Where observation is concerned, chance favours only the prepared mind.

— LOUIS PASTEUR

Appendix A

Networked computing environment within the chemical and materials engineering department

A.1 Introduction

In these notes you will be introduced to the *networked computing environment* within the department of chemical and materials engineering at the University of Alberta. You can choose to work with a variety of workstations such as the IBM RS/6000 which use the AIX operating system (a version of Unix), SUN workstations using SUNOS operating system, or the personal computers (Prospec 486) which use OS/2 and/or Win95 with DOS emulation. Hence forth the computers will be identified merely by the operating system used to run them viz. AIX, SunOS, OS/2 or WIN95 machines. Computers that are available for student use are distributed as follows in the Chemical and Mineral Engineering (CME) building.

Computers for teaching and research

AIX machines Nine IBM RS 6000 - model 220 in room CME 244 + One IBM RS 6000 - model 550 as server

SUNOS machines Three SUN workstations in room CME 244.

OS/2 machines Twenty Prospec 486 computers in room CME 244

WIN95 machines a variety of PC's in room CME 474.

Computers for graduate research

AIX machines A variety of IBM RS 6000 machines in room CME 475. These machines belong to individual researchers; but all staff and graduate students can have access provided the priority based on ownership is honored.

You are expected to be familiar with the basic concepts of operating systems, file systems, editors and structured programming and debugging principles. I will focus on the concepts that are unique to the networked environment and discuss briefly the software tools that are available for various tasks. This document is intended merely as an introduction to get you started and to identify the hardware and software resources that are available on the network. Once you are aware of the *art of the possible*, you are on your own to make the proper choice of machine, software tools and implement solutions strategies for solving a variety of problems. Advanced users should consult the original documentation on each software. Online help is available for most of the software.

In a heterogeneous, networked environment, such as the one we have, the range of services such as file space, printer, plotter capabilities, availability of specialized software *etc.* are distributed on a variety of machines, often matching the software needs with hardware capabilities. For example, graphics packages and word processors on PC's are adequate for most document processing needs. Powerful editors are available on all of the machines; you may already have your own favorite editor on one of these machines. Use it! Data processing, visualization, and simulation of flow and chemical processes are numerically intensive tasks and are best carried out on powerful workstations. MAPLE (a symbolic computation package) and APSEN (a process simulator) are made available only on one IBM RS 6000 machine for licensing reasons. Hence it is essential to develop the skills to navigate through the network, transfer data between computers and select the machine best suited for a task.

Workstations using AIX or SUNOS operating system support multi-tasking and multiple users. Multi-tasking implies that the computer can handle several tasks (or processes) at the same time using time sharing principles. In addition, AIX (and all other flavors of UNIX) can handle

multiple users by keeping track of user directories and accounting information. Thus, you need to get a *userid* and some disk space to store your own files in your *home directory*.

On the other hand OS/2 supports multi-tasking, but not multiple users. This implies that whenever you use an OS/2 machine, you can edit a program in one window and compose a letter in another (*i.e.*, multitasking feature), but you are responsible for keeping your personal files separately in a floppy. If you leave them in a hard drive, there is no guarantee that they will be available to you the next day! But you do not need a *userid* to use OS/2 machines.

All of the machines (AIX, SunOS and OS/2) provide a friendly Graphical User Interface (GUI) to interact with the operating system. The basic element of a GUI is a window and a desktop. AIX uses Motif window manager (a standard that is getting wide acceptance) and OS/2 uses the Presentation Manager (or PM). The figure A.1 illustrates the basic anatomy of a window and how to manipulate its size and location. A typical **AIX** window on IBM RS 6000 and an **OS/2** window on Prospec 486, provide a *shell* through which you can enter commands to the operating system. Well designed applications avoid command line based interaction with the computer; instead you have to merely (double) click on the icon to start an application. These icons, of course, have to be made available to the user and the concept of a *desktop* is used in both OS/2 and AIX to organize and present file systems and groups of applications to the user. Since AIX is a multiuser environment, each user can customize the desktop. This information is saved for each user and during subsequent logins you are presented with your own customized desktop. Under OS/2 there is only one standard desktop administered by the computer support staff - do not mess with it as it will make it difficult for subsequent users to access programs in a standard manner.

While illustrating a dialogue with the computer, following conventions are adopted throughout this document.

`courier font`	will indicate the prompt from the computer
bold font	will indicate a command that you should enter exactly as shown
italics font	will indicate a parameter like a file name, directory name *etc.* that should be *substituted* with the real name.

Unlike DOS, UNIX is case sensitive and most of the UNIX commands are in lower case.

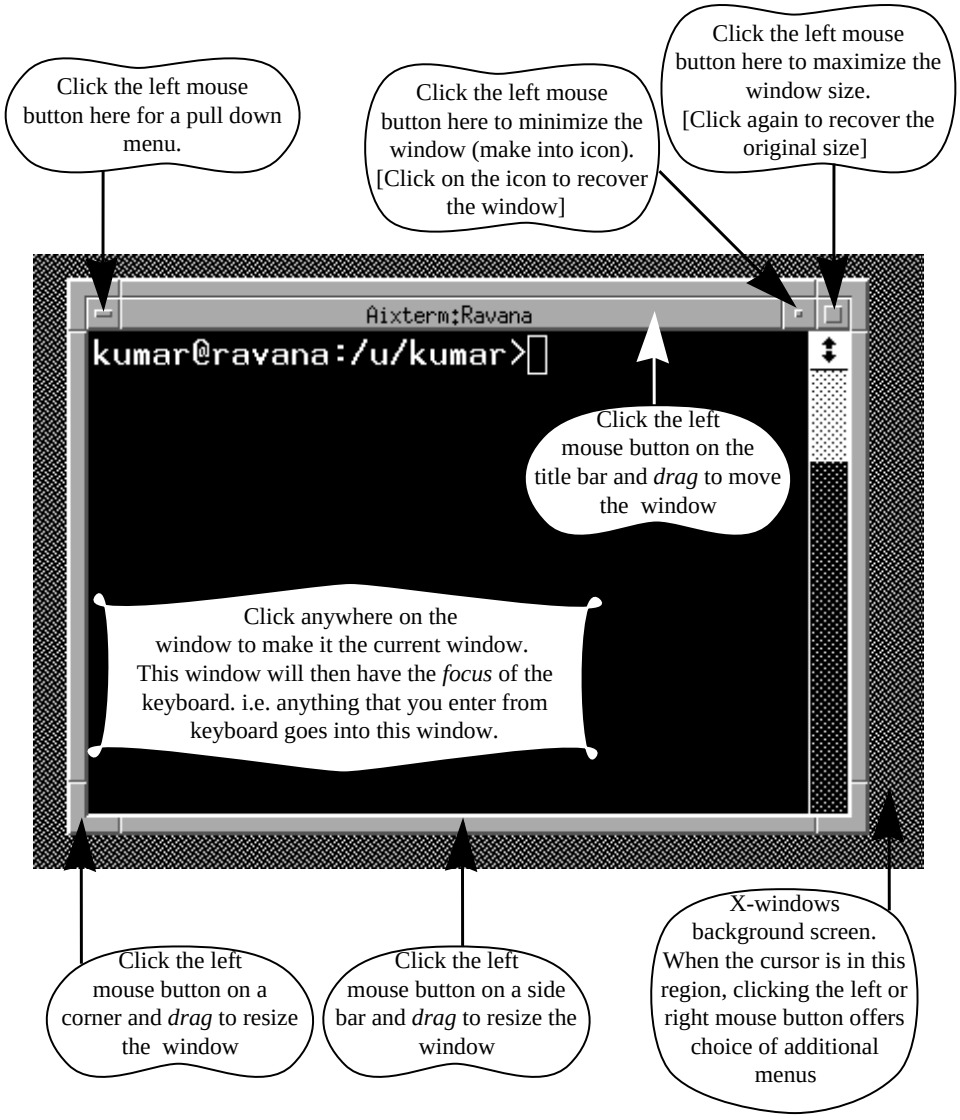


Figure A.1: Anatomy of a Motif window

A.2 Userid

All students get a *userid* on the General Purpose Unix servers (or GPU) maintained by the Computing and Networking Services (CNS). If you also purchase the *Netsurf 97* CD-ROM from CNS (for about $10), you get a powerful set of software for connecting your home computer to the INTERNET through the University. For more information about computing and network services see their Web site at `http:\\www.ualberta.ca\CNS`. Once you get the *userid* on the machines *gpu.srv.ualberta.ca* you can enable access to the machines maintained by the *department of chemical and materials engineering* by (a) signing on to *gpu.srv.ualberta.ca* and (b) running the following script:

```
[userid@gpu]> /afs/ualberta.ca/labs/cmel244/bin/register-244
```

You need to do this only once at the begining of the year. If you encounter any problems see one of the DACS center staff (Mr. Bob Barton or Mr. Jack Gibeau).

