Unix Tutorial
Statistics Orientation

blairc@stat.rice.edu
helpdesk@stat.rice.edu

20th August 2003

Introduction

Unix is the most powerful environment for computation in statistics due to its stability and performance (on the servers like stat007.stat or jungle.owlnet). Throughout your career you will encounter statistical software in the Unix environment. While Unix may seem foreign and intimidating, we're going to try and help ease the transition for you. Just remember that, while Unix is more complicated than the windows platform (and thus has a higher learning curve), it is also much more powerful and configurable. You can change the prompt, the size and position of windows (like emace or netscape) when they start, keyboard shortcuts (aliases), the desktop (gnome/twm/kde) and more.

This tutorial is intended to get you familiar with the Unix environment so that you will be able to use Splus, R, SAS, Matlab, Mathematica, Maple, Emacs and LaTeX/Lyx. These programs are for numerical computation (Splus, R, SAS, MATLAB), symbolic manipulation (Maple, Mathematica), preparing publishable documents (LaTeX) and other programs that make it all possible (Emacs, pico, StarOffice,...).

And the best way to learn is by trying new commands- remember that, at this stage, most everything you do can be quickly and painlessly undone. For help, email helpdesk@stat.rice.edu (`mailto:helpdesk@stat.rice.edu`).

Resources

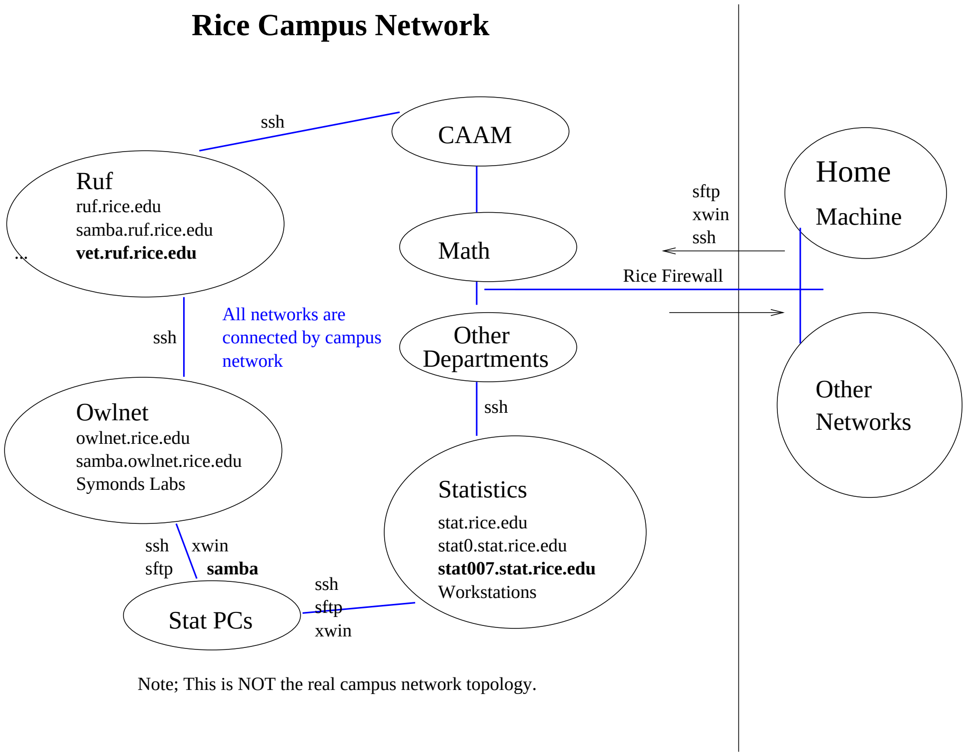
Here are some resources you should check out, especially if you have any questions. The authors have also put together a list of tutorials, guides and references, available at http://www.stat.rice.edu/~blairc/stat/guides.html (`http://www.stat.rice.edu/~blairc/stat/guides.html`). These include the "Student Computing Guide for the Stat Dept". A great source of beginner Unix help is available at http://www.mcsr.olemiss.edu/Unixhelp/ (`http://www.mcsr.olemiss.edu/Unixhelp/`). Google and the web also provide some decent help, try a search for “Unix tutorial step by step”. In the library, try “The Unix Programming Environment”, by Kernighan and Pike for the basics, “Unix in a Nutshell” by Robbins for help with commands, shells, regular expressions and more, or for all those dirty tricks, “Unix Power Tools”, by Peek, O’Reilly and Loukides.

For quick help at the terminal, you can use `man` or `siteman` if `man` returns `No manual entry for ...`. If you want to know about commands that convert files to or from pdf format, try typing `apropos pdf` at the command line. If you know a command, but don’t remember which flags you want to use, try `man tar` or `siteman tar`, to learn about the options for `tar`.

