) is available for users wishing to write their own elements.

MM3

Biotech/Computational Chemistry
Cray
Molecular mechanics chemistry code.

MOPAC
Biotech/Computational Chemistry
Cray
General purpose, semi-empirical molecular orbital program for the study of chemical reactions involving molecules, ions, and linear polymers.

MSC/ARIES
Design Analysis
SGI
Provides interactive graphics modeling, results processing, and close interfacing to the software packages MSC/NASTRAN, MSC/DYTRAN, and MSC/EMAS.

MSC/DYTRAN
Design Analysis
Cray
General purpose, three-dimensional computer program for simulating the dynamic response of solid components, structures, and fluids.

NCAR (UNIX)
Visualization
Cray
Two-dimensional visualization package.

NCSA HDF
Visualization
Cray
Hierarchical data format library and utilities.

NSPCG
General Math and Statistics
Cray
Library of routines for solving linear systems of equations using non-symmetric preconditioned conjugate gradient methods.

PV WAVE CL

Visualization
Silicon Graphics
Software environment for solving problems requiring the application of graphics, mathematics, numerics, and statistics to data and equations.

SCC - Cray C Compiler

Compilers/Libraries
Cray
Standard Cray C Compiler.

SCILIB

Compilers/Libraries
Cray
General purpose scientific library.

SINDA/G

Design Analysis
Cray
Software system which has capabilities that make it suited for solving lumped parameter representations of physical problems governed by diffusion-type equations.

SLAM II

Compilers/Libraries
Silicon Graphics
General purpose simulation development language.

SLATEC

General Math and Statistics
Cray
Math library that supports such calculations as: arithmetic error analysis; elementary and special functions; elementary vector operations; solutions of systems of linear equations; and more.

SPARTAN

BioTech/Computational Chemistry
Cray
Computationl chemistry package that embraces both semi-empirical and ab initio methodologies as well as molecular mechanics.

Software packages are added to the system from time to time. For a current list, contact a local site analyst or the ASC Help Desk.

TEXAS93
Biotech/Computational Chemistry
Cray
Semi-empirical chemistry code.

TGIF
Visualization
Cray
Converts DISSPLA metafiles to/from Transportable Graphics Interface Files.

TRASYS
Design Analysis
Cray
Computer software system with generalized capability to solve the radiation related aspects of thermal analysis problems.

UNICHEM
Biotech/Computational Chemistry
Silicon Graphics
Graphical chemistry package for pre- and post-processing.

X-PLOR
Crystallography
Cray
Supports exploration of conformational space of macromolecules confined to regions allowed by experimental data and error estimates. X-PLOR can be used in the refinement of both x-ray crystal and NMR solution structures. It also supports structured programming in macromolecular simulation.

XDATASLICE
Visualization
Silicon Graphics
Two-dimensional and three-dimensional data and image display and manipulation program for X-Windows.

XIMAGE
Visualization
Silicon Graphics
Two-dimensional image display and manipulation program for X-Windows.

StorageTek 4400

With the advanced computational power available at Alabama Supercomputer Center, users need storage capabilities which provide adequate access speed and allow storage across multiple computing platforms. StorageTek's 4400 Automated Cartridge System (ACS) meets these challenges with an efficient implementation of systems managed storage. The ACS 4400 is a fully automated information storage system that mounts and unmounts 1/2-inch 18-track cartridges. Quick and consistent data access is achieved by the PowderHorn, a high performance free standing robot. PowderHorn uses a balanced "H-arm" with dual "hand" mechanisms on a rotating turntable to achieve up to 350 cartridge exchanges per hour. Mounted on each hand is a solid-state camera which scans the bar-coded cartridge volume label and mounts the cartridge in the appropriate drive.

ASC Cray C90 users access the StorageTek (STK) through the use of Cray's Data Migration Facility (DMF). DMF ensures the availability of file system space by moving selected files from online disk to the STK Automated Cartridge System. This change in residence is transparent to users because the files remain cataloged in their original directories. When a file is migrated to the offline storage system, its i-node (information node/directory entry) remains online, but the data is moved offline. Files can be migrated either by explicit user request or by the automated space management components of the DMF. These components automatically monitor and adjust the contents of the file system based on criteria specified by the system administrator. The StorageTek ACS is an offline storage facility and cannot be considered a *file backup* option. The deletion of a file results in the removal of the i-node and, therefore, the permanent removal of the migrated data.

The ASC C90 file system ***/dmf_stage*** is provided for user files and is under the automated space-management facility, DMF. When an ASC C90 new user account is created, a directory on the ***/dmf_stage*** file system is created for that user. Any user files placed in the user's ***/dmf_stage*** directory will be automatically migrated to tape by DMF. The

/dmf_stage file system hierarchy consists of the file system ***/dmf_stage*** and subdirectories indicating a user's group or university association. For example, a user with username uabxxx01 would have the following DMF staging directory: ***/dmf_stage/uab/uabxxx01*** where uabxxx01 is the name of the user's DMF staging directory. The user can also explicitly migrate files from ***/dev/dsk/home*** to ***/dev/dsk/dmf_stage*** by issuing the command:

dmput [options] ***filelist*** - marks a file for migration to offline storage
dmget ***filename*** - retrieves an offline file from storage

Migrated files are not charged against a user's disk quota. A user can immediately release disk space for usage by including the **'-r'** option to the ***dmput*** command. A user must have enough disk space remaining to retrieve a file from storage and still remain under the user's disk quota if that user is retrieving a file from a directory other than the user's ***/dmf_stage*** directory. Consult the man pages for ***dmput*** and ***dmget*** for examples and further explanation on these commands.

This offline status of a migrated file is indicated by a letter **'m'** in the leftmost column of a long format directory listing. The long format directory listing is obtained by issuing the '**ls -l**' command.

For example:

```
% ls -l
mrwxr-----   1 uabxxx01 uabgroup       3330 May 18  1995 datfile1.dat
mrwxr-----   1 uabxxx01 uabgroup       1863 May 19  1995 datfile2.dat
-rwxr-----   . 1 uabxxx01 uabgroup   1702976 Aug 19  1995 a.out
```

The **'m'** in the left column of the listing indicates that the first two files are migrated to offline storage while the third file, **a.out**, is not migrated and resides on hard disk space.

Technical Support

Three kinds of technical support are provided by the Alabama Supercomputer Center for AREN users: local campus applications analysts, toll-free "hotline", and consultation support from ASC central site analysts.

Local applications analysts provide leadership in the growth of high performance computing technology at their node in their area of responsibility. They can assist the universities in the development of courses, collaborate in research efforts, and coordinate local support requirements.

The focal point for technical support is the local applications analyst. These analysts work as a team to provide the following services to both educational and industrial users across the state:

General Support: Analysts provide assistance in establishing user accounts, job control language, program compilation and execution, and other user support as needed.

User Training: Analysts provide introductory lectures, classroom training, and one-on-one instruction.

Application Program Support: Analysts provide support for conversion of programs, optimization and vectorization of code, use of application packages, and resource management.

Outreach Support: Analysts assist in promotion of ASC resources to potential academic and industrial users, through formal technical presentations, demonstrations, benchmarking of codes, and technical consultation.

Contact Information: If you do not know how to reach the analyst assigned to your campus or institution, please call the Help Desk at (800-338-8320. This information is also available on the Cray via the ***asninfo*** command.

Obtaining an ASC Account

Academic Block: *Available computing resources are allocated to the universities in accordance with policies set by the Board of Directors of the Alabama Supercomputer Authority. The allocation for each university is established semi-annually by the CEO of the Alabama Supercomputer Authority.*

Commercial Block: *Available computing resources are allocated for commercial and non-Alabama academic users by the Alabama Supercomputer Authority.*

The Alabama Supercomputer Authority has designated two categories of accounts:

Academic Accounts are requested by the individual universities according to university developed policy. The local applications analyst can explain the procedure for obtaining an academic account at a particular university.

Commercial Accounts include commercial and non-Alabama university research activities. All commercial accounts are coordinated through the Huntsville ASC office. The ***Agreement for Computing Services*** form must be completed. The local applications analyst can provide assistance in coordinating this agreement.

Charges are not assessed for the use of the Cray for unsponsored research by Alabama's public educational institutions. However, sponsored use by universities is nominally charged at 80% of the prevailing commercial rate. **Some software packages, such as MSC/NASTRAN and ANSYS, have a usage royalty charge which may apply to some accounts**.

Each user must have a separate account on the Cray. To request an account, the user should submit an ***AREN Account Activity Request Form*** to the local applications analyst. These forms are available from the analyst or the campus designated account authority, an individual who administers the institution's allocation of computing time. The completed form is forwarded to the ASC account administrator for processing. Once the account is established, the user is notified by the applications analyst or the account authority.

Tracking Resource Consumption

Every account is given a dollar amount limit for Cray usage. It is the user's responsibility to control resources and to monitor the amount of the allocation which has been used.

A user may review the charges accrued to an account at any time by issuing the **usage** command. The account limit, previous dollar usage, and the account month-to-date totals are displayed.

Resources are assigned and controlled by the ASC accounting system which tracks and logs usage of various system resources. Usage totals are updated as follows:

Resource	**Frequency**
CPU usage	end of job
Memory Residency	end of job
I / O Time	end of job
Connect Time	daily
Pages/Plots	daily
Total Dollars Used	end of job & daily
Previous Months $ Used	monthly

The charge for service is based on a "recorded hour" which is calculated by the accounting system from CPU hours, average memory residency, and input/output time (disk accesses and data I/O). The formula used is:

CPU Hour
+1/10 memory Mwords*Hour
+1/100 I/O Hours
+1/10 Connect Hours
=Recorded Hour

Service Charges

Service charges for the Cray C90, measured in Recorded Hours, are based on CPU time, interactive connect time, memory usage, and disk access. The monetary charge per Recorded Hour is stated in the AREN Policy, ***ARENP04: Processing Charges.*** NQS queue structure and limits are shown in Figure 3. Interactive limits are shown in Figure 4.

Express processing is charged at 1.4 times the base rate. Interactive and prime processing are charged at the base rate, and off-hours processing is charged at 0.6 times the base rate.

Queue	Pri	Run	Nice	CPU/req	Mem/req	File/req	File/proc
small_a	55	2	2	3600sec	6mw	1gb	1gb
small_b	50	3	4	14400sec	6mw	3gb	3gb
small_c	45	4	6	108000sec	6mw	4gb	4gb
medium_a	50	2	4	3600sec	8mw	2gb	2gb
medium_b	45	3	6	14400sec	8mw	4gb	4gb
medium_c	40	2	8	108000sec	8mw	5gb	5gb
large_a	45	1	6	3600sec	12mw	4gb	4gb
large_b	40	1	8	14400sec	12mw	5gb	5gb
express_a	60	2	2	600sec	12mw	4gb	4gb
express_b	60	1	2	600sec	20mw	4gb	4gb
large_c	35	1	8	108000sec	12mw	6gb	6gb
huge_b	40	1	8	14400sec	24mw	15gb	15gb
huge_c	35	1	8	108000sec	24mw	16gb	16gb
sysadm	60	1	2	7200sec	24mw	4gb	4gb

Figure 3. NQS Queue Structure and Limits

	Pri	Nice	CPU/req	Mem/req	File/req	File/proc
interactive	0	0	300sec	4mw	128mb	128mb

Figure 4. Interactive Limits

ASC Accounting System and Procedures

The computer accounting system for the Alabama Cray C90 supercomputer uses standard UNICOS accounting data with several enhancements. The enhancements provide the following functions:

•Dollar limits on accounts
•Current and total resource usage display
•Proprietary program surcharge handling
•Month-end billing information

An account will automatically be prohibited from further processing when the account limit is exceeded. When this occurs, a user will not be allowed to begin an interactive session or batch job and will receive a message stating that the dollar limit has been exceeded for the session.

A user applying for an account will be asked to specify which surcharged programs will be used, and authorization for those programs will be added to the account. When an account uses one of the surcharged programs, the accounting system looks up the associated surcharge rate in an internal table and adds it to the dollar amount charged to the account.

Month-end accounting report files are reviewed by the Alabama Supercomputer Authority. Actual dollar amounts from these reports are used by the Authority's accounting system to compute the amount billed to commercial users and users with software royalty charges. Unsponsored academic research accounts are not billed directly but are limited by the account dollar limit mechanism to ensure an equitable distribution of the high performance computing resources.