A.3 Overview of the network

Figure A.2 provides a conceptual frame work for introducing the network structure. A variety of workstations and personal computers are connected by ethernet. This local area network (LAN) is a subnet of the larger university wide ethernet network as well as the world wide INTERNET network. The underlying communication protocol is called TCP/IP (Transmission Control Protocol/Internet Protocol) and has been widely accepted as the standard.

In addition to providing some basic connectivity between machines, a network enables sharing of hardware and software resources as well as sharing of information and communication on a world wide basis. We will focus on the departmental subnet to illustrate various concepts which can then be easily extended to national and international level networks. The figure A.2 illustrates the logical dependencies for the purpose of sharing resources between various machines and not the physical connections. As a user, we need not concern ourselves with the network hardware connections. It is sufficient to realize that any machine on the network can address any other machine, much like telephone connections. This immediately requires that each machine on the INTERNET have a unique IP number and a host name. For exam-

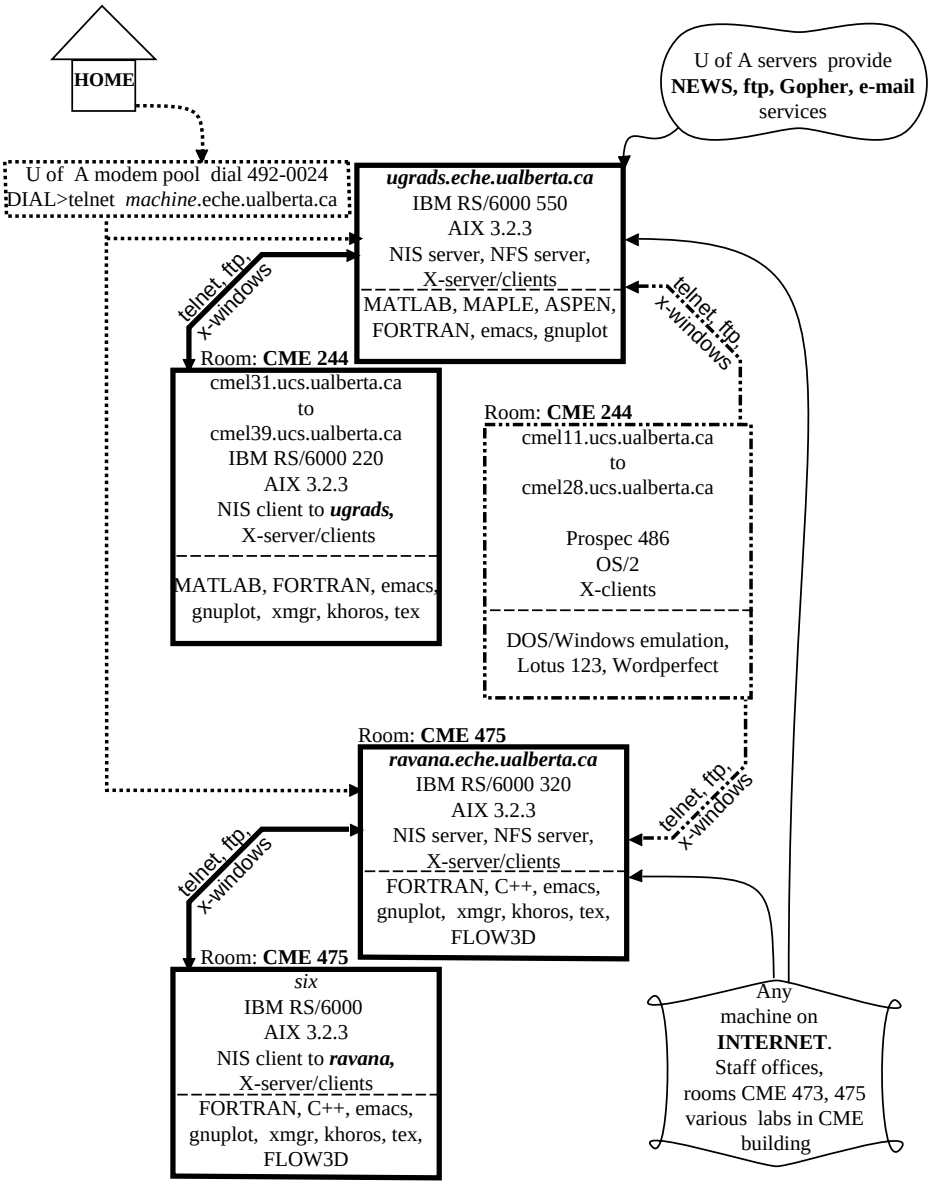


Figure A.2: Logical dependencies between various machines in the network

hostname	IP number	Machine type	Operating system
for undergraduate teaching and course related work			
ugrads.labs.ualberta.ca	129.128.44.50	IBM RS 6000 - 550	AIX 4.1
ugrads1.labs.ualberta.ca	129.128.44.51	Sun Sparc Ultra2	SunOS 5.5.1
ugrads2.labs.ualberta.ca	129.128.44.52	Sun Sparc Ultra2	SunOS 5.5.1
cmel31 .ucs.ualberta.ca	129.128.44.31	IBM RS 6000 - 220	AIX 4.1
. . .	. . .	. . .	. . .
cmel39 .ucs.ualberta.ca	129.128.44.39	IBM RS 6000 - 220	AIX 4.1
cmel11 .ucs.ualberta.ca	129.128.44.11	Prospec 486	OS 2
. . .	. . .	. . .	. . .
cmel28 .ucs.ualberta.ca	129.128.44.29	Prospec 486	OS 2
for graduate research and faculty use			
ravana.eche.ualberta.ca	129.128.56.102	IBM RS 6000 - 320	AIX 4.1
prancer.eche.ualberta.ca	129.128.56.23	IBM RS 6000 - 320	AIX 4.1
comet.eche.ualberta.ca	129.128.56.19	IBM RS 6000 - 375	AIX 4.1
dancer.eche.ualberta.ca	129.128.56.11	IBM RS 6000 - 375	AIX 4.1
cupid.eche.ualberta.ca	129.128.56.82	IBM RS 6000 - 355	AIX 4.1
dina.eche.ualberta.ca	129.128.56.70	IBM RS 6000 - 320	AIX 4.1
jhm2.eche.ualberta.ca	129.128.56.74	IBM RS 6000 - 375	AIX 4.1
chopin.eche.ualberta.ca	129.128.56.11	IBM RS 6000 - 320	AIX 4.1
poincare.eche.ualberta.ca	129.128.56.15	IBM RS 6000 - ???	AIX 4.1
handel.eche.ualberta.ca	129.128.56.54	IBM RS 6000 - 3AT	AIX 4.1
brahms.eche.ualberta.ca	129.128.56.13	IBM RS 6000 - 320	AIX 4.1

Table A.1: List of machines in the chemical and materials engineering network

ple the server within the department of chemical and materials engineering for undergraduate student use has the IP number *129.128.44.50* and the host name *ugrads.labs.ualberta.ca*. The part of the IP number *"129.128.56."* represents the department of chemical and materials engineering subnet and we can connect up to 256 computers to this subnet. Similarly the last part of the host name, viz. *eche.ualberta.ca*, also called *domain name* represents the chemical and materials engineering domain. Different machines on this domain will have different names like, *prancer.eche.ualberta.ca, ravana.eche.ualberta.ca*. Table A.1 lists computers that are generally available for chemical and materials engineering students and staff. To determine your eligibility to have access to any of these resources see one of the DACS center staff. Typically undergraduate students can expect to have access to *ugrads.eche.ualberta.ca* and the associated machines, while graduate students and staff will have

access to *ravana.eche.ualberta.ca* and its associated machines. You can signon to these machines through a variety of means including remote connections from home. These procedures will be outlined later in this document.

A.4 The client-server model

In a networked, distributed computing environment, the client-server model serves a very useful role in both providing a variety of services on the network and in sharing the limited resources with maximum flexibility. From a users point of view understanding the client-server concept helps in (i) navigating through the network and using the resources efficiently and (ii) diagnosing likely sources of problems when things do not work as anticipated.

In simple terms *a server provides a service to a client authorized to request such a service*. Typically, both the server and the client are programs that communicate over the network. The server is always running, listening for requests over the network. Such programs that are running all the time are also called *daemons*. When the client program initiates a request, the server program checks for authentication and provides the service. Some of the common services are explained below.

A.4.1 File servers

Let us take an example of Network File System, often called NFS. It is a set of protocols developed by Sun Microsystems and it has been accepted widely as a standard. It uses TCP/IP for communication, but provides a higher level function in terms of sharing files between various machines in a heterogeneous environment.