Rice University also has some computing guides of dubious age and quality, but they have some interesting stuff in them, http://riceinfo.rice.edu/-Computer/Documents/ (`http://riceinfo.rice.edu/Computer/Documents/`).

If you want specific information about owlnet or ruf, try their webpages, http://www.owlnet.rice.edu (`http://www.owlnet.rice.edu`) or http://www.ruf.rice.edu (`http://www.ruf.rice.edu`).

Figure 1: campus network



Introduction to the Campus Network

As this tutorial begins, you should be sitting behind a PC in the Symonds II lab. This tutorial is all about using Unix, so we will focus on Unix, with a few side comments for windows users.

The first goal is to get you aquainted with the networking on campus. By the end of your first week of classes, you should have at least two separate Unix accounts, one on stat and one on ruf. It would be nice to also have an owlnet PC account so that you can use Symond Lab II PC's. In terms of the picture below, if you are "in the circle" (ie sitting in front of a Unix machine of that network), all you need to do is enter your username and password. If you are not "in the circle", you need to use software (ssh) to connect to the network (the lines between circles).

First of all, for our purposes, a network is a bunch of computers that you can login to, and regardless of the machine you login to, all of your files will be accessible. On the stat network, I can login to any of the following Unix machines, stat007.stat.rice.edu, nova.stat.rice.edu (public workstation at dh2093), etc, and each time I log in, the same files will be there waiting for me. (sidenote: this is called a Network File System, NFS)

There are various ways to connect to the networks. The easiest is to find a machine connected directly to the network, perhaps the one at your desk in your office if it's a Unix machine. But you don't have to be sitting in front of a Unix machine to use it, you don't even need to be anywhere near it, all you need is a way to talk to it (opening a terminal and communicate through SSH).

Connecting to the Stat Network from Windows Using SSH

Now we're ready to login to the stat network. First, you need to login to the computer in front of you. These PC's are on the owlnet network. You will need an owlnet Unix account and then an owlnet PC account to login. We have training accounts for this tutorial and you will have plenty of time to apply for these accounts. (You will seldom use them.) For more information about accounts and how to change passwords, see Chapter 1 of the computing guide, http://www.stat.rice.edu/~helpdesk/compguide/ (http://www.stat.rice.edu/~helpdesk/compguide).

There should be a program called "Secure Shell Client" in the list of programs on the PC in front of you. If you press enter or the space bar, you should be prompted for three things, a "Host Name", a "Username" and a "Password". For the hostname, enter "stat007.stat.rice.edu", enter your username and password if you have one, otherwise raise your hand and we will give you a temporary one to use. Now you should be logged in, and something like this should greet you:

```
SSH Secure Shell 2.2.0 (Build 123)
Copyright (c) 2000 SSH Communications Security Corp - http://www.ssh.com/

This copy of SSH Secure Shell is licensed for educational, charity,
and personal recreational/hobby use.
Any commercial use requires a separate license.


Last login:  Tue Aug 20 2002 21:38:57
Sun Microsystems Inc.  SunOS 5.7 Generic October 1998
You have mail.
stat007%
```

If this happens, congratulations, you're logged in, if not, raise your hand and complain.

Command Prompt

Now we'll begin learning about what you can do with Unix. First, we'll list (`ls`) the files in the current directory. Try this:

```
stat007% ls
```

If nothing happens that means there are no files or directories in your account. So lets make a directory (`mkdir`) called "Tutorial" and then use `ls`.

```
   stat007% mkdir Tutorial
stat007% ls
Tutorial/
```

Now we will learn how to change directories (`cd`), and print the working directory (`pwd`), so we know where we are.

```
stat007% pwd
/home/blairc
stat007% cd Tutorial
stat007% pwd
/home/blairc/Tutorial/
```

Pathnames and Files

These "path names" tell you where you are. By default, you are placed in your `$HOME` when you log in, usually `/home/username`. Another way of remembering your username is the `whoami` command. There are two ways to use path names. The first way is the relative pathname. When we used the command `cd Tutorial`, this was an example of a relative pathname, because the directory `Tutorial` was one directory up from where we were. In absolute pathnames, you started in `/home/blairc`, and then went to `/home/blairc/Tutorial`. Absolute pathnames always being with a slash, '/'. When we used the `pwd` command, this returned the absolute pathname. Let's practice using relative and absolute pathnames some, but use your own username:

```
stat007% cd /home/username/Tutorial
stat007% pwd
/home/username/Tutorial
stat007% cd /home/username
stat007% pwd
/home/blairc
stat007% cd Texas
Texas:  No such file or directory.
stat007% cd Tutorial
stat007% cd ..
```

```
stat007% pwd
/home/blairc
stat007% cd .
/home/blairc
stat007% cd ..
/home
stat007% cd username
/home/blairc
stat007% cd /home/username/Tutorial
stat007% cd ../Tutorial
stat007% pwd
/home/blairc/Tutorial
stat007% cd ../tutorial
../tutorial:  No such file or directory.
```