The accounting system and related procedures handle all the administrative requirements for accounting. Procedures in place for adding new accounts, deleting unused or expired accounts, and modifying existing accounts include actions to handle such situations as increasing the dollar limit, adjusting for charge backs, changing rate schedules and priority factors, changing expiration dates, and handling security in case of forgotten passwords. The applications analyst or account authority for an account can help solve problems relating to such issues.

The *usage* Command

*Accumulated resource usage on the Cray can be displayed with the **usage** command. Options allow users to show actual CPU utilization or account dollar balance.*

The ***usage*** command presents session cost information to the user. It can provide resource usage detail or current month-to-date and account limit information. It is available during each interactive session and provides account information for the session.

Options:

usage [-c] [-s]

-c show actual CPU utilization

-s show account dollar balance for current quarter and year to date

Effective Resource Usage

Offline editing of programs (on a local machine or a personal computer) considerably reduces connect time on the Cray and is highly recommended.

There are several important steps a user can take to reduce computer usage costs and to use the supercomputer efficiently. These include adjusting user priority and type of access, monitoring offline resources, and using good programming techniques.

Users are urged to use the least expensive priority that is effective for their requirements. In addition, careful consideration should be given to whether a job is run in interactive or batch mode.

Online storage should be periodically checked for obsolete or seldom used files. Cray home directories are limited to 10 MB. Printing can be reduced by reviewing the output file using the terminal screen and by discarding runs that are incomplete or, for some other reason, do not need to be printed.

If possible, users should run small test jobs before running larger production jobs; this will minimize debugging costs. The usual care should be given to ensure that input data is accurate and that careful programming techniques are used.

Network Access

Users should contact their applications analyst if assistance is needed to dial into the network.

Many access paths are available to ASC. Dial-up lines are offered in Montgomery, Mobile, and Huntsville. Most campus networks also have dial-up lines available. The dial-up line numbers are available from the campus applications analyst or campus network administrator.

Access from the campuses is usually through the campus network; any user with access to an Internet host can access ASC and AREN.

Network Overview

Access to the Cray C90 is available through the Alabama Research and Education Network.The Alabama Supercomputer Center is connected to the AREN wide area network which extends throughout the state.

The Alabama Research and Education Network not only connects each node in the network directly to the Cray but also interconnects all the nodes to allow resource-sharing and communications among them. To accomplish this, the supercomputer is connected to a 100 Mb/sec FDDI ring, which in turn is connected to a 10 Mb/sec local area ethernet. This ethernet is connected to the AREN wide area network which extends throughout the state through the use of leased DS1, DS0, and ISDN lines.

The network uses the TCP/IP (Transmission Control Protocol/Internet Protocol) protocol suite. TCP/IP provides interactive access, file transfer, and electronic mail service not only to the Cray, but also among all nodes on the network. It also provides a gateway which ties AREN into the global Internet network.

General Network Capabilities

AREN users can communicate directly with the Cray, as well as any other machine on the network which supports TCP/IP.

Users have the following capabilities over the network:

telnet

The **telnet** command allows users on any computer connected to the Alabama Research and Education Network to interact with computers at the Alabama Supercomputer Center as if they were directly connected to them. This is the primary method for interactive access to the Cray. For security reasons, only nodes with Cray accounts have telnet access to the supercomputer.

ftp

The **ftp** command allows users to transfer files between machines.

mail

The **mail** command allows users on any machine connected to the AREN to send electronic mail to any other user on the Internet that has been issued an account/e-mail address.

lpr

The **lpr** command allows users logged into a UNIX based machine such as the Cray or the Sun to print files at a remote site. Contact the campus applications analyst to add remote printers or to get a list of available remote printers.

WWW Access

ASC provides users with access to the World Wide Web (WWW)**.** ASC's web server is located at url: **http://www.asc.edu**

Connecting to the ASC Computing Resources

*In order to connect to a machine on the AREN the user **MUST** have a **user_id/account and password** on that machine.*

When a user dials up to a server, the user must enter their user_id and the appropriate password. See the campus application analyst if any problems occur. After users are logged in or connected to the server, a **local>** prompt is displayed**.** To connect to a host, enter:

local> **telnet hostname**

where *hostname* is the name of the login system.

Examples:

local>**telnet asnc90.asc.edu**

or

local>**telnet asnmail.asc.edu**

Once the user has accessed the network, the ASC systems listed in Figure 5 can be accessed with the TCP/IP host names.

Host Name	Host Operating System	Location
asnc90.asc.edu	UNICOS	ASC
asnmail.asc.edu	UNIX	ASC

Figure 5. Network Hosts

Access from the Nodes

Any processor connected to the AREN or to a connected network can access the ASC facilities if it supports the TCP/IP protocol.

Note: The IBM 3270 systems at TSU do not allow use of the **telnet** command unless the user is using or emulating an IBM 3270 terminal.

When using commands involving a host name, such as **telnet** or **ftp**, users should always specify the full name, e.g. **asnc90.asc.edu.** This should also be done when connecting from a terminal server.

Communication in Dial-up Mode

Dial-up to the host at AREN nodes requires a host account. Dial-up to the terminal servers allows the user to connect to any system recognized by AREN. The AREN servers allow connection to any ASC system and most of the connected campus networks. The AREN servers require the user to enter an identifier and a password before connecting. See the campus Applications Analyst for this information.

Most sites support up to 14,400 baud duplex modem connections, 8 bit, 1 stop bit, and none/no parity.

WWW Access

The Alabama Supercomputer Center provides users with access to the World Wide Web (WWW).

The URL for the ASC web server is:

http://www.asc.edu

The ASC homepage provides links to other sites on the WWW.

Connecting Hardware to the Network

*Permission must be received **before** any hardware can be directly connected to the AREN.*

Universities are permitted to connect certain mainframes and other networks to the AREN.

There are several guidelines to keep in mind:

- Network connections must be made through a router.
- Internet names and addresses must be coordinated with the Alabama Supercomputer Center.
- Please coordinate the connection request with the campus applications analyst or with the ASC service manager.

Applications Analysts

There are four major ways to get help when using the Alabama Research and Education Network and the Alabama Supercomputer Center facilities:

Assigned applications analysts

The "hot-line" help desk at the Alabama Supercomputer Center

On-line documentation and help commands

Use of the MAIL facilities

Primary assistance for any kind of problem is the Applications Analyst who has been assigned to help users at each location.

The applications analyst is available to assist users getting started and to consult on problems which may arise concerning ASC's supercomuting systems, AREN communications network, and user applictions.

The analyst serves as the focal point for acquiring help for the user from the ASC support staff, other applications analysts, or technical specialists.

The ASC Help Desk

The Help Desk is staffed by operators who will assist in solving the problem, or forward it to one or more application analysts. Operators are available 24 hours a day.

A toll-free number **(1-800-338-8320)** is provided for ASC users.

Online Help

Various online help facilities are available. UNICOS or UNIX information can be obtained with the **man** command:

man *title*

or

man -k <*keyword*>

The **man** command locates and prints the entry named *title*. The title is entered in lowercase. The following example reproduces the description of **man** on the standard output:

man man

To get information on the f90 command, type:

man f90

To get general information about ASC, including latest software, policies, and procedures, type:

asninfo

asninfo is a menu-driven program which is self-explanatory.

Using Help Desk Electronic Mail Facilities

Procedures have been established to MAIL a user's problem to the HELPDESK. directly from the prompt.

To send a problem or a question to the Helpdesk, enter **helpdesk** and follow the instructions.

Sample request for help on the Cray:

asnc90$ helpdesk

Welcome to the ASC HelpDesk Facility

Please enter your question or problem.
To end input and send this text as mail, enter a
<Control>D
Your message must contain at least one Carriage Return.

....Description of help needed goes here.....

<cntl>D

Please wait to determine active helpdesk host
Mail delivered to helpdesk@asnmail.asc.edu.

Thank you for using the HelpDesk Facility.

asnc90$

ASC Cray C90 Operating System

The ASC Cray C90 runs under the UNICOS operating system, which is the Cray Research, Inc. version of UNIX.

The interactive UNIX operating system was developed by AT&T and has been extended and enhanced at the University of California at Berkeley. UNICOS is based on the AT&T UNIX System V with some enhancements from Berkeley UNIX (bsd 4.3). As such, it is very much like UNIX operating systems on a wide variety of other computers.

UNICOS provides the standard UNIX commands, libraries, and features, such as user shells, pipes, tees, and filters. Also included are text editors (ed, ex, and vi), communications programs (telnet, ftp, and mail), compilers (C and FORTRAN), and the Network Queuing System (NQS) - a Cray enhancement of UNIX to provide "batch" operation. NQS allows users to create a file of commands for the computer to execute at a later time rather than simply typing in the commands one by one from a terminal. Effective use of the Cray C90, particularly for large jobs, requires use of the NQS system.

Complete UNICOS documentation is available in the manuals listed in the *Documentation* section of this manual. Many popular UNIX System V guides and textbooks also provide valuable information and are generally applicable to UNICOS.

Most documentation for UNICOS and for the additional programs provided at the Alabama Supercomputer Center can be accessed online via the **man** command. See the *Obtaining Assistance* section of this manual for information about getting help.

Using UNICOS interactively is described in the following sections. The NQS batch system is designed to accept the same commands as for interactive use. You can prepare a file for submittal to the NQS system with one of the Cray text editors and then place it into an appropriate batch queue.

Logging in to the Cray C90

Interactive access to the CRAY C90 is obtained through the **telnet** command.

Example of a Cray C90 login:

```
% telnet asnc90.asc.edu
Trying...
Connected to asnc90.asc.edu.
Escape character is '^]'.


Cray UNICOS (asnc90) (ttyp022)


login: asndsc01
Password:
Last successful login was : Mon AUG 6 14:10:14 from 129.66.32.250

      ********************************************************************************
      *              Welcome to the Alabama Research and Education Network           *
      *                                   UNICOS 8.0.3                               *
      ********************************************************************************


            This is a State of Alabama computer facility managed by
            the Alabama Supercomputer Authority and its contractor
            Nichols Research Corporation.  The system is for use by
            State of Alabama educational institutions and authorized
            commercial accounts.

            In accessing the system, users consent to the monitoring
            of usage for purposes of accounting and the detection of
            unauthorized access.  Please note that access to this facility
            must be specifically authorized by ASA and that unless you are
            so authorized, access and any other use may expose you to
            criminal and/or civil proceedings.

********************************************************************************
```

```
        The batch queue limits have been modified since the upgrade to
        UNICOS 8.0.3. The total batch job limit is four (4) jobs per user.
        Individual queue limits (per queue, per user) are as follows:

                Queue                   Limit
                -----                   -----

                small_a                   2
                small_b                   2
                small_c                   1

                medium_a                  2
                medium_b                  2
                medium_c                  1

                large_a                   2
                large_b                   2
                large_c                   1

                huge_b                    2
                huge_c                    1

                express_a                 2
                express_b                 2

**************************************************************************
You have mail.

asnc90$
```

Logging Off of the C90

To logoff the C90, enter **exit** and a return.

```
asnc90$ exit
Connection closed.
%
```

Login Profiles and Shells

At login, the user's default shell is started, and the shell script is executed. The default shell for accounts is the Bourne shell, and the shell script is ***.profile*** *in the user's root directory.*

The login shell can be changed to the C shell with the **csh** command. This will also change the login shell script to **.login**. When an account is created, the following scripts are copied into the user's directory from the default scripts in ***/usr/skel.*** The user may alter his login script as appropriate.

The Korn shell is also available on the UNICOS system. See **man ksh** for details.

/usr/skel/.profile	login script for Bourne shell, default
/usr/skel/.login	login script for C shell
/usr/skel/.logout	logout script for C shell
/usr/skel/.cshrc	additional script for C shell

Terminal Control

UNICOS supports many different types of interactive terminals connected through **telnet**. The default login script prompts the user for a terminal type. Valid entries include **vt100, vt220,** and many others.

The UNICOS terminal control characters can be listed by entering **stty -a**. The defaults are:

interrupt = CTRL-C

quit = CTRL-

end-of-file = CTRL-D

When using telnet from an IBM 3270 device, use the percent sign in place of the control key.

Issuing the following command will list, screen by screen, the valid terminal names which may be used:

$ ls -C /usr/lib/terminfo/* | more

Dataset Naming Conventions in UNICOS

UNICOS is CASE SENSITIVE! *Uppercase is distinguished from lowercase:* ***prog.for*** *is not the same file as* ***Prog.for****, for example.*

The same rules apply to both file and directory names in UNICOS which is based on UNIX System V.