In figure A.3, the computer *chopin.eche.ualberta.ca* has a local file system called */usr/local* which contains a variety of very useful programs like *emacs, gnuplot, TeX, xmgr, pine* etc. This file system is quite large with about 450 Megabytes of programs and data that can be used by other similar machines. So this file system is "*exported*" to group of machines within the network. Thus *chopin.eche.ualberta.ca* acts as the NFS server to this group of machines. The client machines then "*mount*" this file system and make it appear as a local file system. In this way the user sees the same file structure on each of the machine and the administrator has to install and maintain only one copy of the software.

In a similar fashion, the authorization to signon to a group of machines (*i.e.*, user ids and passwords) is also maintained in a central loca-

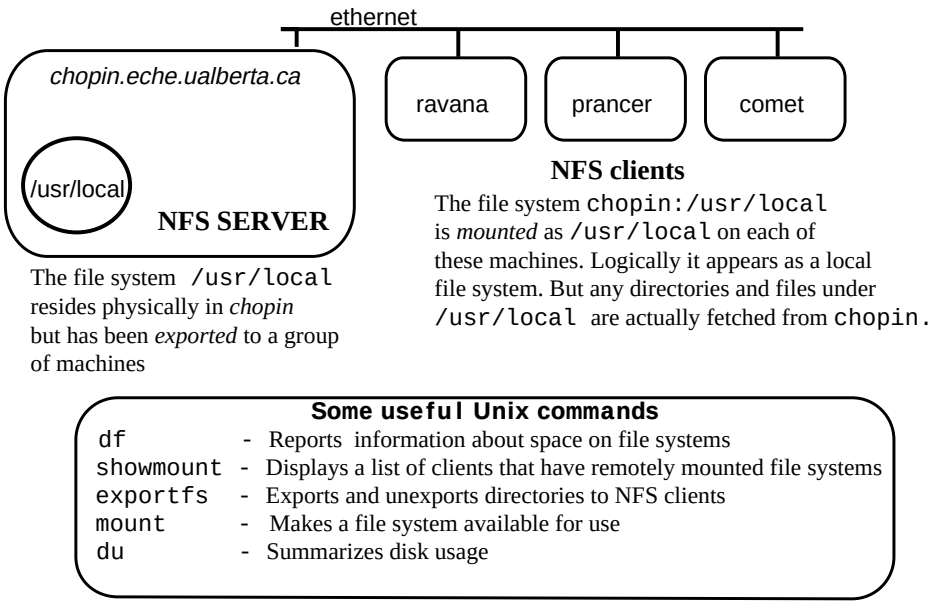


Figure A.3: Network File System server and clients

tion and served by the NIS (Network Information Service) server machine. Figure A.4 shows a such a setup.

The Andrew File System or AFS

The University of Alberta supports sharing of the same file space from a variety of machines on campus. There is a single *home directory* for each *userid* from any UNIX machine on campus. CNS provides limited amount of disk space for each user (about 10Meg). In addition the department of chemical and materials engineering provides additional space to undergraduate students under a directory called *labdisk* and for graduate students under a directory called *chemeng*. The relevant directories will be found as a subdirectory under the *home directory* of each user. To learn more about AFS go to the web site http://www.ualberta.ca/HELP/afs/afs1.html.

A.4.2 License servers

The client-server model is also used to manage software licenses. Due to financial constraints, unrestricted licenses are rarely purchased. Typ-

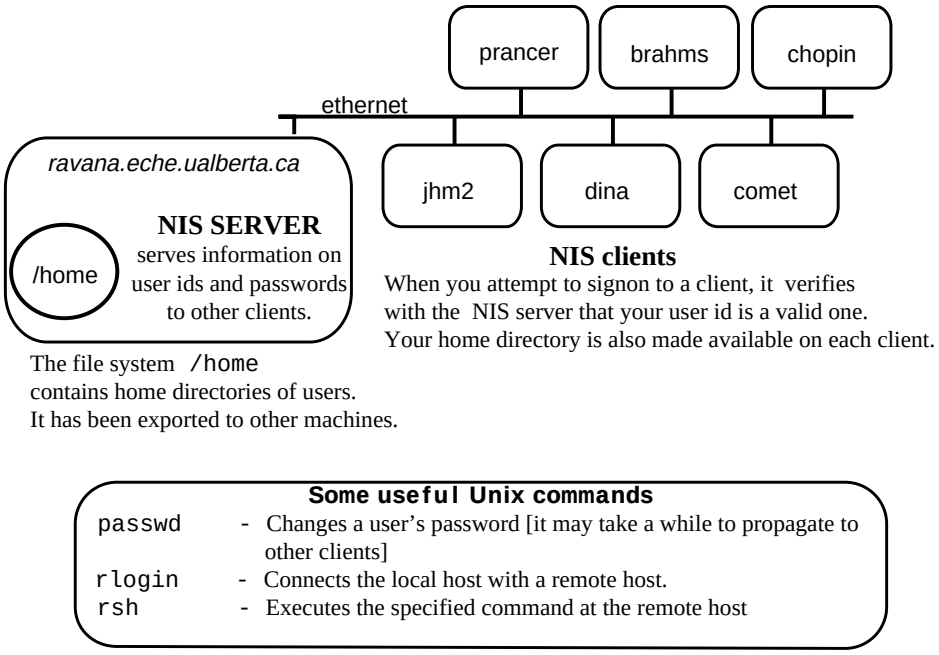


Figure A.4: Network Information Service

ically each software is licensed to run only on a specific group of machines. We also rely heavily on a large collection of freely available software. Two of the common licensing arrangements are *floating network license* and *node locked license*. The former allows a fixed number of concurrent users of a software on any machine in the network, while the later allows any number of users on a single machine.

Typically a *license server* (which is a small program) runs on one of the machines in the network as a *daemon*. Let us use FLOW3D and ASPEN PLUS as examples. FLOW3D is a commercial software package for use in fluid dynamics simulations and ASPEN PLUS is a very powerful steady state process flow sheet simulator. For managing FLOW3D, the license server (two programs named *lmgrd* and *CFDSd*) runs all the time on *bluejay.ucs.ualberta.ca*. Any time FLOW3D is started from any of the clients within the network, the client contacts the server to check whether it has the permission to start a new copy of the program. The server can either grant the permission or deny it because that particular client is not allowed to run that software or all of the available licenses are in use at that time. For FLOW3D we have license for 100 concurrent users.

For ASPEN PLUS, the department of chemical and materials engineering has a multiuser, node locked license. This implies that any number of users can use the program at the same time, but the software can run on only one machine. In our case ASPEN PLUS runs on *ugrads1* or *ugrads2*.

For MATLAB the university has a site license for up to 150 concurrent users. The most recent version of MATLAB 5.0 is available on *ugrads1* or *ugrads2*.

Table A.2 lists all of the software available on the undergraduate and graduate research network and the licensing status of each software.

A.4.3 X servers

X-windows is a sophisticated communication/window management software developed at MIT. It uses TCP/IP for communication. It allows you to interact with the local workstation through windows, menus, dialog boxes, icons etc., in an intuitive way and minimizes the need for interaction through command lines. One can view this as the next logical step in the historical progression of the way humans have interacted with computers. Originally it was accomplished through batch processing, followed by interactive computing using commands which is being replaced currently by more intuitive interaction through a Graphical User Interface (GUI) using menus and icons. One can expect this to be followed by voice level interaction, leading ultimately to interaction with

Software	ugrad-net	grad-net	Windows	Comments
ASPEN PLUS	ugrads1		X	process simulator floating, multiuser license
MAPLE	num.srv		X	symbolic computation package floating, multiuser license
MATLAB	ugrads1 cmel31 - cmel39	prancer comet	X	powerful linear algebra package with numerous tool boxes floating, multiuser license
FORTRAN	all AIX machines			IBM HESC license agreement
xde	all AIX machines		X	powerful debugger IBM HESC license agreement
emacs	all AIX machines			a powerful editor GNU public license
gnuplot	all AIX machines			a simple 2-D graphics package GNU public license
tgif	all AIX machines		X	a powerful drawing package free to use license
pine	all AIX machines			a standard e-mail program free to use license
tin	all AIX machines			a standard news reader free to use license
TeX	all AIX machines			a powerful typesetter free to use license
123	OS/2		Windows 3.1	spread sheet
Freelance	OS/2		Windows 3.1	Graphics
Wordperfect	OS/2		Windows 3.1	word processor
HYSIM	OS/2		DOS	process simulator

Table A.2: List of software on the chemical and materials engineering network

machines using natural language!