Now we're learned how to go backwards using relative pathnames '`..`', as well as stay where we are '`.`'. What happens when you use the `ls` command in the `/home` directory? What happens when you use the `ls` command in the `/` directory? (`/` directory is called the "root" directory- try going to the root directory and typing `cd ...` What happens? What if you type `cd;pwd`? (the semicolon tells it that a newline is starting) Let's learn a little more about files, what kinds of files there are, and how we can manipulate them. First we need to learn a little more about `ls`.

There are files in your directory that `ls` didn't tell you about. These files are "hidden", because you probably won't ever touch most of them. We can tell `ls` to list all files using the `-a` flag.

```
stat007% ls -a
./ ../ .alias .bash_logout .bash_profile .bashrc .cshrc .emacs
```

These are sometimes called dot files, and these usually control the settings of different applications (many of them end with 'rc'). Would you like to know more about what's in them? Pick one, or use one of mine (hint: do not open the .Xauthority file, as this could cause your terminal to become illegible, causing you to have to log out, then login again- try it if you want to see what happens). To move through the `more` command, use the space bar (to go down by pages), the return key (to go down by lines), the "b" key (to go back), or the "q" key to quit.

The goal here is to learn how to use the `more` command- use any file in your directory- I just happened to choose the file named `.alias`. Do you have a `.cshrc` file in your directory? A `.alias` file?

```
stat007% more .alias
# Blair's aliases
#
# Alias for prompt is in .tcshrc, all others below.
# File management/System calls
alias lsa 'ls -a'
alias lsl 'ls -l'
alias lsal 'ls -al'
alias lsla 'ls -al'
alias lal 'ls -al'
alias lm 'ls | more'
#change file to executable
alias chx 'chmod a+x $*'
# list directories
alias lsdir 'ls -l | grep ^d | more'
alias ls 'ls -F'
```

The `more` program lets you view the contents of files, and has lots of options (for example the "/" lets you search for a word). More on this later. There's also a similar command called `less` that does about the same thing (try it).

Optional Examples (If you're feeling adventurous)

If you're moving fast, try this example, otherwise skip it and come back later (it's not that important). Notice that in the example above, I have a list of aliases (these are read every time I log in).

```
stat007% lsa
lsa:  Command not found.
stat007% alias lsa 'ls -a'
stat007% ls -a
./ ../ .alias .bash_logout .bash_profile .bashrc .cshrc .emacs
stat007% lsa
./ ../ .alias .bash_logout .bash_profile .bashrc .cshrc .emacs
stat007% which lsa
lsa:  aliased to ls -a
```

What happens if you type `which ls` or `which Splus` or `which alias`? You can undo an alias using `unalias lsa`. We will come back to this during C-shell toturial.

Changing Your Shell (Making Life Easier)

First of all, type `echo $SHELL`. If it says `tcsh` at the end, then you can skip this section (and the functions below should work fine), otherwise, read on. If you have problems changing your shell (maybe your password isn't accepted), then you can simply type `tcsh` at the prompt, and it will be your shell for this session only (until you type `exit`.

Are you having problems with the backspace key? Does it make funny characters on your screen? Wouldn't it be easier if you could just press the up arrow

to scroll through previous commands? Or just type the first couple of letters of a file name and have it autocomplete? Well, those are the advantages of using a different shell, like tcshell. We won't dive into any details, but this should make your life at the command prompt a lot easier. So, do this next example, and ask any questions if things don't work right. Just make sure you make the `/usr/local/bin/tcsh` your new shell.
If you are using a training account, it should already be the default. This topic is also covered in the C Shell tutorial.

```
stat007% passwd -r nis -e
Enter login(NIS) password:
Old shell:  /usr/bin/csh
New shell:  /usr/local/bin/tcsh
Login shell changed.
```

You will need to wait around ten minutes and log-out log-in to make this change affective so you can simply type `tcsh` to proceed.

Files, Moving, Copying, Removing

Note that the goal of this section is to get a replica of my `/home/blairc/Tutorial` directory and all of its contents into your home directory. We'll see how `cp` and `mv` can and cannot help us with this.