- Nearly any keyboard character may be used in file and directory names. It is best to avoid the characters that UNICOS uses in other situations. such as **/ \ " ' * ; - ? [ ] ( ) ~ ! $ { }** < and > which all potentially can create confusion.
- Names may be from 1 to 14 characters long. Periods and underlines may be used to substitute for blanks (which are not allowed) to clarify what the names mean. For example, a documentation file might be designated **read.me** or **read_me**.

Directories in a UNIX system are organized in a tree structure. The root directory is designated **/**. Users on the system have their directories and files placed in a branch of the root corresponding to their node. Other important directories, such as ***sys*** and ***bin***, also branch directly from the root. These directories are designated **/home**, **/bin**, and **/sys**, and are illustrated in Figure 6.

Subdirectories are indicated with a / preceeded by the name of their parent directory. If there is a user subdirectory called **asn** in **home**, for example, the full designation for that subdirectory, starting from the root, would be **/home/asn**. The user **asndsc01** might have organized FORTRAN programs into a subdirectory called **fort**, and one of those programs might reside in the file **prog.f** within the **fort** subdirectory. The full path designation for the file **prog.f** thus would be:

/home/asn/asndsc01/fort/prog.f

Any given directory has a parent directory, in which it is a subdirectory. Shorthand for the parent directory is "**..**".

To get to the directory **mail** from **fort** in Figure 7, one uses the "***change directory***" command **cd** in UNICOS:

cd ../mail

The shorthand designation, "**.**", represents the current directory.

To copy all files with extension **.f** from the parent directory to the current directory, use the following command:

cp ../*.f .

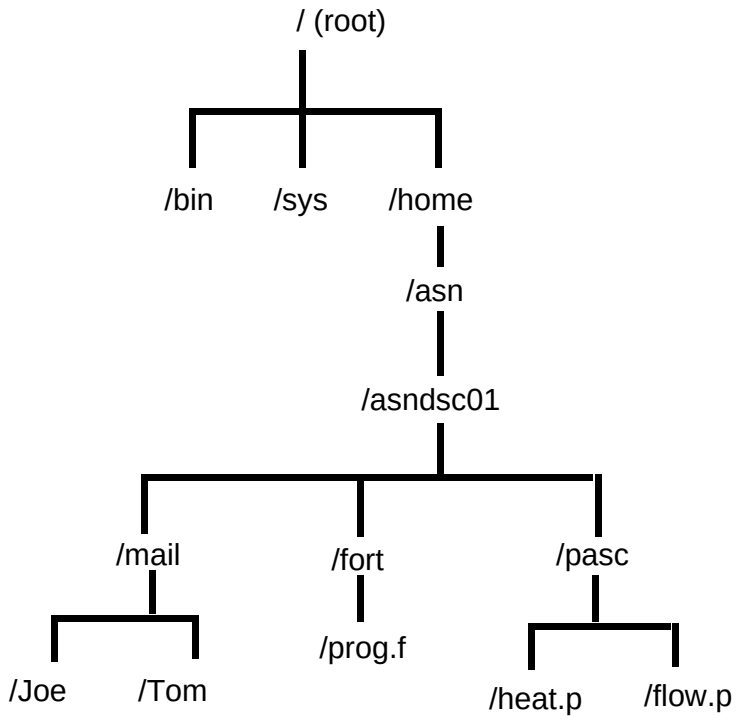


Figure 6. Sample UNIX Tree Structure

ASC UNICOS File Organization

files on **/home**, **/dmf-stage**, and the **/tmp** file systems. Files on **/home** and **/dmf-stage** are moved between online and offline storage by Cray's Data Migration Facility (DMF) and the Storage Tek tape system. Each user has a symbolic link in their **$HOME** directory to their own directory on **/dmf-stage**.

The files on the ASC C90 are organized as diagrammed in Figure 7. Users may locate

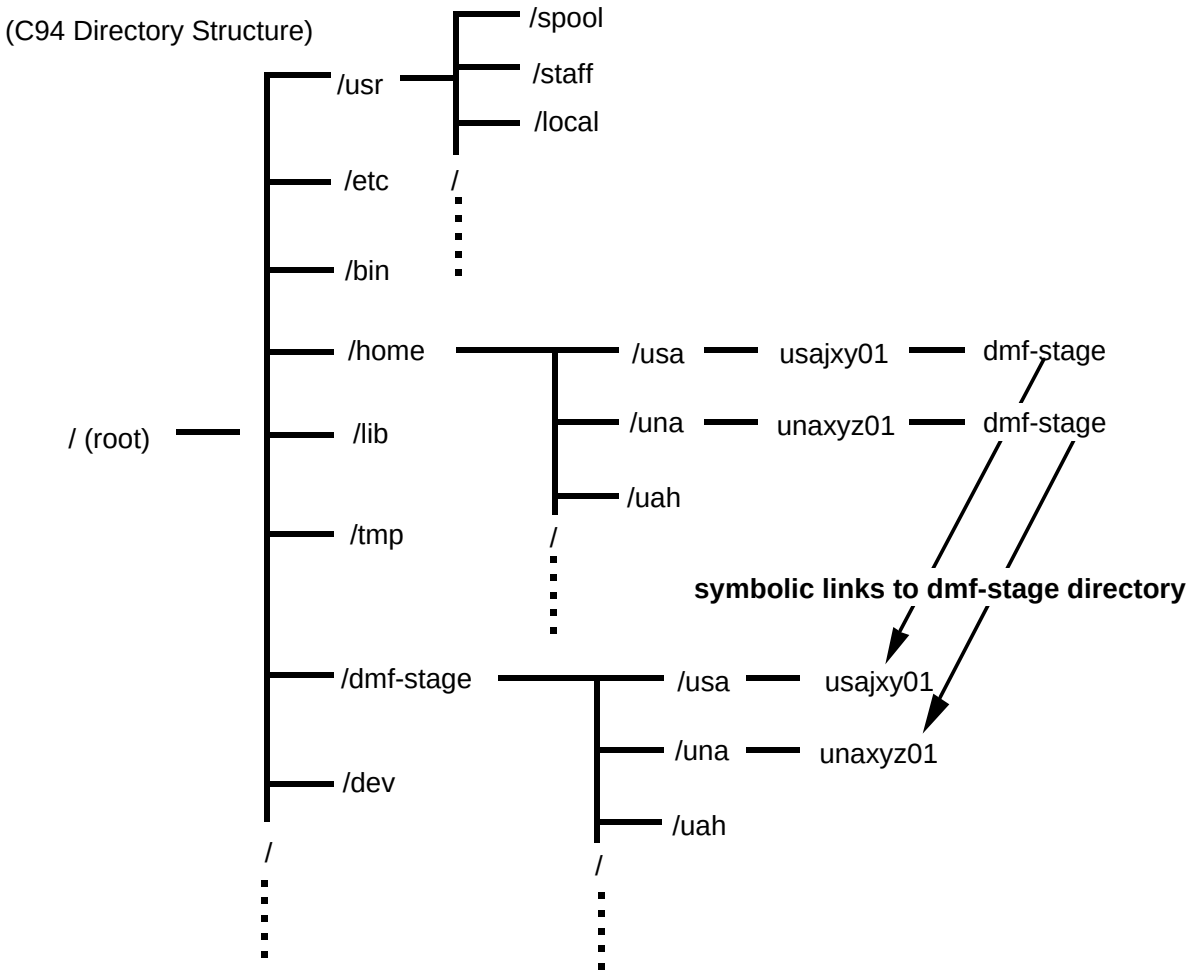


Figure 7. UNICOS File Organization

Manipulating Files and Directories

UNIX/UNICOS commands are used to create, move, copy, or delete files and directories. Each command must be in lower case as shown.

cd

cd without arguments puts the user in the user's home directory. With a directory name as an argument, the command moves the user to that directory.

cp

cp makes copies of files in two ways.

cp filea fileb

makes a new copy of **filea** and names it **fileb**.

cp [list of files] directory

puts copies of all the files named into the directory named. Contrast this to the **mv** command which moves or renames a file.

ln

ln creates a link between files.
Example:

The following links the existing file example.c to ex.c

ln example.c ex.c

The following creates symbolic links.

ln -s /usr/include incl

See the online **man** pages for many other ways to use **ln**.

mkdir

mkdir makes a new subdirectory in the current directory. Example:

mkdir fort

makes a subdirectory called **fort**

mv

mv moves or changes the name of a file. Examples:

mv filea fileb

changes the name of **filea** to **fileb.** If the second argument is a directory, the file is moved to that directory.

One can also specify that the file have a new name in the new directory:

mv filea direc/fileb

would move **filea** to directory **direc** and give it the name **fileb** in that directory.

Contrast this to the **cp** command which copies files.

pwd

pwd returns the name of the current working directory. It simply tells you the current directory.

rm

rm removes each file in a list from a directory. Option **-i** causes **rm** to inquire whether each file should be removed or not. Option **-r** causes rm to delete a directory along with any files or directories in it.

rmdir

rmdir removes an empty directory from the current directory. Example:

rmdir fort

removes the subdirectory named **fort** (if it is empty of files).

To remove a directory and all files in that directory, either remove the files first and then remove the directory or use the **rm -r** option described above.

Frequently Used UNICOS Commands

Since UNICOS is, for the most part, a standard UNIX System V operating system, the usual set of commands is in effect.

cat

cat concatenates and prints the files given as arguments. The output goes to the standard output, which is usually the screen.

cat filea

would print **filea** on the screen.

If no file is given, input is taken from the keyboard. A CTRL-D terminates keyboard input.

Often output is redirected with the operator >. Example:

cat filea fileb > filec

concatenates **filea** and **fileb** and places the result in **filec**.

chmod

chmod changes the file permission status of a file. Permissions may be granted to read, write, or execute the file, and the permission may be given to the user, the user's group, or to the world. When one uses the **ls -l** command these permissions are listed at the left as a series of r's, w's, or x's, with - indicating that permission is not granted.

For example, -rwxrwxrwx indicates read, write, and execute permission is granted to all three groups; -r--r--r-- grants only read permission to each group. **chmod** changes the status of these permissions. The form is the following:

chmod [ugo] [+ -] [rwx] ***files***

u,g, or **o** stand for user, group, or others. The **+** or **-** indicate whether the permission is to be given or denied. And the **r, w** or **x** indicate whether read, write, or execute permission is to be given.

Using ***chmod*** *allows the used to change the permissions on directories since access to a file requires that:*

- *the user has access to the directory containing the desired file; and*
- *the user has appropriate access to the file itself.*

Example:

chmod ug+x filea

will give execute permission to the user and the user's group with respect to **filea.** When a new file or directory is created, the default permissions granted are those specified by the user's **umask** (or default permission mask). Each new ASC user has a umask defined in his default login script (.profile for Bourne shell or .login for the C shell). The user can change the **umask** value to suit the user's needs. See the **man** page on **umask** for details (use **man umask**).

Example: to give group read access to a file named **myprog.f** in a subdirectory called **fort** (i.e. give group read permission to the file **/uab/uabxyz01/fort/myprog.f**) use the following commands:

chmod g+x fort
sets execute permission for directory

chmod g+r fort/myprog.f
sets read permission for file

date

date outputs the date and time

echo

echo repeats whatever text is given to it on the standard output. For example,

echo What's up, doc?

will print **What's up, doc?** on the screen (default for standard output). **echo** is useful for inserting messages in shell scripts, described later.

ls

ls lists the files in the current directory or the directory named as an argument. There are many options:

ls -a [directory]
lists all files, including files whose names start with a period

ls -c [directory]
lists files by date of creation.

ls -l [directory]
lists files in long form: links, owner, size, date and time of last change.

ls -p [directory]
subdirectories are indicated by /.

ls -r [directory]
reverses the listing order.

l**s -s [directory]**
gives the sizes of files in blocks.

ls -C [directory]
lists files in columns using full screen width.

ls -R [directory]
recursively lists files in current directory and all subdirectories.

mail

mail, rmail, and mailx invoke an electronic mail system. **mail** and **mailx** can be used to both send and receive mail. **rmail** only sends mail. Example:

mailx jim@asnuab.asc.edu

sends mail to jim at the node asnuab.asc.edu. The user names are arguments to the command. Users are prompted for a subject after the command.

The letter should be entered line by line and finished with a line containing only a period. Alternatively, the redirection operator < may be used to route a letter prepared with an editor to mail.