While Microsoft (Windows, Windows NT), Apple (MacIntosh), NeXT (NeXTSTEP), IBM (OS2-PM) and others have their own Graphical User Interface to interact with their operating system and application programs, X-Windows goes one step further, in making the GUI software *independent* of the underlying hardware and operating system. How this is accomplished is of interest only to programmers. From a users point of view we should understand the concepts and be able to reap the benefits of these features. In particular, X-windows makes the following possible:

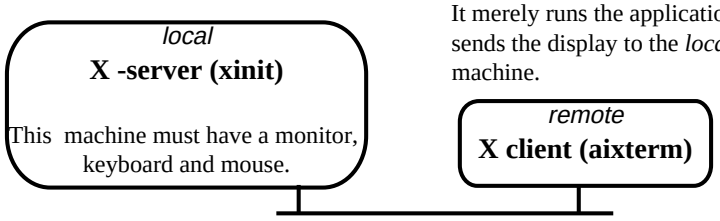
- Signon from a *local* machine to a *remote* machine. The two machines can be either of the same make or completely different, *e.g.*, a 486 running under OS/2 can be the *local* machine and IBM RS 6000 can be the *remote* machine.
- Run an application software on the *remote* machine *e.g.*, MATLAB, ASPEN PLUS, FLOW3D *etc.* , but have the X-windows display routed to your *local* machine. In other words you can interact with a remote computer, located as far away as in another building or another continent, with full window capabilities! (One needs a high speed link to interact with a computer in another continent, of course)

A conceptual description of the client-server interaction and the steps required to establish this interaction between a *local* and a *remote* computer using X-windows are illustrated in figure A.5. Note that under AIX the server program is called `xinit` and the client program is called `aixterm`. Under SunOS, the client program is called `xterm`.

A.4.4 News server

The INTERNET network provides a news service. Discussions and exchange of ideas take place through organized and moderated news groups. It is more like a conference on thousands of topics. You can choose to participate in topics of your choice. The topics are as wide ranging as the ones on classical music (rec.music.classical), MATLAB (comp.soft-sys.matlab), the AIX operating system (comp.unix.aix), dynamical systems theory (comp.theory.dynamic-sys) etc. The server is maintained and administered by Computing and Network Services. The client program is called a news reader and several versions of news readers exist. On the AIX machines, the news reader can be invoked with the command

NOTE: The string `user@machine:dir` is a prompt from the computer



Note: The client machine need not even have a monitor attached to it. It merely runs the applications and sends the display to the *local* machine.

Command sequence	What they do
1. **login**	signon to the *local* machine with user id and password
2. **xinit**	start the X-server
3. **xhost** *remote*	permit *remote* machine to send display to *local*
4. **rlogin** *remote*	signon to the *remote* machine from *local*
5. **aixterm -display** *local:0 &*	start X-client on *remote* and redirect the display to *local.*

NOTE: Substitute the actual name of the machine for *local* and *remote.* Use fully qualified *domain names* if the machines are on different subnet.

Some useful Unix commands

`xinit` - starts the X-server on the *local* machine
`xhost` - permit *remote* host to send display to *local* machine
`aixterm` - start a X-terminal (or X-client) and send the display to the machine *local.*

Some useful Unix files to explore

If these files are found in your home directory, they are used in controling **rlogin** between machines and in customizing your X-window environment.

`.rhosts` - controls **rlogin** between machines
`.xinitrc` - customizes the X-server.
`.Xdefaults` - customizes the Window appearance and functionality
`.mwmrc` - customizes the Motif window manager

Figure A.5: X-Windows client-server model

```
user@machine:dir> tin
```

This program maintains a local database of the news groups that you have subscribed. It fetches the latest articles from the server and allows you to read and, optionally respond to articles. If you post a response, it goes to the whole internet community. This service is a privilege and should be used responsibly. Any abuse of the system will result in denial of access to the entire network services.

A.4.5 Gopher service

While the News service allows two-way communication and thus active participation, Gopher is a worldwide one-way information retrieval service. A number of Universities and organizations participate in providing this service. The University of Alberta is one of the participants. The Gopher server is maintained by Computing and Network Services on one of their workstations. Any client on the campus ethernet (or even from a home computer with a modem) can connect to this server and browse through a wealth of information on such topics as

```
1.  What is CWIS?/
2.  What's New on CWIS?/
3.  Student Information and Services/
4.  Libraries on the Network/
5.  University Faculties and Departments/
6. Administrative Policies and Procedures (MAPPS)/
7. Campus Phone Directory <CSO>
8. Computing Resources/
9. International CWIS Systems/
10. Magazines and Publications/
11. University Services Directory/
12. Weather Report.
```

The `Student Information and Services` includes job posting, and the `International CWIS Systems` gives access to other Gopher services around the world. There are also nodes that link you to the Library of Congress and other university libraries. To connect to Gopher from any computer on campus with telnet feature enter,

```
user@machine:dir> telnet gopher.srv.ualberta.ca
```

Use as your userid *gopherc*; no password is required. When you connect

to the campus network from home through a modem by dialing 492-0024, *CWIS* will be provided as a menu option and selecting it will connect you to the Gopher server. When you are logged on to *ugrads.ucs.ualberta.ca*, you can also access *gopher* by entering

```
user@machine:dir> gopher
```

When you are logged on to *gpu.srv.ualberta.ca*, you can also access *gopher* or *world wide web* by entering

```
user@machine:dir> lynx
```

and selecting the appropriate menu.

A.4.6 World wide web service

This is a high end version of information sharing tool (like gopher) using windows and hypertext links established between computers on the internet on a global level. It also operates on a client-server model. *Netscape* is available on all of the AIX machines. Anyone with some information to share can operate a server on one of their workstations and there are thousands of such servers on the INTERNET. To access information on WWW, start the client on your local workstation by entering the following after you have started X-windows.

```
user@machine:dir> netscape
```

Your client should automatically know where the nearest server is located. Typically there will be home page from where you can begin your exploration. Have fun! Watch the time you spend on this one!! It is a time sink!!!

A.5 Communication

A.5.1 Software tools

There are many communication software packages available to connect from one computer to another. The following are available for use on the Chemical And Materials Engineering network.

kermit Preferred for communication from home computer to Chemical And Materials Engineering network via a modem. It allows

for VT100 emulation (*i.e.*, full screen use of editors like *emacs*, news reader *tin*, e-mail program *pine* and *gopher*) and tektronix emulation (*i.e.*, plotting programs like *gnuplot* can display graphs on your home computer in color!) and file transfer between home computer and the university computer. You can get a copy of MS-KERMIT Ver 3.10 for your PC from DACS center staff. It can be distributed freely without any licensing problems. Additional details can be found in Gianone (1991).

telnet Preferred for communication between UNIX machines or from OS/2 to UNIX machines over ethernet. It supports VT100 emulation, but has no graphic or file transfer support. Only command line interaction with the remote machine is possible.

ftp Preferred for file transfer over ethernet. Both AIX and OS/2 provide support for **ftp**.

tn3270 Preferred for ethernet communication from DOS based PC's (in room CME 473, staff offices *etc.*) to workstations. It provides VT100 and tektronix emulation as well as **ftp** support.

aixterm Preferred for remote login between AIX machines or from OS/2 machines to AIX machines. It supports full GUI features with the remote machine.

A.5.2 Connection to the AIX machines from OS/2 machines

If you are using a 486 PC under OS/2 in room CME 244, the following steps will establish a X-windows connection to a remote IBM RS 6000 running AIX.

- Make sure that the X-server is running under OS/2. This server is called presentation manager X or **PMX server** and, under normal operating conditions, it should be active and running. To verify that it is running, look for the PMX icon on the screen. If you do not find one, the key sequence **ctrl Esc** will give a list of windows. Check if PMX is listed. If you do not find it, you can start a X-server as follows:
  - ▷ Open a **OS/2 Full Screen Window** by double clicking on the appropriate icon.
  - ▷ To start the PMX server, enter

    `[C:\]` **xinit**

    If PMX server is already running, you will be informed of that fact.

- Open a **OS/2 Full Screen Window**, if one is not open already.
- **Telnet** to one of the IBM RS 6000 machines - *e.g.*, *cmel34.ucs.ualberta.ca*

  `[C:\]` **telnet cmel*34*.ucs.ualberta.ca**

  Remember that you can choose any one of the 9 AIX machines in the network. You will see the same home directory and other software resources from each of the nine machines. Hence choose a random number between 31 and 39, so that no one machine is over loaded all the time.
- Signon to the *remote* AIX machine.

  `login:` *userid*
  `Password:` *password*

- Start the *aixterm* client on the *remote* machine, redirecting the display to the *local* machine by entering

  `user@machine:dir>` **open-Xpc** *nn*

  where *nn* is the local station number of the Prospec 486 running OS/2. This effectively tells the AIX machine (in this example cmel34) to open a X-Window and display the window on the local workstation (i.e. cmel*nn*). Alternatively, you can enter,

  `user@machine:dir>` nohup aixterm -display cmel*nn*.ucs:0 &

  Observe that the computer prompt `user@machine:dir>` identifies your userid, the machine name and the present directory.

  NOTE: open-Xpc is nothing but a script that executes the above command for you!
- Exit from the AIX machine and from the OS/2 Full Screen Window. i.e. exit twice!

  `user@machine:dir>` **exit**
  `[C:\]` **exit**

- After some delay, an AIX window will open on your local 486 station! Now you are connected to the AIX machine and typically you will be in a shell called “ksh”. You can start any application like MATLAB, xmgr *etc.* by simply entering the name of the program *e.g.*,

```
user@machine:dir> matlab
```

If an application, such as ASPEN PLUS, uses X-windows effectively, it might open up other windows, dialogue boxes *etc.* and also provide online help windows. To start ASPEN PLUS enter

```
user@machine:dir> mmg
```

which is the graphical version of the model manager.

- Before you leave the OS/2 station, either logout from the *remote* machine with the logout command as follows

  ```
  user@machine:dir> logout
  ```

  You can also use ctrl-D key to logout

  or close the *aixterm* window.
  Caution: If you leave the OS/2 station without completing this last step, the next person using that OS/2 station will have access to your account on the AIX machine!

A.5.3 Connection to the AIX machines from a home computer

Use of Kermit is recommended. You also need a modem. Kermit has an initialization file which can be setup in such a way that it automatically dials the telephone number. The available phone numbers to connect to the University of Alberta network server are : 492-4811 or 492-0024 (2400 Baud) , 492-0096 (9600 Baud) or 492-3214 (high speed modem). Sample initialization files can be obtained from DACS center staff. You start the kermit program by entering

```
C:> kermit -finitfile.ini
```

where *initfile.ini* is the name of the initialization file. By default it is *mskermit.ini*. If no file by name *mskermit.ini* exists in your directory and you do not specify explicitly the name of an initialization file as shown above with the **-f** flag, then the Kermit program will start on the PC, but you will be left with the prompt

```
MS-KERMIT>
```

At this stage you can ask for additional help on kermit by entering the kermit command **help** or **?**. But the connection to the University network must be done manually with the following commands

```
MSKERMIT>set speed 2400          ;this sets the modem speed
MSKERMIT>set port com2           ;this sets the port
MSKERMIT>OUTPUT ATDT4924811 ;this dials the number
MSKERMIT>connect                 ;this connects you to the Univ. network
```

These steps can be automated by putting the above kermit commands in the initialization file. Note that anything following a semi-colon is treated as a comment by MS-KERMIT. Whichever procedure you use, you should get the following prompt if everything has gone well up to this stage.

```
DIAL>
```

At this stage enter

`telnet,` *129.128.44.50*

or

`telnet,` *ugrads.eche.ualberta.ca*

to get connected to the AIX machine or any other valid internet number (or name) for which you have a valid *userid*.

A.5.4 Connection to the AIX machine from a DOS machine using tn3270

All the PCs in room CME 473 and those in staff offices that have ethernet cards are connected to INTERNET. The communication program **tn3270** is also available in each of the machine. On these machines you can start the connection by entering,

`[C:\]` **tn3270** *remote.machine*

Substitute the actual *machine* name.

You should be connected directly to the *remote* AIX machine. This communication program is quite powerful. It provides full VT 100 emulation and hence allows full screen use of editors (emacs, vi), news reader (tin) *etc.* It also supports tektronix emulation and has facilities for capturing graphics screen images as Postscript files on the local PC. Use the `Alt-H` key sequence to get brief online help.

A.5.5 File transfer with Kermit

A client-server concept is used once again to transfer files between the *remote* IBM RS 6000 and the *local* home computer. The MS-KERMIT on your PC is the client. You must start a server on the IBM RS 6000, called the C-KERMIT. Once you are logged in to the IBM RS 6000, enter

```
user@machine:dir> kermit
```

to start the C-KERMIT program. The prompt string should change to

```
C-Kermit>
```

Enter,

```
C-Kermit> server
```

This command is for C-Kermit

to put C-KERMIT in server mode. Then escape back to the *local* MS-KERMIT (Typically **ctrl-[C** is the escape sequence. The prompt should be,

```
MS-Kermit>
```

Any command you enter now is acted up on by the *local* PC. To fetch a file from the AIX machine (in your home directory, of course) to local PC, enter,

```
MS-Kermit> get filename
```

This command is for MS-Kermit

To send a file from your local PC to the AIX machine (in your home directory, of course) enter,

```
MS-Kermit> send filename
```

Once the file transfer is completed, enter

```
MS-Kermit> fin
```

Although this command is for MS-Kermit, it tells C-Kermit to terminate server mode. The next command, C, re-establishes direct connection to AIX

to signal the server that file transfer session is finished. Then enter,

```
MS-Kermit> C
```

to re-connect to the AIX machine. Note that the C-Kermit is still running on the AIX; it has only terminated its server mode. Finally enter

FTP subcommand	its function
help	ask for help on FTP subcommands
get *remotefname localfname*	get a file from the remote machine
mget *remote-pattern*	get multiple files matching the pattern
put *localfname remotefname*	put a file on the remote machine
mput *local-pattern*	send multiple files matching the pattern
bin	to enable file transfer in binary mode
ascii	to enable file transfer in ascii mode (this is the default)
ls	list files in the current directory on the remote machine
dir	also list files in the current directory on the remote machine
cd	change the directory on the *remote* machine
lcd	change the directory on the *local* machine
pwd	display present working directory on remote machine
quit	to terminate the FTP session

Table A.3: List of ftp commands

```
C-Kermit> quit
```

The `quit` command is acted on by C-Kermit

to terminate the C-KERMIT program on the AIX machine and to return to the shell level.

A.5.6 File transfer with ftp

The file transfer program called **ftp** allows transfer of files between machines on ethernet. This is also implemented on the client-server model. From a *local* machine connect to the *remote* machine as follows:

```
user@machine:dir> ftp 129.128.44.50
```

On the *remote* machine the ftp daemon called *ftpd* acts as the server. You will be prompted for *userid* and *password*. Once the connection is established you can use the commands shown in Table A.3 to transfer file.

The FTP procedure summarized above is essentially the same on most of the machines. Be bold and try them out and observe how fast the file transfer is compared to KERMIT through a serial line.

A.5.7 File transfer from AIX to OS/2 machines

ftp feature is also available under OS/2 machine. If you want to make a copy of your personal files on the AIX machine to a floppy disk, do the following.

- Make sure that you have a writable, formatted floppy disk in drive `A:` of an OS/2 machine.
- Open a **OS/2 Full Screen Window**, if one is not open already.
- Enter,

  `[C:\]` **ftp** *remote.aix.machine*

  Substitute the actual *machine* name.

- Signon as usual.
- Change the local directory to `A:` using

  `[C:\]` **lcd A:**

- Transfer the file using,

  `[C:\]` **get** *remote.filename*

  Substitute the actual *file* name.

A.6 Operating systems

A.6.1 How to signon and logout

The first time you signon to an AIX machine you will be forced to change your password. Subsequent times, you will be given informational message regarding your previous singon time and date and any unsuccessful attempts to singon to your id. Change your passwords periodically. Make sure that you always logout before you leave your workstation.

The following commands summarize the syntax for loging in and out of AIX systems and for changing passwords.

Login procedure

`login:` *userid*
`password:` *password*

Substitute the actual *userid* and *password*.

Logout procedure

```
user@machine:dir> logout
```

You can also use **ctrl-D** key to logout

Changing password

```
user@machine:dir> passwd
```

Respond to the prompts for the old and new passwords

A.6.2 Customizing your environment - the .profile file

Once you login successfully to an AIX machine, you will always start the session in your home directory. There should be a file in your home directory named **.profile**. Every time you login, the contents of this file are executed once. Hence one can use this file to customize the working environment on an AIX machine with the help of this file. There is also a system wide profile file, which is used to control such things as the default search path for finding executable programs, controlling the prompt string, identifying the terminal type, and to define a number of environment variables that other application programs might need.

You can feel free to copy my version of this file in **/u/kumar/.profile** and adopt it to your needs. My version of this file enables the *command line editing* features - *i.e.*, all the commands that you enter during a session are stored in a buffer and you can scroll back and forth to retrieve previous commands with **Ctrl-p** for previous and **Ctrl-n** for next. After retrieving a previous command you can edit it with the cursor movement keys **Ctrl-b** for backward, **Ctrl-f** forward and **Ctrl-d** for deleting the current character. In effect it supports the same editing capabilities on the command line that the editor **emacs** supports for a file. More on **emacs** later. You can also set **alias** for the most frequently used commands. For example you can set

alias dir='ls -al'

so that when you enter **dir** you will get the directory listing. Another useful alias is

alias rm='rm -i'

which prompts you for confirmation before removing (deleting) files.

A.6.3 File management

If you are familiar with the DOS directory and file structure, it should be equally easy to work with the file systems of OS/2 and AIX. While DOS file

names are restricted to 12 characters, AIX allows very long file names. But the mechanisms for creating directories and navigating up and down the directory tree structure are essentially the same. Frequently used commands that relate to file management are tabulated below.