Example:

mail jim tom < letter

would send the previously prepared letter to users jim and tom.

man

man provides online documentation for all UNIX commands and utilities, making quite detailed descriptions instantly available. The commands or utilities are arguments to man. Example:

man cat

will give a summary of the use of the concatenate command.

man -k keyword

will give information on all commands relevant to the given keyword.

passwd

passwd allows the user to change his password. Prompts are issued in order to supply the old and new passwords.

paste

Merges same lines of files or subsequent lines of a file.

lpr

lpr is used to print files from the Cray to printers at the Alabama Supercomputer Center and certain printers at some of the universities. The basic form is:

lpr -Pprinter filename

See the section on printing for a complete description of this command.

sed

sed copies files (standard input by default) to standard output, edited according to a script of commands. The script is obtained from either the script operand string or a combination of the arguments from the **-e** script and **-f** sfile options.

sort

sort *filename* sorts the lines in the file in an extended alphabetical order, that is, the alphabet is extended to include special symbols and digits. The options for **sort** are the following:

sort -b [file]
ignores initial blanks

sort -d [file]
uses a dictionary order, without special digits or symbols

sort -f [file]
upper and lower case treated as equals

sort -n [file]
numbers are treated in the order of arithmetic value

sort -o filename [file]
puts the output in file **filename**

sort -r [file]
reverse the sort order

tail

tail prints the last part of a file given to it as an argument. The options are listed below:

tail +/-number b c [file]

tail starts printing **number** lines (or blocks or characters, with **b** or **c** added to the string) from the top (+) or bottom (-) of the file. Default is 10.

wc

wc (**w**ord **c**ount) counts lines, words, and characters in files. **wc file** will return counts for all three. The parameters **-l**, **-w**, and **-c** will limit the report to lines, words, and characters, respectively.

Example:

```
$wc filea
$20 100 500
```

```
20 lines, 100 words, 500 characters
```

who

who lists the login names of all users currently on the system, their terminals, and when they logged on.

who am I gives the same information about the user only.

write

write allows communication with other users currently on the system. The user names are arguments to the command. Each message should be terminated with **-o-** ("over") or **-oo-** ("over-and-out") and a CTRL-D to return to normal screen mode.

mesg y or **mesg n** may be set by any user to permit or not permit, respectively, **write** communications to come through.

Regular Expression Pattern Matching

*The **grep** command matches patterns of ASCII characters in files fed to it as parameters. Positions of characters and multiple occurrences of a character may be matched as well as special characters.*

grep

The form of the command is

grep *regular-expression filea fileb filec etc.*

The regular expression should be enclosed in single quotes if it contains one or more blanks. A regular expression contains the usual ASCII characters, but some characters have special meanings, depending on their location within the expression. The characters with special meaning are (. *** [ ]** \ **$ and** ^). The period substitutes for any character in one position in the expression.

Example:

.abc will match with aabc, babc, 7abc, and so on, and **a.bc** matches a1bc, a2bc, azbc, etc.

Multiple occurrences of a character may be matched with an asterisk.

Example:

a*bc will match patterns with zero or more occurrences of **a**
bc abc aaaaaaaaaaaaabc etc.

A backslash before a special character will remove the special meaning from the character, so that it can be matched for itself.

Example:

***** within a regular-expression will match * in that position.

Pattern Scanning and Processing Language

awk

The syntax for the ***awk*** utility is:

awk [-F ERE] [-v assignment] ***program*** **[argument]**

awk [-F ERE] -f ***progfile*** **[-v assignment]** [**argument]**

The utility ***awk*** scans each input file for lines that match any of a set of patterns specified in ***program***. Each pattern in ***program*** may have an associated action that will be performed when a line of a file matches the pattern. The set of patterns may appear either literally as ***program*** or in a file specified by using the **-f** option. This version of ***awk*** provides capabilities that were not available with previous versions. ***awk*** accepts the following options and arguments:

-F ERE

Defines the input field separator to be the Extended Regular Expression ERE. (Currently, only Basic Regular Expressions are supported.)

-v assignment

assignments in the form **x=xvalue y=yvalu**e can be passed to **awk**; **x** and **y** are awk built-in variables. The specified variable assignment occurs prior to executing the awk program, including the actions associated with ***BEGIN*** patterns, if any. Multiple occurrences of the **-v** option can be specified.

-f *progfile*

File that contains the set of pattern-action statements. If multiple instances of this option are specified, the concatenation of

the files specified as ***progfile*** is used as the ***awk*** program.

program

Set of patterns for which ***awk*** scans file. To protect the program string from the shell enclose it in single quotation marks.

argument

Either of the following types of arguments can be specified:

assignment

assignments in the form **x=xvalue y=yvalue** can be passed to ***awk***; **x** and **y** are ***awk*** built-in variables.

file

Input files; if no files exist, the standard input is read. The ***file name -*** specifies the standard input. Each input line is matched against the pattern portion of every pattern-action statement; the associated action is performed for each matched pattern.

Redirection of Input and Output

Standard input and output may be redirected for any process with the use of the symbols < and >.

The symbol < redirects standard input (by default, the keyboard) so that input is taken from the file named after the symbol.

The symbol > redirects standard output (by default, the screen) to the file named after the symbol.

Note: ***Any information currently stored in the file specified is overwritten and lost.***

For example,

cat > quasar

will take input from the keyboard and place it in the file **quasar** until a CTRL-D is issued.

The special symbol >> appends new data to the named file.

The symbol <<***xxx*** redirects input until it encounters a line with only ***xxx***, beginning in column 1.

Pipes, Tees, and Filters

The standard output of a UNIX command can be fed directly into the standard input of another. This is the concept of the **pipe** (symbol |)**.** Tapping into the stream as it proceeds from the standard output of one command to the standard input of the next is the concept of the **tee** (symbol **tee**)**.**

A **filter** is a particular type of command that may stand in the middle positions of a series of commands linked by pipes and tees**.** All intermediate files necessary to the processes are handled automatically. To illustrate these concepts:

The output of **who** may be piped to the input of **pr**, in order to format it differently, and then to **wc**, in order to count the lines. The command **pr** is a filter.

who | pr | wc -l

In order to tap into the stream between **who** and **pr**, a tee is inserted. The output of **who** will go to the file **whom** and also to the input of **pr**.

who | tee whom | pr | wc -l

Examples of filters are **cat**, **grep**, **nroff**, **pr**, **rev**, **sed**, **sort**, **tail**, **tr**, **uniq**, and **wc**, some of which are described here**.**

Process Status

The command **ps** will allow one to monitor the progress of processes on the system. If one has programs or other processes running in the background or in batch mode, useful information such as the amount of CPU time used may be obtained. The process identification number, the terminal initiating the command, the CPU time expended, and the command which initiated the process are all given.

If the option **-f** is selected, the status, priority, and size of the processes also are listed.

Information about the status of the NQS batch queues can be obtained using the **qstat** and **nqslimits** commands. See the online **man** pages for details.

Shell Scripts

A shell script is a file that contains one or more UNIX commands. New shells can be created by users issuing the ***sh*** *command.*

A shell is the process which interacts with the commands issued by a logged-on user. Each logged-on user has his own shell. A new shell (and even multiple new shells) can be created by a user with the command **sh**. The command can also execute shell scripts.

If the file **check** contained the lines

```
#
date
who
```

then the command **sh check** would create a separate shell from the one the user is operating under, execute the commands (printing the date and the current users on the screen), and return to the user's original shell. The new shell would disappear as soon as the commands were finished executing.

The command **sh** alone, without any arguments, creates new shells that the user can treat as if they were his original shell. Each is a separate environment in which commands may be issued and new processes launched. **sh** may be given repeatedly, a new shell being created each time. The command CTRL-D returns the user to the previous shell, a series of CTRL-Ds returns the user to the original sign-on shell. Issuing one more CTRL-D signs off the user .

Sample Interactive UNICOS Session

The following section demonstrates the use of some of the basic UNICOS commands on the Cray C90.

```
% telnet asnc90.asc.edu
Trying...
Connected to asnc90.asc.edu.
Escape character is '^]'.

Cray UNICOS (asnc90) (ttyp026)


login: asndsc01
Password:
Last successful login was : Wed Aug  8 15:01:16 from 129.66.32.250

        ****************************************************************
        *              Welcome to the Alabama Research and Education Network      *
        *                                 UNICOS 8.0.3                                *
        ****************************************************************


                    This is a State of Alabama computer facility managed by
                    the Alabama Supercomputer Authority and its contractor
                    Nichols Research Corporation.  The system is for use by
                    State of Alabama educational institutions and authorized
                    commercial accounts.

                    In accessing the system, users consent to the monitoring
                    of usage for purposes of accounting and the detection of
                    unauthorized access.  Please note that access to this facility
                    must be specifically authorized by ASA and that unless you are
                    so authorized, access and any other use may expose you to
                    criminal and/or civil proceedings.


*********************************************************************
                          Messages may appear here
*********************************************************************
You have mail.

asnc90$ who                    <----- command to find out who is logged on
```

```
asnegb01        ttyp000         Aug  8 14:21  (EricB.TUCC.UAB.EDU)
root            console         Aug  7 09:55
usawaw01        ttyp001         Aug  8 13:34  (192.245.222.13)
asnphd01        ttyp002         Aug  8 14:42  (asnfddi.asc.edu)
uahycz01        ttyp003         Aug  8 13:36  (cmmr.uah.edu)
aubmlm01        ttyp004         Aug  8 12:41  (mckee.chem.auburn.edu)
asnphd01        ttyp005         Aug  8 11:54  (nrcsgi1.asc.edu)
uabgxt01        ttyp006         Aug  8 14:46  (picasso.chem.uab.edu)
oper            ttyp008         Aug  7 13:44  (129.66.32.212)
asnphd01        ttyp009         Aug  8 15:00  (asnc90.asc.edu)
asnegb01        ttyp010         Aug  8 08:29  (EricB.TUCC.UAB.EDU)
asndlm01        ttyp014         Aug  8 10:10  (129.66.32.225)
asnjdb01        ttyp015         Aug  8 15:01  (becker.csc.usouthal.edu)
crirxh01        ttyp016         Aug  8 08:49  (cri.asc.edu)
uabtxb01        ttyp017         Aug  8 11:16  (138.26.83.1)
asnakm01        ttyp018         Aug  7 12:22  (columbia.asnvis.ua.edu)
uahycz01        ttyp020         Aug  8 14:56  (cmmr.uah.edu)
crirxh01        ttyp021         Aug  8 10:18  (cri.asc.edu)
oper            ttyp022         Aug  8 11:08  (129.66.32.212)
aamgsl01        ttyp023         Aug  8 11:48  (198.180.132.250)
uabsck01        ttyp024         Aug  8 15:00  (ftp.PHY.UAB.EDU)
aubfxs01        ttyp025         Aug  8 13:28  (kharazmi.eng.auburn.edu)
asndsc01        ttyp026         Aug  8 15:02  (asnmail.asc.edu)
ualhxk01        ttyp029         Aug  8 14:07  (uarrp.gri.ua.edu)
uahymc01        ttyp030         Aug  8 14:17  (ebs330.eb.uah.edu)
```

asnc90$ **finger paul** **<----- command to identify a user or users id**

```
Login name: asnphd01                In real life: Paul H Duggan
Directory: /home/staff/asnphd01         Shell: /bin/csh
On since Aug  9 08:48:55 on ttyp020 from nrcsgi1.asc.edu
1 hour 7 minutes Idle Time
No Plan.

Login name: uahpmd01                In real life: Paul M. Dean
Directory: /home/uah/uahpmd01           Shell: /bin/csh
Last login Thu Feb 23 08:13 on ftpd from ebs330.eb.uah.edu
No Plan.

Login name: usapny01                In real life: Paul N. Younger
Directory: /home/usa/usapny01           Shell: /bin/csh
Last login Thu May 26, 1995 on /dev/ttyp002 from usaaxp.usa.edu
No Plan.

Login name: uabpko01                In real life: Paul K. Owens
Directory: /home/uab/uabpko01           Shell: /bin/csh
Last login Mon Sept 16 17:11 on /dev/ttyp011 from display.asnvis.uab.edu
No Plan.
```

asnc90$ **ls -l** **<----- command to list files**

```
total 2520
-rwx------      1 asndsc01 staff          214840      Aug  8 14:58 a.out
-rw-r-----      1 asndsc01 staff          278         Aug  8 14:00 c94.job
-rw-r-----      1 root     staff          1993        Aug  8 13:46 c94.job.o38160
drwx------      2 asndsc01 staff          4096        Aug  8 11:57 hello
-rw-------      1 asndsc01 staff          2011        Aug  8 13:56 hello.out
-rw-------      1 asndsc01 staff          77          Aug  8 14:58 name.c
-rw-------      1 asndsc01 staff          75          Aug  8 14:51 name.p
drwxr-xr-x      2 asndsc01 staff          4096        Aug  3 09:55 ncube
```

asnc90$ **cd hello** **<----- command to change directories**

asnc90$ **pwd** **<----- command to print working directory**

```
/usr/staff/asndsc01/hello
```

asnc90$ **ls -l** **<----- command to list files**

```
total 2008
-rwx------      1 asndsc01 staff          1023960     Aug  8 11:57 a.out
-rw-r-----      1 asndsc01 staff          75          Aug  8 11:50 hello.f
```

asnc90$ **cat hello.f** **<----- command to list contents of file hello.f**

```
      program hi
      print*, 'Hello '
      stop
      end
```

asnc90$ **cat hello.f >hello.newf** **<----- list hello.f to hello.newf**

asnc90$ **cat hello.newf** **<----- list hello.newf**

```
      program hi
      print*, 'Hello '
      stop
      end
```

asnc90$ **mkdir fortran** **<----- command to make a directory**

asnc90$ **ls -l** **<----- list files**

```
total 2024
-rwx------      1 asndsc01 staff          1023960     Aug  8 11:57 a.out
drwx------      2 asndsc01 staff          4096        Aug  8 15:45 fortran
-rw-r-----      1 asndsc01 staff          75          Aug  8 11:50 hello.f
-rw-------      1 asndsc01 staff          75          Aug  8 15:45 hello.newf
```

asnc90$ **cd fortran** **<----- change to directory named fortran**

asnc90$ **ls -l** **<----- list files**

```
total 0                          <----- there are no files in this directory at this time
asnc90$ pwd                      <----- print working directory
/usr/staff/asndsc01/hello/fortran

asnc90$ cp /usr/staff/asndsc01/hello/hello.f hello.ff      <----- copy file hello.f from
                                                                  parent directory to hello.ff in
                                                                  new  directory
asnc90$ ls                       <----- list file to see if hello.ff was copied to the new
                                        directory
hello.ff

asnc90$ cat hello.ff             <----- list contents of hello.ff
      program hi
      print*, 'Hello '
      stop
      end

asnc90$ date                     <----- prints current date on screen
Wed Aug  8 15:49:10 CST 1995

asnc90$ ls -l                    <----- list files
total 8
-rw-r-----          1 asndsc01 staff              75              Aug  8 15:48 hello.ff

asnc90$ chmod u+x hello.ff       <----- change permissions on hello.ff

asnc90$ ls -l                    <----- list file to see new permissions
total 8
-rwxr-----   1 asndsc01 staff        75 Aug  8 15:48 hello.ff

asnc90$ cd ..                    <----- change to parent directory

asnc90$ pwd                      <----- print to check name of parent directory
/usr/staff/asndsc01/hello

asnc90$ rmdir fortran            <----- command  to delete a directory
rmdir: fortran -- Directory not empty

asnc90$ cd fortran               <----- change to directory named fortran to delete
                                        contents

asnc90$ ls                       <----- list files
hello.ff
```

```
asnc90$ rm hello.ff              <----- command to remove the file named hello.ff

asnc90$ cd ..                    <----- change to parent directory

asnc90$ rmdir fortran            <----- remove or delete directory named fortran

asnc90$ ls -l                    <----- list file to see if fortran directory was deleted
total 2016
-rwx------      1 asndsc01 staff        1023960      Aug  8 11:57 a.out
-rw-r-----      1 asndsc01 staff        75           Aug  8 11:50 hello.f
-rw-------      1 asndsc01 staff        75           Aug  8 15:45 hello.newf

asnc90$ exit                     <----- command to log off or end session on Cray
Connection closed.
%
```

Introduction to Line-Oriented Text Editor ex

*The **ex** text editor is powerful and useful in an environment in which the use of a full screen editor is not practical.*

(Reproduced with permission of the publisher, Howard W. Sams and Co., Indianapolis Indiana, UNIX System V Primer, Waite, Augtin and Prata, © 1984.)

To invoke **ex** type the following command:

ex ***[options] filename***

(**ex** can also be accessed from within **vi**, the visual editor; See the description below of **vi**.) The options are as follows:

-	A minus suppresses messages that would be sent by the editor during an interactive session.
-v	Calls up the **vi** editor
-t ***tag***	Allows one to edit the file containing the tag, positioning the editor at the tag
-r ***file***	Recovers the named file after a system crash. If no file is specified, a list of recoverable files is given
-R	Allows the file to be read only locking out accidental editing
+***command***	Editing begins immediately with the command given
-L	Begin LISP mode. LISP programs will be properly aligned.

*The **ex** editor has three different modes: the Command, the Insert, and the Visual Modes. The command mode is the normal mode in which one enters the editor.*

-X Begin encryption mode. (Edit an encrypted file.)

In the **command mode**, the editor prompts with a colon for the commands summarized under command mode. This is the normal mode in which one enters the editor.

The **text input or insert mode** is invoked with one of three commands **a**, **i**, or **c**, as summarized under Text Input Mode.

The **visual mode,** invoked with **vi**, enters **vi**; **ex** is reentered by typing **Q** or **e** in **vi**'s command mode.

Sample Session

The user should be in the directory in which the new file is to be located.

To begin editing a new file called **test** in the current working directory, type:

```
$ ex test
```

The editor will respond with

```
"test"[Newfile]
:
```

To insert text, use the append command, **a,** at the colon prompt**:**

```
:a
Fourscore and seven years ago,
our forefathers brought forth
on this continent  (etc)

guv'ment for the people,
of the people, by the people
shall not perish  (etc)
.
```

New text will be appended to the current line position (in this case the top of the file) until a single period and only a single period, is entered on a line. Then insert mode is terminated, and the colon prompt signifying command mode is returned. Text can be saved by writing it with the **w** command and exiting the editor with the **q** command.

The user can now edit **test** and change some of the text. Enter

```
$ ex test
(editor responds with filename and
number of lines
and characters in the file)
:
```

Move to the fifth line of the file by giving four carriage returns or by having the editor print line 5 (**5p**). Then use the substitute command to change **guv'ment** to **government:**

```
:5p
guv'ment for the people
:s/guv'ment/government/
government for the people
:
```

Note that the line is displayed after the substitution. The change could be saved with a **w** command.

Text Input Mode

a **a**ppend lines after line dot, unless line number is specified; e.g. **3a** adds new text after line 3.

i **i**nserts lines before line dot, unless line number is specified; e. g. **3i** adds new text before line 3.

c **c**hange line dot or lines specified in address; e. g. **2,4c** deletes lines 2 through 4 and adds new text that is typed in.

Command Mode

p — **p**rint on the screen specified lines. If no lines are specified, print the current line. **2,4p** prints lines 2, 3, and 4.

d — **d**elete specified lines. If none specified, delete line dot; e. g. **5,8d** deletes lines 5 through 8.

m — **m**ove specified lines to line named after **m**; e. g. **1,2m5** moves lines 1 and 2, placing them after line 5.

co — **co**py specified lines to line named after **co**; e. g. **2,4co$** copies lines 2 through 4, placing them at the end of buffer.

r ***filename*** — **r**eads in the contents of ***filename*** at current line or at line specified.

S/*one*/*two*/ — **s**ubstitutes the word *"two"* for the word *"one"* for the first occurrence in the specified lines.

/***nice***/ — Searches for the next line to contain the word(s) between the slashes: in this case, the string "*nice*".

g — **g**lobal search or substitute generally used with **s** or **s/pat1/pat2/g**. Substitutes "pat2" for "pat1" for all occurrences of "pat 1" in the specified lines.

nu — **nu**mber the lines and print them on the screen; e.g..**1,$nu** numbers and prints all the lines in a file.

u — **u**ndo the last change made in the buffer.

z — Print lines on the screen; e. g., **3z** prints the file on screen starting with line 3

*There are many good books with tutorials on the use of the editor **ex.** One such is published by Cray Research, Inc. and is titled UNICOS Text Editors Primer (publication number SG-2050).*

Leaving the editor

w	**w**rite the specified lines that are addressed to a named file; e. g., **2,5w popcorn** writes lines 2 through 5 into file **popcorn**.
q	**q**uit.
q!	**q**uit without writing changes to file.

Addressing Lines

.	This character addresses the current line, called "line dot". The current line is the last line affected by a command. Thus, .p prints the current line.
.=	This command prints the line number of the current line; e. g., editor responds with a number like "5".
$	This character addresses the last line in the buffer; e. g., $d deletes the last line.
n	A decimal number n addresses the nth line; e. g., 3p prints line number 3.
+ -	The + and - are used in conjunction with a reference line which may be specified with n or $ or current line; e. g., $-5,$p prints the last six lines of the buffer.
<return>	When used with no command, it is equivalent to the next line. Very useful for stepping through the buffer.

***Note:** If users accidentally hit **o** or **vi** while in **ex**, the way to get back to **ex** is to hit the **<esc>** key, then the **Q** key.*

Introduction to the Screen-Oriented Editor vi

*The visual editor **vi** must be customized to a particular visual terminal. **vi** cannot be used with an IBM 3270 terminal.*

(Reproduced with permission of the publisher, Howard W. Sams and Co., Indianapolis Indiana, UNIX System V Primer, Waite, Augtin and Prata, © 1984.)

Sample vi Session

Edit a file in the directory where the file is located or should be located (in the case of a new file). Begin editing a new file by typing the following:

$ vi *file*

Insertion of text can be done after issuing the **i** command:

i
Augy had a little lamb,
its fleece was white as snow (etc)
<esc>

Move about in the file with the **h,j,k,and l** keys when in command mode (the <esc> returns one to command mode).

Cursor Positioning Keys--in the Command Mode

<h>	Moves cursor one character to the left.
<j>	Moves cursor down one line anywhere in text.
<k>	Moves cursor up one line anywhere in text.
<l>	Moves cursor one character to the right.

Move with these keys to the "f" in "fleece" in the second line, and type **cw.** The word "fleece" disappears, replaced by a $. Type "fur" in substitution.

To save the file, with changes, type **ZZ**. This exits to the shell.

Entering text input mode--End this mode with an <esc>

a **a**ppends text after the cursor. Enter as many lines and<return>'s as needed.

i **i**nserts text before the cursor. Enter as many lines of text and <return>'s as needed.

o **o**pens a new line below cursor. Ready for the text input.

O **O**pens a new line above cursor. Ready for the text input.

R **R**eplaces characters on the screen, starting at the cursor, with any characters typed.

These commands, after execution, return the editor to the command mode

r **r**eplaces a single character under the cursor with a single character that is typed.

/happy Search sequence; looks for next occurrence of pattern following / (in this case, the word "*happ*y")

?lark Search sequence; like /, but searches backwards from the cursor.

n Used after / or **?** to advance to the next occurrence in the buffer of the pattern.

u **u**ndo the last command.

U **U**ndo all the changes to the current line.

x Deletes character under the cursor.

<del> or # or <CTRL-H> or <rub>
This backspace feature of the shell also works in the editor. These commands move the cursor character by character, left within a line, erasing each character from the buffer.

<CTRL-F> Scrolls or pages the screen forward one page at a time.

<CTRL-B> Scrolls or pages the screen backward one page at a time.

<CTRL-D> Scrolls or pages the screen down one-half page at time.

<CTRL-U> Scrolls or pages the screen up one-half page at time

<CTRL-G> Identifies the line where the cursor is located by line number.

nG Positions the cursor at line **n** in the file.

Operators in the Command Mode

d **d**eletes indicated text starting at the cursor. For example, use **dw** to delete a word and **dd** to delete a line; **3dd** deletes 3 lines. Deleted text is stored temporarily in a buffer whose contents can be printed out with the **p** command. Also, **d** can be used with named buffers in the manner described for **y** command.

c Deletes indicated text starting at the cursor and enters Text Input Mode. Thus, **cw** deletes from the cursor to the **end** of the word, allowing users to add text between those positions.

y Copies indicated text, starting at the cursor, and stores it in a buffer. There are nine unnamed buffers (1-9) that store the last nine **d**elete or **y**ank operations, and 26 named buffers (a-z) that can be used for storage. The double quote mark (") is used to tell the editor the name of the buffer. Thus, **"cy$** will store text from the cursor to the end of the line in a buffer named **c**.

p The **p**ut command, used to **p**ut down "delete" and "yank" buffer contents after the cursor or on the next line. Command **p** puts the last item yanked or deleted back into the file just after the cursor, and **"cp** will put the contents of buffer **c** after the cursor.