Task	UNIX	OS/2	DOS 5.0 or above
seek online help	**man** *command*	**help** *command*	**help** *command*
list the directory	**ls -al**	**dir**	**dir**
list contents of a file list one page at a time	**cat** *fname* **more** *fname*	**type** *fname* **more** *fname*	**type** *fname* **more** < *fname*
create a file	**touch** *fname*		
erase a file	**rm** *fname*	**erase** *fname*	**erase** *fname*
copy a file	**cp** *fn1 fn2*	**copy** *fn1 fn2*	**copy** *fn1 fn2*
append *fn1 to fn2*	**cat** *fn1 >> fn2*		**copy** *fn2+fn1*
rename a file	**mv** *fn1 fn2*	**rename** *fn1 fn2*	**rename** *fn1 fn2*
create a directory	**mkdir** *dirname*	**mkdir** *dirname*	**mkdir** *dirname*
change directory	**cd** *dirname*	**cd** *dirname*	**cd** *dirname*
delete an empty directory	**rmdir** *dirname*	**rmdir** *dirname*	**rmdir** *dirname*
present working directory	**pwd**	cd	**prompt** pg
disk usage summary	**du -s**		**chkdsk**
status of system	**ps -l**	**pstat**	

A.6.4 How to get online help

Each of these commands, particularly in UNIX, can take several optional parameters, flags etc. that further identify any specific features of the command that you want to enable. This command list is by no means complete. They are the more frequently used commands. On UNIX additional details on each of the command can be found with the command

`user@machine:dir>` **man** *command*

These are called manual pages or man pages for short. Try

`user@machine:dir>` **man man**

to get started! Although the man pages provide only limited help, they are always available from any type of terminal. Much more exhaustive online help using hypertext is available when you are connected to an AIX machine via X-windows. To access this enter

`user@machine:dir>` **info**

and follow the instructions. Extensive help based on hypertext is also

available on OS/2 machines.

A.7 Editors

The functions that one expects from a good editor have several common features. These can be broadly grouped into the following.

cursor movements	by characters, words, lines, blocks, etc.
delete text	by characters, words, lines, blocks, etc.
insert text	by characters, words, lines, blocks, etc.
move text	by characters, words, lines, blocks, etc.
copy text	by characters, words, lines, blocks, etc.
locate text	by strings perhaps with *regular expressions*
search & replace	*text1* by *text2*

A good editor must have a set of commands or keyboard sequences that invoke functions to carry out the above tasks. On AIX there are several editors. I prefer **emacs**, as I find it to be extremely powerful; it also has built in online help and hence you can learn at your own pace and grow with it. OS/2 also has a full screen visual editor. I am sure that you will have your own favorite editor. If you have one use it and ignore this section.

A.7.1 Emacs - *the ultimate in editors*

If you want to learn the ultimate in editors (my biased view of course!) try **emacs**. To start this editor in AIX use the command,

```
user@machine:dir> emacs
```

The anatomy of the emacs editor screen is shown in Figure A.6. This is also a full screen, visual editor. It works under both X-windows and VT100 emulation.

Buffers

Emacs uses the concept of a buffer to keep a temporary copy of the file that you are editing. You can edit any number of files at a time and each

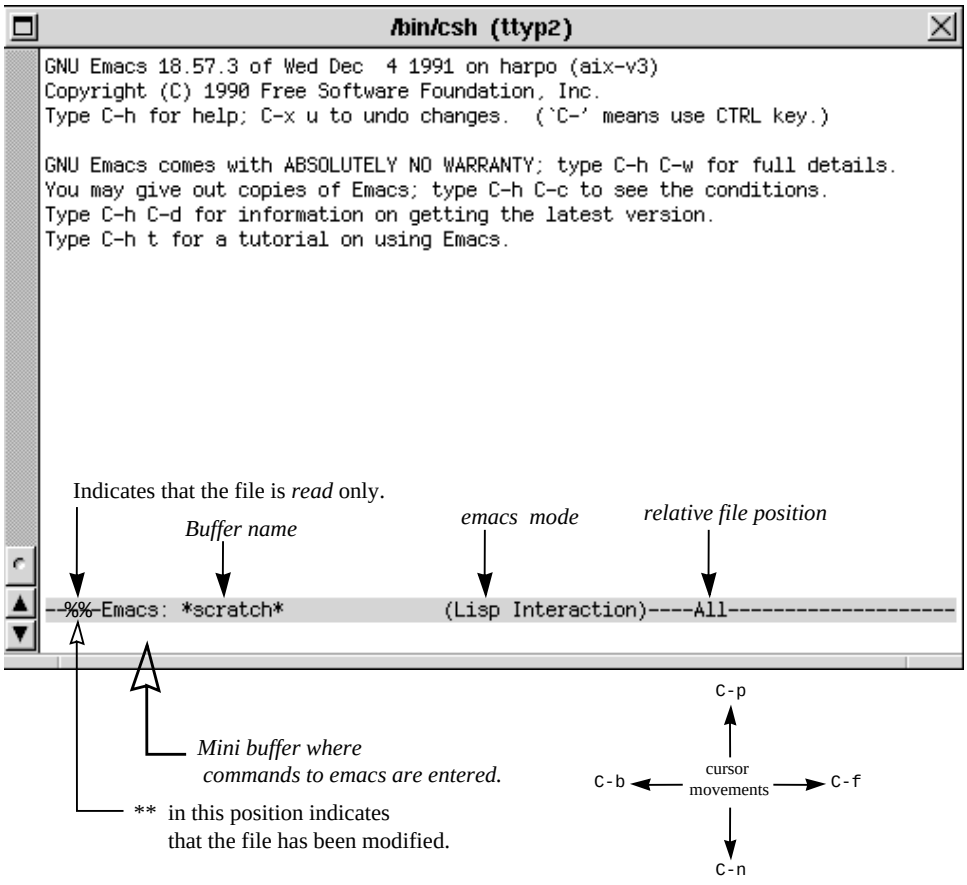


Figure A.6: Anatomy of an emacs window

file is kept in a separate buffer. There is a one line command buffer called *minibuffer* at the bottom of the screen and a text buffer at the top. A status line in reverse video separates the minibuffer from the text buffer. The key sequence **C-x b** allows you to cycle through the buffers.

Key sequences

Being the most powerful editor, emacs provides a large number of editing functions. These functions are accessed either by entering the function name in the command minibuffer or by directly entering a key sequence. This process of attaching key sequences to functions is called key bindings. Actually when you enter a key sequence like **C-f**, it invokes a function called `forward-char`. Some of the frequently used functions and their key bindings are listed in Table A.7.1. In these **C-x** means the keeping the control key pressed, enter the character x, while **M-x** implies a two-character key sequence with the **Esc** key as the first one followed by the character x. If you get into trouble or become confused with this editor at any time enter the key sequence **C-g C-g** to discontinue what you started. This means that while keeping the control key down, enter the letter **g** couple of times.

Command completion

If you find the use of key sequences difficult to remember,but would rather remember only the function names (and that too only vaguely!), then you can enter **M-x** followed by the first few letters of the function name and the *space-bar* key. A list of all possible functions beginning with those few letters will be displayed in a lower window. This process is called function completion. Try

M-x sa *space-bar*

You will see that the function name will be completed till `save-` and stop as an ambiguity exists at the point. Another *space-bar* will show all functions that begin with `save-` and you can continue to identify the one you want - *e.g.*, **b** *space-bar* will identify the function as `save-buffer`. Recall that this function is bound to the key **C-x C-s**.

This concept of completion is also used in selecting a file. For example, the key sequence **C-x C-f** will get a new file into a buffer for editing. Once you execute the key sequence, the current directory will be displayed in the minibuffer. Entering the first few characters of a file name followed by *space-bar* will complete the file name till the next point of ambiguity.

task	key sequence	function-name
ask for help	**C-h C-h**	help-for-help
ask for tutorial	**C-h t**	help-with-tutorial
ask for information	**C-h i**	info
to end emacs session	**C-x C-c**	save-bu□ers-kill-emacs
abort if you get into trouble	**C-g**	keyboard-quit
edit a new file in a new bu□er	**C-x C-f**	find-file
insert a new file at cursor position	**C-x i**	insert-file
switch to a di□erent bu□er	**C-x b**	switch-to-bu□er
save the current bu□er into the file	**C-x C-s**	save-bu□er
save the current bu□er as a new file	**C-x C-w**	write-file
move one character backward	**C-b**	backward-char
move one character forward	**C-f**	forward-char
move to previous line	**C-p**	previous-line
move to next line	**C-n**	next-line
delete the current character	**C-d**	delete-char
undo	**C-x u**	advertised-undo
move to beginning of line	**C-a**	beginning-of-line
move to end of line	**C-e**	end-of-line
open a new line for typing	**C-o**	open-line
kill from cursor to end of line	**C-k**	kill-line
yank it back (explore kill ring!)	**C-y**	yank
move to beginning of bu□er	**C-x [**	backward-page
move to end of bu□er	**C-x]**	forward-page
page forward to next screen	**C-v**	scroll-up
page backward to previous screen	**M-v**	scroll-down
redisplay current screen	**C-l**	recenter
incremental search forward	**C-s** *string*	isearch-forward
incremental search backward	**C-r**	isearch-backward
search & replace	**Mx %**	query-replace
to cancel search	**C-g C-g**	
start recording keystrokes	**C-x (**	start-kbd-macro
end recording keystrokes	**C-x)**	end-kbd-macro
replay/execute recorded keystrokes	**C-x e**	call-last-kbd-macro

NOTE: C-h means keeping the control key down, enter the key “h”.
M-v means press and release the Meta key or the Esc key, then press the key “v”.
C-x [means keeping the control key down, enter the key “x”, then press the key “[”

Table A.4: List of frequently used emacs functions and their key bindings

Emacs contains many more functions than listed in Table A.7.1 and not all functions have key bindings. You are encouraged to go through the online tutorial which can be invoked with the keystrokes **C-h t**. Online documentation is available via **C-h C-h i**. The key sequence **C-h C-h b** will show all the key bindings within emacs in a buffer called *Help* after splitting the screen into two windows. You can switch to the lower window with the key sequence **C-x o** which is the same as invoking the function *other-window*. Repeating the key sequence **C-x o** will cycle you through the various windows. To expand the current buffer into full screen use the key sequence **C-x 1**.

This is only a minimal introduction to emacs. This editor is not for the uninitiated user. You need patience to master this editor, but if you persist the rewards in terms of increased productivity are great. With this introduction you should be able to explore emacs deeper and deeper! If you are bored while using emacs, chat with the doctor - use **M-x doctor** - have fun.

A.8 Fortran compilers

Fortran compilers and X-window based debuggers are available on all of the AIX machines. The FORTRAN compiler on the AIX server is named **xlf**. To compile your code use the command

`user@machine:dir>` **xlf** *srcfile.f* **-O**

This is adequate for any self contained FORTRAN program. The **-O** option invokes optimization of the code. The executable code from the compiler will automatically be stored in a file named **a.out**. To execute your program use,

`user@machine:dir>` **a.out**

If you read data from unit 5 and write data to unit 6 within your FORTRAN program, these I/O will be redirected, by default, to your terminal. If you want to use files connected to these units, then you can use

`user@machine:dir>` **a.out** <*indata* >*outdata*

Another way to use these files is to OPEN then explicitly within your FORTRAN program.

If your program calls any library routines like the **nswc** math library, then you must specify the location of the library as follows in the com-

pilation step.

```
user@machine:dir> xlf srcfile.f -O -lnswc -L/usr/local/lib
```

Here **-l** parameter identifies the name of the file containing the library and the **-L** parameter identifies the directory where the library file resides. Note that the library file will be named **libnswc.a**. By convention, however, the prefix **lib** and the postfix **.a** need not be specified in the parameter **-l**.

Use the **man xlf** to find about other parameters the **xlf** compiler can accept.

A.9 Debuggers

A source level debugger allows one to step through the program, executing one line at a time, set break points at pre-selected lines, display values of variables and change values of variables. There is an excellent X-Windows based front end called **xde** to the debugger (**dbx**) on the RISC server that is quite easy to use. Find out more about it using **man xde**. If you want to use this debugger, however, you should access the computer through X-Windows.

In order to enable the debugging features you should compile the program with the **-g** option as follows.

```
user@machine:dir> xlf srcfile.f -g
```

The **a.out** file so produced will contain all the symbol table information which is needed by the debugger. To invoke the debugger use,

```
user@machine:dir> xde a.out
```

This will open up several windows, one of which will show the source listing. Other windows provide menus and buttons that you can use to set break points, begin execution, display variables *etc.* There is also online help explaining the various features of **xde**.

For the real computer hacks, a command line version of the debugger called **dbx** is available. You can use this to debug FORTRAN, C and PASCAL programs. Venturing into this is recommended only if you know the language C quite well. This can be accessed, however, without X-windows.

A.10 Application programs

There are a large number of application programs on the AIX machines. They are all located under the directory "/usr/local". The procedure for starting several useful applications is given in the next few sections. Online help or online documentation is often adequate to learn more about the software.

A.10.1 ASPEN PLUS

APSEN is a powerful steady state process flow sheet simulator. It is useful for carrying out rigorous, steady state mass and energy balance calculations. It has a fairly extensive thermodynamic data base. The X-window based front end, called the **model manager** allows you to define the problem interactively. The flow sheet can be constructed graphically by grabbing modules and connecting them up. Operating conditions can be defined interactively by filling out forms. The expert system interface ensures that all required input parameters are specified before the simulation is started. ASPEN PLUS is licensed to run only on *ugrads.eche.ualberta.ca*.

- Establish a X-window connection to *ugrads* from any OS/2 machine (follow section A.5.2) or from other AIX machines (see figure A.5).
- From the command line enter,

  `user@machine:dir>` **mmg**

  to start the model manager. Explore more on your own!

Advanced users of ASPEN PLUS can use the simulator directly from the command line using

`user@machine:dir>` **aspen**

The input file containing the commands to ASPEN PLUS must be prepared by the user using any standard editor. One can think of the Model Manager as a front end that enables you to build the command file for ASPEN PLUS in a GUI environment.

A.10.2 Xmgr

This is a powerful 2-D plotting and data analysis package. You can control every facet of the graph with this program. It runs only under X-

windows. This program is available on most of the AIX machines. To start the program

- Establish a X-window connection to *ugrads* from any OS/2 machine (follow section A.5.2) or from other AIX machines (see figure A.5).
- From the command line enter,

  `user@machine:dir>` **xmgr**

  to open up a main window. Explore more on your own!

A.10.3 TEX

TEX is a powerful typesetting package that is widely available on several platforms and it has very few licensing restrictions. This document, in fact, was typeset using TEX. It is available on most of the AIX machines. The authoritative document on TEX is by Knuth (Knuth, 1984) and on LATEX is by Lamport (Lamport, 1986). The steps for compiling, previewing and printing a TEX document are outlined here. See the above references on how to create a TEX document. To compile a TEX file use,

`user@machine:dir>` **tex** *file*.tex

To compile a LATEX file use,

`user@machine:dir>` **latex** *file*.tex

Either of these steps should produce a file named *file*.dvi, which is a device independent file. The contents of this "dvi" file can then be used to either preview the document on the screen or send it to a printer. To preview use,

`user@machine:dir>` **xdvi.sh** *file*.dvi

Note that on the AIX machine, you must be using X-windows to use the previewer. A program called `dvips` takes the "dvi" file and sends it to a postscript printer. Since printer configurations vary, see one of the support staff to find out the exact procedure for printing a document.

A.10.4 **pine** - the mail reader

Pine is the standard mail reader on our AIX network. To start it simply enter

```
user@machine:dir> pine
```

This program will work under VT100 emulation and offer full screen support. Online help can be accessed with the question mark, ?. Note that e-mail addresses are formed as *userid@machine.***eche.ualberta.ca**. Here *userid* is your signon name or id on the AIX machine, *machine* is the name of the machine. The rest of the e-mail address, **eche.ualberta.ca** refers to the chemical and materials engineering subnet *domain* name. If you have accounts on several machines it is recommended that you select and consistently use one machine as your primary e-mail system.

A.10.5 **tin** - the news reader

Tin is the standard news reader on our AIX network. To start it simply enter

```
user@machine:dir> tin
```

This program will work under VT100 emulation. Online help can be accessed with the "h" key.

A.11 Distributed Queueing System

The Distributed Queueing System (DQS) allows CPU intensive tasks to be executed on a machine that is relatively free within the graduate network, while at the same time allowing faster keyboard/terminal response to interactive users. Any task that takes longer that 5 min of CPU time should be submitted through the DQS. Otherwise the jobs will be terminated automatically. The commands that you should be familiar with are

- **qsub** - for submitting batch jobs.
- **qstat** - to monitor the status of jobs in the DQS
- **qmon** - under X-windows to monitor status of jobs.
- **qdel** - to delete your own (and only your) jobs.

For the experts, online documentation via man pages is available on each of the above commands. The following examples illustrate how to construct a *script* file that you would submit using **qsub**.