Scopes for Use with Operators

e The scope from the cursor to the **end** of the current word; e. g., if the cursor is on the "u" in "current", and the user types **de**, then "urrent" is deleted.

w The scope is from the cursor to the beginning of the next **w o r d**, including the space.

b The scope is from the letter before the cursor, backwards, to the **beginning** of the word.

$ The scope is from the cursor to the end of the line.

O The scope is from just before the cursor to the beginning of the line.

) The scope is from the cursor to the beginning of the next sentence. A sentence is ended by ".", "!", or "?", followed by 2 spaces or by an "end of line" (provided by the <return> key).

(The scope is from just before the cursor back to the beginning of the sentence containing the cursor.

} The scope is from the cursor to the end of a paragraph. A paragraph begins after an empty line.

{ The scope is from just before the cursor back to the beginning of a paragraph.

Leaving the Editor

<esc>:w Write the contents of the buffer into the current file of the same name. Can write to a new filename. Also, can send partial buffer contents using line numbers, such as **A:3,10w popcorn**.

<esc>:q Quit the buffer after a :**w** command.

<esc>:wq Write and quit, placing buffer contents in file.

<esc>:q! Quits buffer without making changes in file. ***Dangerous***.

<esc>ZZ Write and quit, placing buffer contents in file.

Using the ex editor while in vi

: Gives a colon (:) prompt at the bottom of the screen and lets users make one **ex** command. Users are returned to the **vi** mode when the command finishes execution.

Q **Q**uits **vi** and places users in the **ex** editor, giving users a command mode prompt, the colon (:) at the bottom of the screen. Users can get back to **vi** while in the command mode.

When in Doubt

<esc> Puts users in the command mode.

There are many good books with tutorials on the use of the editor vi. One such is published by Cray Research, Inc. and is titled UNICOS Text Editors Primer (publication number SG-2050).

Compiling and Running Programs

There are three compilers available on the ASC Cray C90.

C (cc)
FORTRAN 90 (f90)
FORTRAN 77 (cf77)

This section provides examples on running programs using the F90 and C compilers. The CF77 compiler is in maintenance mode and is being phased out. Batch processing uses the same commands embedded in the batch file.

FORTRAN Programs

The **f90** is a full ANSI FORTRAN 90 compiler, with enhanced optimization and vectorization capabilities; **f90** is the recommended compiler for the majority of users. **f90** is the newly-developed FORTRAN autotasking compiler which can automatically optimize FORTRAN code for multi-processor Cray systems; **f90** offers a variety of options for experienced users.

The user should check the manual for precise use of the various options. Online documentation is available with the **man f90** command.

The sequence for running a FORTRAN program is as follows:

```
asnc90$f90 source.f
```

*source must be in a **.f** file*

```
asnc90$./a.out
```

*f90 produces **a.out** executable*

Example of Compiling and Running a Program Using f90

```
asnc90$ ls
hello.f
asnc90$ f90 hello.f
asnc90$ ls -l

total 2016
-rwx------   1 asndsc01 staff    1023960 Aug  9 15:16 a.out
-rw-r-----   1 asndsc01 staff        .....75 Aug  9 11:50 hello.f

asnc90$ ./a.out
 Hello from the C90!
 STOP executed at line 3 in Fortran routine 'HI'
 CP: 0.002s,  Wallclock: 0.035s,  2.2% of 2-CPU Machine
 HWM mem: 131987, HWM stack: 2048, Stack overflows: 0
asnc90$
```

The default executable file is ***a.out***. This can be changed with the **-o** option:

```
asnc90$f90  -o  myexec.hello hello.f

asnc90$./myexec.hello
```

The listing file is controlled by the **-e** and **-r** options. The default is no listing. The form is:

```
asnc90$f90  -e options  -r listfile hello.f
```

-e controls the options to be turned on; **-d** controls the options to be turned off.

C Programs

*The Cray C compiler is invoked with the **cc** command. It has enhanced optimization and vectorization features and complies with the ANSI-C standard.*

Online documentation is available with the **man cc** command.

The C compiler is called with the **CC** command:

asnc90$**cc source.c**
compile source.c,
*produces executable **a.out***

asnc90$./**a.out** <=== execute the program

It is not necessary to call the segldr if the C compiler is called without options and with a single input file.

Example of Listing, Compiling, and Running a C program

```
asnc90$ ls
hello.c
asnc90$ ls -l
total 8
-rw-------   1 asndsc01 staff          77        Aug  9 15:40 hello.c
asnc90$ cat hello.c
#include <stdio.h>
main(){
printf ("Hello World .....from C\n");
return;
}

asnc90$ cc hello.c
asnc90$ ls -l
total 432
-rwx------   1 asndsc01 staff      214840        Aug  9 15:40 a.out
-rw-------   1 asndsc01 staff          77        Aug  9 15:40 hello.c
asnc90$ ./a.out
Hello World .....from C
asnc90$
```

Batch Processing

UNICOS provides a facility called NQS (Network Queuing System) for the submittal of 'batch' jobs. A batch job is a standard UNICOS command stream.

Batch jobs may be submitted using the **qsub** command. For example:

asnc90$ **qsub -q express_a myscript**

asnc90$ **qsub -lm 12mw -lt 600 myscript**

File-Naming Precautions

When users are running multiple batch jobs, they must ensure that file names are unique.

Under UNICOS, a user's files are shared by all batch and interactive jobs running under the same userid. Hence two separate jobs running simultaneously under the same userid, each writing to file abc, will write to the same file and produce garbage. One way to avoid doing this is to qualify file names by the job process id, which is accessed through the shell variable ***$$***.

For example, instead of abc.xyz, use **abc.$$.xyz**.

Another method, illustrated below, is to create a directory qualified by the process id, and place all temporary files in that directory.

Files created by batch jobs are **NOT** automatically removed at job termination unless they are located in the **$TMPDIR** directory. Any files in **$TMPDIR** are removed upon exiting a session. Files not in **$TMPDIR** stay in the created working directory until explicitly deleted with the **rm -r** command.

Preparing a Batch Job

A batch job is essentially a shell script file that is submitted via NQS to the batch queues.

A sample batch script follows:

```
asnc90$ cat xmp.job

#!bin/sh
set -x <-- echo statements to

cd /tmp/asnger08
cd plots
f90 fplot.f
#
# end batch script
#
```

Using qsub

The **qsub** command is issued by a user to submit a UNICOS file into the batch execution queue. By default, all outputs - **stdout and stderr** - remain on the C90 as files ***filename.oprocess*** and ***filename.eprocess.*** However, the Bourne shell variable **$ROUT** can be defined to ensure that such output is returned to the remote node. For complete information on qsub, see **man qsub**. The format of the qsub command is:

qsub [options] file

options

There are various options to control the batch job. Most of these options can also be included in the batch job stream as **#QSUB** option.

Some important options are:

-eo
directs stderr output to the same destination as stdout

-o filename
directs stdout to the stated file destination

Note:
By default, under NQS all stdout and stderr I/O is spooled into a private NQS directory. The output files are returned ***upon job completion****.*

The user should redirect the job I/O at run-time in order to prevent being unable to review any results until job completion. To redirect job I/O:

./a.out > /tmp/stdout.$$ 2>stderr.$$

-lt cputime
set process time limit to cputime CPU seconds

-lm memsize
set process memory limit to memsize words

file
the UNICOS command file to submit to the batch queue

Example:
A job file C90.job is submitted to the **express** queue by choosing memory and time limits associated with the **express** queue, and directing both **stderr** and **stdout** to the same output file (**-eo** option):

```
asnc90$qsub -eo -lt 500 -lm 5mw -q batch C90.job
Request 8602.asnC90 submitted to queue: express.
asnc90$

asnc90$qstat -r
------------------------------
NQS 61.2 BATCH REQUEST SUMMARY
------------------------------
IDENTIFIER    NAME    USER     QUEUE                JID  PRTY REQMEM REQTIM ST
------------ ------- -------- -------------------- ---- ---- ------ ------ ---
 8602.asncray C90.job asnjsn01 express              2357   15   5000    600 R03
no pipe queue entries
no device queue entries
asnc90$
```

Output shows up in file C90.job.o<pid>

```
asnc90$
asnc90$ls -l xmp*
-rw-r--r--   1 asnjsn01 root          79 Jun 14 18:03 C90.job
-rw-r--r--   1 asnjsn01 root        5569 Jun 14 18:03 C90.job.o8602

asnc90$cat C90.job.o8602

[.....message of the day.........}
You have mail.
news: gaussian CF7750.news amber g92

Mon Jun 3 18:03:16 CDT 1996

+ cd /tmp/asnjsn08
+ cd plots
+ f90 fplot.f

            [ output from the job shows up here ]


+ test -n
+ test -n BATCH
+ test BATCH = BATCH
asnc90$
```

Selecting a Job Priority

Job Priority Selections:

Express
1.4 times batch cost

Prime
1.0 times batch cost (default)

Non-Prime
0.6 times batch cost

Jobs submitted to the express queue are automatically charged at 1.4 times the prime time rate. Interactive and batch jobs run between 8:00 a.m. and 6:00 p.m. are charged at the prime rate. Interactive sessions and batch jobs run between 6:00 p.m. and 8:00 a.m. are charged at the non-prime rate.

The **nqslimits** command will display a list of the available NQS queues, their respective priorities, and limits for CPU time, memory usage and file size.

ftp (File Transfer Protocol)

Files can be transferred between systems on the network using the **ftp** command. Files are usually transferred as ASCII files. Binary files can also be transferred. Use of **ftp** requires a valid userid and password on the remote system. The commands covered in this section are:

ftp	Establish a remote connection
get	Move a file from the remote host to the local host
put	Move a file from the local host to a remote host
mget	Retrieve multiple files from a remote host
mput	Send several files to a remote host
mkdir	Create a new directory on a remote host
cd	Change directories on a remote host
dir	List files in the current remote directory
quit	Exit from ftp

For additional information about the ftp command, enter **man ftp**.

Connecting to a Remote System

To establish a connection to a remote system, use the **ftp** command. After the connection is established, provide a valid userid and password:

```
>> ftp asnc90.asc.edu
Connected to asnc90.asc.edu.
220 asnc90 FTP server (Version 5.2 Fri
Sep 7 14:09:58 CDT 1995) ready.
Name (asnc90.asc.edu:asnxyz01): asnxyz01
331 Password required for asnxyz01.
Password:
230 User asnxyz01 logged in.
ftp>
```

The get Command

The **get** command is used to transfer a file from the remote system to the local system. The syntax is:

get *remotefilename localfilename*

If the local file name is omitted, the remote file name will be used for both files.

```
ftp> get file1.x
200 PORT command successful.
150 Opening ASCII mode data connection
for file1.x (14 bytes).
226 Transfer complete.
local: file1.x remote: file1.x
15 bytes received in 0.04 seconds (0.37
Kbytes/s)
ftp>
```

The put Command

The **put** command is used to send a file from the local host to the remote host. The syntax is:

put *localfilename remotefilename*

If the remote file name is omitted, the local file name will be used for both files.

```
ftp> put xyz.x file3.x
200 PORT command successful.
150 Opening ASCII mode data connection
for file3.x.
226 Transfer complete.
local: xyz.x remote: file3.x
15 bytes sent in 1e-06 seconds (1.5e+04
Kbytes/s)
ftp>
```

The mget Command

The **mget** command is used to transfer multiple files from the remote host to the local host. Standard UNIX 'wildcard' characters can be used. The syntax is:

mget *wildcardname*

for example ***mget file**** will transfer all files beginning with the characters 'file'.

```
ftp> mget file*
200 PORT command successful.
150 Opening ASCII mode data connection
for file1.x (14 bytes).
226 Transfer complete.
local: file1.x remote: file1.x
15 bytes received in 0.04 seconds (0.37
Kbytes/s)
200 PORT command successful.
150 Opening ASCII mode data connection
for file2.x (14 bytes).
226 Transfer complete.
local: file2.x remote: file2.x
15 bytes received in 0.05 seconds (0.29
Kbytes/s)
ftp>
```

The mput Command

The **mput** command is used to transfer multiple files from the local host to the remote host. Standard UNIX 'wildcard' characters can be used. The syntax is:

mput *wildcardname*

For example **mput file*** will transfer all files beginning with the characters 'file'.

```
ftp> mput file*
200 PORT command successful.
150 Opening ASCII mode data connection
for file1.x.
226 Transfer complete.
local: file1.x remote: file1.x
15 bytes sent in 1e-06 seconds (1.5e+04
Kbytes/s)
200 PORT command successful.
150 Opening ASCII mode data connection
for file2.x.
226 Transfer complete.
local: file2.x remote: file2.x
15 bytes sent in 1e-06 seconds (1.5e+04
Kbytes/s)
ftp>
```

The mkdir Command

The **mkdir** command is used to create a new subdirectory on the remote host. The syntax is:

mkdir *new-sub-dir*

```
ftp> mkdir demoftp
257 MKD command successful.
ftp>
```

The cd Command

The **cd** command is used to change directories on the remote host. The syntax is:

cd *dir-name*

```
ftp> cd demoftp
250 CWD command successful.
ftp>
```

The dir Command

The **dir** command is used to get a listing of the files in the current remote directory. The syntax is:

dir

```
ftp> dir
200 PORT command successful.
150 Opening ASCII mode data connection for /usr/ucb/ls.
total 8
-rw-r-----  1 asnger01 u_staff        14 Jun 13 15:58 file1.x
-rw-r-----  1 asnger01 u_staff        14 Jun 13 15:58 file2.x
226 Transfer complete.
135 bytes received in 0.13 seconds (1 Kbytes/s)
ftp>
```

The quit Command

The **quit** command exits from the **ftp** session. The syntax is:

quit

```
ftp> quit
221 Goodbye.
>>
```

Printing

ASC CRAY C90 users may print output at the Alabama Supercomputer Center in Huntsville and at some of the network sites.