A.11.1 Example of a CPU intensive MATLAB job

Construct a *script file* named, say `test.bat`, containing the following lines.

```
#!/bin/csh
# make the current directory the CWD
#$ -cwd
# lets put STDOUT/STDERR in the file "gaga"
#$ -eo gaga
# i'd like to know when she fires
#$ -mu user@address.ualberta.ca
#$ -mb
# and when she finishes
#$ -me
matlab >out  « 'eof'          #the shell starts matlab; output goes to "out"
secant('ass3a',[0,1],eps,1)    %matlab acts on this line and runs secant
fzero('ass3a',0.5,eps,1)       %matlab acts on this line and runs fzero
quit                           %matlab acts on this line and quits
eof
ps -ael                        #back in the shell; output goes to "gaga"
ls -al                         #The shell acts on this too!
```

In the above file any line beginning with "#$" is a command for the queueing system and any line that begins with "#" is a comment line. The command

`#$ -eo gaga`

send all of the output that normally appears on the screen during an interactive session to a file named "gaga". [Change the file name to something different from "gaga"!] The command

`#$ -mu user@address.ualberta.ca`

sends mail to the user when the job starts. Use your correct e-mail address here. Similarly the command

`#$ -me`

sends mail when the job finishes.

After the preamble you can put any sequence of commands that you would normally enter during an interactive session and these will be executed in sequential order. In the above example the command

`matlab >out « 'eof'`

starts MATLAB, redirects MATLAB output to a file named `out` and takes the input to MATLAB from the following lines until "eof", the end-of-file marker. The lines following "eof" should make sense to the shell, as it interprets these lines. If you understand these principles you can construct any complicated script using the full programming capabilities of

the Kron Shell! To submit the job to DQS, use,

```
user@machine:dir> qsub -G Medium 1 test.job
```

The machines are grouped into "High", "Medium" and "Low" groups based on the hardware capabilities. Here "-G" identifies the group as "Medium" and only one machine is requested for the job with "1". Note that DQS supports "parallel virtual machine" for jobs that can be executed in parallel on more than a single machine. In such cases you should request the number of machines by replacing "1" with "n" where "n" can be between 1 and 9 since we have only nine machines. The default group is "Medium" and the default number of machines is "1". So you could have simply entered,

```
user@machine:dir> qsub test.job
```

The command `qstat` shows the current status of your job.

```
user@machine:dir> qstat
```

A.11.2 Example of a FLOW3D job

Construct a *script file* named, say *m01.bat*, containing the following lines.

```
#!/bin/csh
#$ -cwd
#$ -eo gaga
#$ -mu user@address.ualberta.ca
#$ -mb
#$ -me
runf3d -fort m02.f -command m02.fc -release 3.2.1 -geom m01.geo
```

A.12 Printing reports, graphs *etc.*

In room CME 244 there are several old Epson printers connected directly to the OS/2 machines. Each printer serves two OS/2 machines. In room CME 473 each of the DOS machine is connected directly to a HP Laser Jet printer.

Since AIX and OS/2 have network support, printing on these machines can be done using network printers. Hence there are no printers connected directly to each of these machines. There are two network

LaserJet printers, one located in room CME 244 and the other in CME 473. The network printers also operate on the client-server model. **It is important that you understand how the print servers operate and use this facility in a responsible manner**. If the facility is abused this service will be discontinued and you will have to take your print jobs to the Micro DEMO center at the book store!

- You are responsible for providing your own paper.
- Do not send a print request to a remote printer in another location.
- If you send a print job to a remote location, the job can be deleted by the system administrator if it interferes with other queued print jobs.
- Be present in the room where the printer is located and be prepared to attend to the printer immediately after you send a print job to the printer - *i.e.*, feed paper, watch for paper jams, collect your output *etc.* .
- The printers are to be used only for course/research related work. For personal needs take your print job to the Micro DEMO center.

Most applications running under OS/2, AIX or DOS can generate printed output for a number of different types of printers. Each type of printer/plotter understands a particular set of instructions. For example HP LaserJet uses PCL (Print Control Language) format, while HP plotters use HPGL (HP Graphics Language). Another widely used page description language is called Postscript. Application programs use something called a printer driver to generate the output suitable for a particular output device.

If the output is generated on a computer that is connected directly to a printer (like most DOS machines to HP LaserJet in CME 473 or OS/2 machines to Epson in CME 244) then you can send the print job directly to the printer, typically through the printer port LPT1:

If the output is generated on a computer that does not have a printer connected directly to it or you want a LaerJet output from an OS/2 machine, then you must use one of the network printers. In this case you must first save the output from the application program into a file. Most well designed application programs will give you the option to select the output device and to save the printed output into a file. Use the following convention in naming the output files:

filename.ps	postscript
filename.eps	Encapsulated postscript (Useful for merging graphics with text)
filename.pcl	HP Laser Jet
filename.hgl	HP plotters using HPGL instruction set
filename.dot	Epson dot matrix printers

Note that such files contain specific instructions that are understood by specific printers/plotters. Hence they must be sent to the appropriate printers. If you send a PCL file to an Epson printer you will get garbled output that will make no sense!

A.12.1 Using the network printer from AIX machines

The simplest approach is to use the script named **prnt** on AIX machines available within the chemical and materials engineering department. It is a script written by Bob Barton. Hence it is available only on machines maintained by Bob Barton.

`user@machine:dir>` **prnt** *options file_name*

Unix experts may want to try the basic set of unix commands **lpr, lpq, lprm** to submit, monitor and manage a print task. On the AIX machines, the command to submit a PCL file to a network printer is

`user@machine:dir>` **lpr -P***printer_name file_name***.pcl**

where the *printer_name* is either LJ_244 (LaerJet printer in room CME 244), LJ_473 (LaserJet printer in room CME 473) or PS_475 (the postscript printer in room CME 475). Your job is then queued and you can examine the status of the queue with the command,

Note that PS_475 is available only on the graduate network, while LJ_244 and LJ_473 are available on the undergraduate network.

`user@machine:dir>` **lpq -P***printer_name*

You can remove your print jobs from the queue with the command,

`user@machine:dir>` **lprm -P***printer_name job_number*

where *job_number* is the number returned by the `lpq` command.

If your application program does not support the HP Laser Jet (*i.e.*, PCL format) output device, you can select the postcript output device and save the file as *fname*.ps. A public domain utility program called

"ghostscript" allows such postscript files to be either previewed on the monitor of an AIX machine under X-windows or convert the file to PCL format for printing on a Laser Jet printer. To use this conversion program enter,

```
user@machine:dir> gs -sDEVICE=laserjet -sOutputFile=fname.pcl fname.ps
```

If you are using X-windows and want to preview the contents of the file *fname*.ps enter,

```
user@machine:dir> DISPLAY=hostname:0
user@machine:dir> gs fname.ps
```

where *hostname* is the name of the X-client. To find out more about "ghostscript" options use one of

```
user@machine:dir> gs -?
user@machine:dir> man gs
```

Important Note: Once you have printed the document delete the output files from your home directory.

A.12.2 Using the network printer from OS/2 machines

The "lpr" command is also available on OS/2 machines. From an application program you should select the output device to be HP LaserJet and save the output into a file as discussed in the previous section. This file will be on the local computer in the directory that you select. You can print such an output file as follows:

On HP LaserJet in CME 473

```
[C:\] lpr -ssprint.ucs.ualbeta.ca -pHPLaserJ fname
```

A.13 Anonymous ftp service - Netlib and other archives

A number of computer sites on the INTERNET provide a useful software distribution service called anonymous-ftp-service. You can signon to their machines as an anonymous user and retrieve (or download) software. You should use **anonymous** as the *userid* and your *e-mail* address

as the password. Recall from section §A.10.4, that the e-mail addresses have the general form *userid@machine*.**eche.ualberta.ca**. You must be courteous in using such services. Normally you are requested to download software only during off-peak hours. You must also conform to all the software licensing conditions. A sample ftp session is given below.

Sample ftp session

```
user@machine:dir> ftp ftp-os2.nmsu.edu
Connected to hobbes.NMSU.Edu.
220 hobbes FTP server (Version 5.1 (NeXT 1.0) Tue Jul 21, 1992)
ready.
Name (ftp-os2.nmsu.edu:kumar): anonymous
331 Guest login ok, send ident as password.
Password: userid@machine.eche.ualberta.ca
230 Guest login ok, access restrictions apply.
ftp>
```

Note that it is not possible to connect to INTERNET from the subnet 129.128.44.*nnn* (*i.e.*, from room CME 244). Some well known sites and the type of software that they serve are listed below.

Type of Software	internet address	domain name
MS-DOS related software	192.88.110.20	WSMR-SIMTEL20.ARMY.MIL
MS-DOS related software	128.252.135.4	wuarchive.wustl.edu
AIX related software	128.97.2.211	Harpo.SEAS.UCLA.EDU
dynamical systems theory	132.239.86.10	lyapunov.ucsd.edu
TEXrelated software	192.92.115.8	Niord.SHSU.edu
TEXrelated software	134.173.4.23	ymir.claremont.edu
NeXT related software	128.210.15.30	sonata.cc.purdue.edu
OS/2 related software	128.123.35.151	ftp-os2.nmsu.edu
U of A ftp server	129.128.76.12	ftp.srv.ualberta.ca
numerical analysis software	192.20.225.2	research.att.com