The lpr Command

The **lpr** command is used to print files from the C90 system. The form is

lpr [options] filenames

Options:

-Pprinter	Print on device ***printer***. ***printer*** must be in the /etc/printcap table on the C90.
-r	Remove the file upon completion of spooling or upon completion of printing (with the -s option).
-m	Send mail upon completion.
-#num	Print ***num*** copies of each file specified.

For additional options, use ***man lpr***.

If a user tries to spool a file that is too large, it will be truncated.

lpr will NOT print binary files.

Printing at the Central Site

The ASC is equipped with a QMS 2025 laser printer. The QMS is capable of printing on either 8-1/2x11 or 11x17 paper, vertically or horizontally, at 10, 12, and 15 characters/inch. Output printed at the ASC will be filed there or mailed to the user, as requested on the output. All jobs initiated at a remote site are expected to be printed at the remote site.

The command below shows the available format for central site printing. The default is **lp**, standard letter form.

lpr myfile1 myfile2

Remote Site Printing

UNICOS files may be printed at selected remote sites. Check with the site application analyst for more information about using remote site printers.

Users may also print locally by moving a file to a remote machine with the **ftp** command and printing using the remote print command.

Optimization and Vectorization on the C90

This section presents an introduction to the C90 optimization concepts. Optimization refers to any technique used to improve the performance of a user's program, either to reduce resource consumption, to improve job turnaround, or to reduce job cost. Among the techniques used to achieve this are reducing the CPU time by more effective coding, memory management techniques, and I/O management techniques.

Vectorization is a specific optimization technique aimed at producing a program that makes maximum use of the vector capabilities of the C90. Since most users are processing with FORTRAN programs, they must rely on the tools provided within the FORTRAN compilers (F90 and CF77) to achieve maximum performance.

An additional method of improving vector performance is to use code that has already been vectorized; use of the IMSL Parallel Libraries, BCSLIB and SCILIB libraries can produce significant performance improvements.

Enhanced I/O performance is obtained by using the **/tmp** directory. Adjusting file types and buffer sizes may also improve I/O performance.

Remember that optimizing a program is in itself a source of cost. Take care in the pursuit of efficiency so that the cost of improving the code does not exceed the savings provided by the improvements. In general, the first large improvement in performance is the easiest to obtain. Successive increments of improvement become increasingly difficult and expensive to obtain. Knowing when to quit is every bit as important as knowing where to start.

The following Cray documents will provide additional information and are available through the local applications analyst.

Many of these are readily available online via 'docview' or through the CrayDoc (tm) online browser for Sun and SGI workstations (if CrayDoc is installed on a server on campus).

Cray Standard C Reference Manual, publication SR-2074

UNICOS System Libraries Reference Manual, publication SR-2080

UNICOS Performance Utilities Reference Manual, publication SR-2040

UNICOS CDBX Symbolic Debugger Reference Manual, publication SR-2091

CF77 Ready Reference, publication SQ-3770

CF77 Commands and Directives, publication SR-3771

CF77 Fortran Language Reference Manual, publication SR-3772

CF77 Optimization Guide, publication SG-3773

I/O User's Guide, publication SG-3075

CF90 Ready Reference, publication SQ-3900

CF90 Commands and Directives Reference Manual, publication SR-3901

CF90 Fortran Language Reference Manual, publication SR-3902

Introducing the Cray TotalView Debugger, publication IN-2502

Compiler Information File (CIF) Reference Manual, publication SR-2401

Math Library Reference Manual, publication SR-2138

Scientific Libraries Reference Manual, publication SR-2081

Application Programmer Library Reference Manual, publication SR-2165

Vectorization

The Cray C90 compilers attempt to produce vector code for loops. C attempts to vectorize **FOR loops.** CF77 and F90 attempt to vectorize inner **DO loops** and some **IF loops**. Additionally CF77 attempts to microtask code at the **DO loop** level.

Vectorization in C

The C compilers attempt to vectorize all FOR loops encountered in a program. To disable this feature use the ***-h novector*** option. Both pointer and non-pointer loops will be checked.

Vectorization in CF77 and F90

The CF77 and F90 compilers attempt to vectorize **DO loops** and certain **IF loops**. Vectorization is controlled by the **-o** option of the CF77 command and the **-O** option of the F90 command. The default is full optimization. For best optimization the use of the CF77 compiler interface is preferred because it includes the load step by default.

Vectorization Inhibitors

In general, the following will inhibit vector processing in a loop:

Procedure calls, including I/O such as **READ** or **WRITE**

Function calls to other than the standard mathematical functions

Any statement other than an assignment statement, an IF statement or a compound statement containing only assignment or IF statements

Run-time array bounds checking

Dependencies

A dependency is an expression in a loop which requires the value of an expression from a previous iteration of the loop. Dependencies inhibit vectorization because vector processing changes the order of statement execution and values from previous iterations may not be available.

The following dependencies (also called recurrence) will inhibit vectorization for all compilers:

a(i)=a(i-1) in a incrementing loop

a(i)=a(i+1)in a decrementing loop

a(i)=a(i+j) in any loop if the value of j is not known (ambiguous subscript)

Other variations exist; check the appropriate compiler reference manual for more details.

Vectorization Techniques

The following steps are recommended to improve the performance of a program by increasing the amount of vector code:

Get the program running correctly **before** starting optimization.

Use **flowtrace** to determine which routines use the highest percentage of CPU time.

Modify the selected routines. This may require rearranging code, rewriting code, or changing algorithms. Use of the VectorPak routines is recommended.

/tmp use

*The **$TMPDIR** working directory on /tmp is automatically created at the beginning of a session and automatically removed upon exiting a session on the Cray C90.*

To use /tmp, simply copy the desired data files into /tmp before use and move any output files back to the $HOME or dmf-stage directory when processing is complete. **Files in /tmp are deleted after 24 hours to maintain an adequate amount of space.**

Examples of /tmp Use

There are two different ways to create working directories on /tmp. One way is to use the temporary directory automatically created by the system for a session under $TMPDIR. The other is to go to /tmp and create a working directory.

The use of $TMPDIR as the working directory is the preferred method since this directory is automatically removed by the system at the end of the session.

USING $TMPDIR

Change directory to $TMPDIR
```
asnc90$ cd $TMPDIR
```

Copy files from the $HOME directory to /tmp
```
asnc90$ cp $HOME/prog.f $TMPDIR
```

Copy any input data files to /tmp
```
asnc90$ cp $HOME/mydata $TMPDIR
```

Compile the program
```
asnc90$ f90 prog.f
```

Execute the program
```
asnc90$ ./a.out
```

Copy the files in $TMPDIR back to $HOME
```
asnc90$ cp * $HOME
```

Change directory to $HOME
```
asnc90$ cd
```

Log off or Exit the Session
```
asnc90$ exit
```

Creating a Working Directory on /tmp

Create a working directory

```
asnc90$ mkdir /tmp/asnusr01
```

Change directory to /tmp/asnusr01

```
asnc90$ cd /tmp/asnusr01
```

Copy files from the $HOME directory to /tmp

```
asnc90$ cp $HOME/prog.f /tmp/asnusr01
```

*All working directories created on **/tmp** by the user should be removed after use. All files must be moved off of /tmp back to $HOME. Files in /tmp are deleted after 24 hours to maintain an adequate amount of space.*

Copy any input data files to /tmp

```
asnc90$ cp $HOME/mydata /tmp/asnusr01
```

Compile the program

```
asnc90$ f90 prog.f
```

Execute STOP in PROG

```
asnc90$ ./a.out
```

Copy the files on /tmp back to $HOME

```
asnc90$ cp * $HOME
```

Change directory to $HOME

```
asnc90$ cd
```

Remove directory created on /tmp

```
asnc90$ rm -r /tmp/asnusr01
```

Exit the Session

```
asnc90$ exit
```

Library Usage

Substantial performance improvements can be obtained by using the programs in the system libraries. These routines have been optimized for the CRAY C90, and in most cases, the routines are faster than user developed routines.

The chief guidelines are as follows:

- Remove obstacles to vectorization.
- Avoid memory bank conflicts.
- Maximize vector lengths.
- Keep memory references to a minimum.
- Bring as many C90 hardware resources to bear on the problem simultaneously and as continuously as possible.

Note: FORTRAN subprograms can be invoked from C, but the appropriate manual should be checked for differences in calling sequences, character strings, array element ordering and handling of pointers.

Multitasking

Multitasking is the parallel execution of parts of a program on two or more processors. This permits executing a program in less wall clock time than would be required for a non-multitasked program. The three modes of multitasking available on the C90 are:

Macrotasking (the original implementation)
: Multitasking applied at the program or subroutine level

Microtasking
: Multitasking applied at the loop level within a program block

Autotasking
: Automatic (compiler generated) multitasking of loop iterations.

See the CF77 and F90 manuals for additional information on microtasking.

Available Documentation

The following manuals are available for reference from the application analysts.

UNICOS User Commands Ready Reference Manual
SQ-2056

UNICOS User Commands Reference Manual
SR-2011

UNICOS Message Reference Manual
SR-2200

UNICOS Performance Utilities Reference Manual
SR-2040

UNICOS I/O Technical Note
SN-3075

UNICOS CDBX Symbolic Debugger Reference Manual
SR-2091

UNICOS X Window System Reference Manual
SR-2101

UPDATE Reference Manual
SR-0013 K

UNICOS CDBX Debugger User's Guide
SG-2094

UNICOS 6.1 System Administration
SG-2113

UNICOS File Formats and Special Files Reference Manual
SR-2014

UNICOS Primer
SG-2010

UNICOS Kernel Error Message Manual
SR-2015

UNICOS Support Tools Guide
SG-2016

Cray C Reference Manual
SR-2024

Cray Standard C Ready Reference
SQ-2076

Cray Standard C Programmer's Reference Manual
SR-2074

Fortran (CFT) Reference Manual
SR-0009

UNICOS CFT Reference Card
SQ-0022

UNICOS CFT77 Reference Card
SQ-0138

CF77 Ready Reference, publication
SQ-3770

CF77 Commands and Directives, publication
SR-3771

CF77 Fortran Language Reference Manual, publication
SR-3772

CF77 Optimization Guide, publication
SG-3773

I/O User's Guide, publication
SG-3075

Application Program I/O Guide, publication
SG-2168

Guide to Parallel Vectorization Applications, publication SG-2182

CF90 Ready Reference, publication SQ-3900

CF90 Commands and Directives Reference Manual, publication SR-3901

CF90 Fortran Language Reference Manual, publication
SR-3902

Introducing the Cray TotalView Debugger, publication IN-2502

Compiler Information File (CIF) Reference Manual, publication SR-2401

Math Library Reference Manual, publication SR-2138

Scientific Libraries Reference Manual, publication SR-2081

Application Programmer Library Reference Manual, publication SR-2165

Volume 4: UNICOS System Calls Reference Manual SR-2012

CRAY Multitasking Programmer's Manual SR-0222 G

Segment Loader (SEGLDR) and ld Reference Manual SR-0066

TCP/IP and OSI Network User's Guide SG-2009

Docview User's Guide SG-2109

Docview Writer's Guide SG-2118

Many of these are readily available on-line via 'docview' or through the CrayDoc (tm) on-line browser for Sun and SGI workstations (if CrayDoc is installed on a server on campus).

ASC Internet Connection

The Alabama Supercomputer Center and the Alabama Research and Education Network are connected to the Internet through connections to regional Internet providers. This section explains the various services available through the network, and the proper methods of accessing the ASC systems from the Internet.

Addressing

Bitnet addresses are of the form name@institution, for example:

smith@harvard

Internet addresses are of the form user@subnet.subnet.domain, for example:

smithg@eng.auburn.edu

The top level domains are:

edu	US educational institutions
com	US commercial institutions
gov	US government agencies
mil	US military sites
net	miscellaneous US networks
org	other US organizations

The primary ASC machines on the AREN have the following addresses:

C90	asnc90.asc.edu
asnmail	asnmail.asc.edu

Bitnet

To send mail to a Bitnet address, use the CUNY gateway:

Bitnet address

smith@auducvax

Gateway address

smith%auducvax@CUNYVM.CUNY.EDU

Other gateways addresses include:

uga.cc.uga.edu
um.cc.umcih.edu

CompuServe

To send mail to a CompuServe address, use the form:

12345.1234%compuserve.com@relay1.uu.net

where 12345.1234 is the CompuServe 12345,1234 account number.

To send mail from CompuServe to Internet, use internet:user@internet.address, for example:

internet:aubuser@eng.auburn.edu

Anonymous FTP

Many Internet sites permit access to archives and other directories via 'anonymous' FTP. FTP to the site and log on as 'anonymous'.

```
sun>ftp wuarchive.wustl.edu
userid:anonymous
send userid as password
password:asnuser@asnc90.asc.edu

(message will appear here...)

$
```

Anonymous FTP is a courtesy provided by the site. Please observe the rules:

- Logon, **get** or **put** files and then get off. Access ports may be limited.
- Access during the site's off hours (evening/night) is easier and appreciated.
- Observe local rules for posting and retrieving items.

Resources and information about FTP can be retrieved from the Internet site:

http://www.screen.com/understand/ftp.html

This site provides some of the latest ftp software, ftp guides and tutorials, lists of FTP sites, and advice on searching FTP sites.

http://hoohoo.ncsa.uiuc.edu/ftp

This web site has a "monster ftp site list".

World Wide Web (WWW) Sites

The WWW world consists of documents and links. It allows documents to be shared across multiple platforms in a standard format. Indexes are special documents which may be searched rather than read. The result of such a search is another ("virtual") document containing links to the documents found. A simple protocol ("HTTP ") is used to allow a browser program to request a keyword search by a remote information server.

To follow a link, a reader clicks with a mouse (or types in a number if he or she has no mouse). To search an index, a reader gives keywords or concepts. These are the only operations necessary to access the entire world of data.

The Internet holds a wealth of resources which can enhance knowledge, provide the most current information available in a field, and stimulate productivity; however, it can be time consuming and frustrating, except with browsers. The Mosaic browser, developed by the National Center for Supercomputing Applications (NCSA), is public domain software. Netscape is a commercial browser. There are many others available. The user's browser interprets the document allowing the publisher to focus on content. Documents may include text, audio, video, graphics, applications, etc. Browser documents can also embed links to related resources via hypertext.

Finding Information

There is a vast array of information and services available through the Internet. It is free for the taking -- if a user knows where to look.

Inter-Links
http://www.nova.edu/Inter-Links

Inter-Links was designed to help a user navigate the Internet and find useful resources by following a "hypertext" analogy. Embedded in information are "links" (highlighted text). Choosing a link takes the user to related information which also may have links.

Using the Internet Catalogs to Find Information

Yahoo

http://www.yahoo.com.

Yahoo is a hierarchical subject-oriented catalogue for the World Wide Web and Internet. It is a database of links to other sites. Yahoo does not provide any original content, only links to sites that already exist. To get listed, a user must set up a page on the World Wide Web and follow the "add url" instructions at the Yahoo site.

Galaxy

http://www.einet.net

The Galaxy is a directory service intended to make information on EINet and the Internet easy to find and access. Galaxy includes public information as well as commercial information and services provided by EINet customers and affiliates.

Galaxy is a prototype associated with the Manufacturing Automation and Design Engineering (MADE) program and will be under more or less constant change for the foreseeable future.

GNN-Global Network Navigator

http://www.gnn.com

GNN has many publications, from news and reviews to feature articles, advice, and commentary. Here's a brief description of some of what GNN offers:

Navigating the Net
http://www.gnn.com:80/gnn/wic/index.html

The Whole Internet Catalog contains a descriptive listing of the most useful Net resources and services, and it provides live links to those resources.

GNN Business Pages
http://www.gnn.com/gnn/bus/index.html

GNN Direct lets users browse product catalogs and order online. The GNN Business Pages tell users about companies that have established a presence on the Internet.

Yanoffs List

http://www.spectracom.com/islist

A voluntary listing of Internet sites by Scott Yanoff.

Public Accessible Mailing Lists

http://www.NeoSoft.com:80/internet/paml

This is a list of mailing lists available primarily through the Internet and the UUCP network. A mailing list differs from a newsgroup in that users do not receive anything unless they specifically request it. To be added to a mailing list, mail a note to the contact for that list, listed with each entry.

The WWW Virtual Library

http://www.w3.org/hypertext/DataSources/bySubject/Overview.html

This is a distributed subject catalog. Mail to maintainers of the specified subject or www-request@mail.w3.org to add pointers to this list.

Searches

There are new tools that attack the problem that has plagued many information spaces before the Web--how can a user find resources related to a topic or locate a specific resource? In ftp-space, there is Archie; in gopherspace, there is Veronica. For the Web, there is a variety of Web robots, wanderers, and spiders that have been crawling through the Web and collecting information.

Archie Servers

About Archie 3.0

http://pubweb.nexor.co.uk/public/archie/readme.html

The Archie information system is a network-based information tool offering proactive data retrieval and indexing for widely distributed collections of data.

Gateways to the Archie system have been built for a number of other information delivery systems, including the Wide Area Information System (WAIS), Gopher and the World Wide Web .

A list of Archie services (gateways) in the World Wide Web can be found at:

http://pubweb.nexor.co.uk/archie.html.

Veronica

Veronica is a resource-discovery system providing access to information resources held on most (99% +) of the world's gopher servers. In addition to native gopher data, Veronica includes references to many resources provided by other types of information servers, such as WWW servers, usenet archives, and telnet-accessible information services. Veronica queries are keyword-in-title searches. A simple query can be quite powerful because a large number of information servers are included in the index. Veronica is accessed through gopher client software.

World Wide Web Robots, Wanderers, and Spiders

World Wide Web robots, wanderers, and spiders are all programs that traverse the Web automatically.

Web Search Engines

In order to find a particular site or document on the Internet or to look for a resource list on a particular subject, use one of the many available on-line search engines. These engines allow users to search for information in many different ways - some search the titles or headers of documents, others search the documents themselves, and still others search other indexes or directories.

InfoSeek Search

http://guide.infoseek.com

InfoSeek is a comprehensive and accurate WWW search engine. Users can type their search in plain English or just enter key words and phrases.

Yahoo

http://www.yahoo.com

Yahoo is a hierarchical subject-oriented catalog for the World Wide Web and Internet. Yahoo is a database of links to other sites. Yahoo does not provide any original content.

Lycos

http://a2z.lycos.com

Lycos is a web spider search tool. Lycos lives up to its name--rather than catching its "prey" (URLs on a server) in massive single-server sweeps Lycos uses an innovative, probabilistic scheme to skip from sever to server in Webspace.

Webcrawler

http://webcrawler.com

The WebCrawler is a tool that searches by indexing and automatically navigating the Web. It creates an index by an incomplete breadth-first traversal, then relies on an automatic navigation mechanism to fill in the holes. Users can issue queries directly to the pre-computed index or to a search program that explores new documents in real time.

ALTAVISTA

http://altavista.digital.com

AltaVista gives users access to the largest Web index: 30 million pages found on 275,600 servers, and 4 million articles from 14,000 Usenet news groups. One of the most popular uses of AltaVista is searching for answers for homework or thesis work. Since AltaVista indexes all words, even common ones, users can find exact phrases such as "to be or not to be".

Excite

http://www.excite.com

Excite gives users access to searches, reviews of over 60,000 sites, a guide to the world, maps, travel resources, and top citiy listings.

Gopher Servers/Clients

Described as "the first Internet application that my mom and dad could use" by project leader Mark MacCahill, Gopher was developed at the University of Minnesota (where the little furry animal is the campus mascot). Its introduction was a true landmark in the evolution of the Internet, and Gopher remains one of the most popular tools for leaping full continents in a single mouse-click.

The description above and the latest information on gopher sites and gopher software can be found at the following web site:

http://www.screen.com/understand/gopher.html

Web Servers/Clients

The World Wide Web (WWW) is the universe of network-accessible information, an embodiment of human knowledge. It is an initiative started at CERN, now with many participants.

It has a body of software, and a set of protocols and conventions. WWW uses hypertext and multimedia techniques to make the web easy for anyone to roam, browse, and contribute to. The information provided in this section was retrieved from the WWW home page.

Clients and servers for many platforms exist and are under continual development. Much more information about all aspects of the web is available on-line.

The WWW model gets over the frustrating incompatibilities of data format between suppliers and reader by allowing negotiation of format between a smart browser and a smart server. This should provide a basis for extension into multimedia and allow those who share application standards to make full use of them across the web.

Getting Started

A user with nothing else but an Internet connection, should telnet to info.cern.ch (no userid or password). This very simple interface works with any terminal and gives a user access to anything on the web. It starts a user at a special beginner's entry point. Use it to find up-to-date information on the WWW client program a user needs to run on a computer, with details of how to get it. This is the crudest interface to the web -- do not judge the web by this. Just use it to find the best client for a given machine.

WWW Software

This is a list of products related to the WWW:

Client software

Programs to access the web directly from a computer

Server software

Programs for publishing information on the web

Gateways

Servers to make other existing information systems visible on the web

Mail Robot

A server which will return any web document by mail, given a request sent by mail. Also manages mailing lists.

Common Code Library

The CERN World-Wide Web Library of Common Code is a general code base that can be used to build clients and servers. It contains code for accessing HTTP, FTP, Gopher, News, WAIS, Telnet servers, and the local file system. Furthermore it provides modules for parsing, managing and presenting hypertext objects to the user and a wide spectra of generic programming utilities. The Library is the basis for many World-Wide Web applications and all the CERN WWW software is build on top of it.

Terminal based browsers

Line Mode Browser

This program gives WWW readership to anyone with a dumb terminal. A general purpose information retrieval tool.

"Lynx" full screen browser
http://www.cc.ukans.edu/lynx_help/Lynx_users_guide.html

This is a hypertext browser for vt100s using full screen, arrow keys, and highlighting.

Emacs w3-mode

WWW browse mode for emacs. Uses multiple fonts when used with Lemacs or Epoch.

Mosaic

http://www.ncsa.uiuc.edu/SDG/Software/WinMosaic/HomePage.html

NCSA Mosaic software is a distributed hypermedia system designed for information discovery and retrieval over the Internet. NCSA Mosaic provides a single interface to the variety of protocols, data formats, and information servers available throughout the Internet.

Netscape

http://home.netscape.com

With its immense popularity, Netscape Navigator has been recognized as the most powerful and innovative client software for easily exchanging information on the Internet. Netscape server products allow organizations to easily set up and maintain servers for conducting secure commerce over networks.

WWW Servers

A listing of World Wide Web servers may be found at the following site:

http://www.w3.org/hypertext/DataSources/WWW/Servers.html

A WWW server, like the ftp daemon , is a program which responds to an incoming TCP connection and provides a service to the caller. There are many varieties of WWW server software to accomodate different forms of data.

Useful Web Sites

Supercomputing

Cray Research
http://www.cray.com

Governmental References

Whole Internet Government Resources
http://gnn.com/wic/wics/govt.new.html

Some Alabama Sites

AlaWeb
http://alaweb.asc.edu

Alabama Research and Education Network
http://www.asc.edu

General Facts and Information

http://www.yahoo.com/Regional_Information/CIA_World_Factbook

http://www.yahoo.com/Regional_Information/States/Alabama

Comments and Corrections

Use this form to submit comments or corrections.

___ Comments enclosed

___ Corrections enclosed

Comments or corrections:

_____

_____

_____

_____

_____

Name: _____

Street: _____

City: _____

State: _____ Zip _____

Phone: _____

Email: _____

Mail to: **Alabama Supercomputer Center**
Attention: Operations - User Manual
686 Discovery Drive
Huntsville, AL 35806

Users may also email comments and/or corrections to:

helpdesk@asc.edu