Now it's time to learn about files in Unix, specifically how to move them (`mv`, copy them `cp`, search through them and finally remove them `rm`. Let's see if the `cp` command can help us out. Try these commands yourself (Now I will only show error messages, not the regular output):

```
stat007% cd
stat007% pwd
stat007% cp /home/blairc/Tutorial/compguide.pdf Tutorial/
stat007% ls Tutorial/
stat007% ls /home/blairc/Tutorial/
```

What if we want to get all of the files in the `/home/blairc/Tutorial` directory? We'll try a couple of ways to do this. What happens if we want to look at them? The file extensions here are for postscript files (`ps`), encapsulated postscript files (`eps`, typically returned by mathematics software and inserted into publications), portable document format files (`pdf`) and Micosoft Word documents (`doc`). You'll come across all of these in the Unix environment, and it's important that you know how to open them. What happens if we try to use `cp` or `mv` without any flags?

```
stat007% cd
stat007% pwd
stat007% mv /home/blairc/Tutorial/ Tutorial/
mv:  /home/blairc/Tutorial:  Permission denied
stat007% cp /home/blairc/Tutorial/ Tutorial/
```

```
cp:  /home/blairc/Tutorial:  is a directory
```

Well, we can only `mv` files that belong to us, and we can’t copy directories without a special flag. There are two ways to get these files- let’s try both. We can use something called a wildcard ’*’, as well as the ’recursive’ `-r` flag for copy.

```
stat007% cd
stat007% pwd
stat007% cp /home/blairc/Tutorial/* Tutorial/
stat007% ls
stat007% ls Tutorial
stat007% mv Tutorial/README ./
stat007% ls
```

Notice that the ’*’ only copied files, not directories. You can also use commands like ’*.ps’ to only copy postscript files. Now we’ll try some of the new features of the tcshell out. First, press the up arrow. What happens? Now try this:

```
stat007% ls /ho
```

and press the TAB key. It should finish off the directory name. Now try:

```
stat007% ls /home/usern
```

and press TAB. Does it autocomplete? Why or why not?

```
stat007% ls /home/username/Tu
```

and the TAB key one more time should get you...

```
stat007% ls /home/username/Tutorial
```

How fast can you get to /home/username/Tutorial/Scripts/Dir...? (relative pathnames are faster).

```
stat007% more README
stat007% rm Tutorial/*
stat007% ls
stat007% ls Tutorial
stat007% rmdir Tutorial
stat007% ls
```

Now we’ll try this another way...

```
stat007% cp -r /home/blairc/Tutorial/ ./
stat007% ls
stat007% ls Tutorial
stat007% cd Tutorial
stat007% pico README
```

Pico is one of the simplest text editors in Unix. You can move the cursor around with the arrows, type, and use the backspace or delete key to remove things. This introduces another common Unix feature- macros. Notice at the bottom of pico that there is a list of things like `^X Exit`. The carrot, ^, means that we use the control key and the X key (hold down the control key, then press the X key quickly). What can you find with ctrl-T? (spell checkers are a weakness in Unix). While we're at it, try this out as well:

```
stat007% cd
stat007% rm Tutorial
stat007% rmdir Tutorial
```

How can you get rid of a directory that's full of stuff? Or directories of directories? Well, there are two flags we'll use, the recursive flag `-r`, and the force rm flag, `-f`. First another quick trick (using aliases).

```
stat007% alias rmd ’rm -rf \!*’
stat007% which rmd
stat007% rmd Tutorial
stat007% ls
```

Now it's gone. Notice that the \`!*` tells the alias that you can pass arguments to it. (ie, everything after rmd is placed there, so `rmd Tutorial OtherDir` is exactly the same as `rm -rf Tutorial OtherDir`. Try this if you're confused (and ask questions). Also, note that sometimes you will be prompted by `rm` or `cp` if you really want to remove these files. That is because these commands are aliased with the `-i` flag. To stop this, type `unalias rm` for example. To return to it to asking if it should delete, type `alias rm ’rm -i’`. (many of these aliases are set up in either your `.cshrc` file or other places. If you add these aliases into your `.cshrc` file, they'll be available everytime you login).

Optional Exercises

If you're still moving pretty fast, try these as well. What happened to the directory `Scripts` in `Tutorial` when you copied it each way? What happens when you use the recursive flag for `ls` (it's really `-R`, so what happens when you type `ls -R`? Also, try making a directory like this: `mkdir NewDir/OtherDir/InsideDir`. Why does it complain? What if we make the parent directories, `mkdir -p NewDir/OtherDir/InsideDir`? Would it be nice to make an alias for this, so that mkdir automatically made parents? Get a file to delete. What happens if you try `rm -i`?

Opening Files of Different Types, An Intro to X

Up to now, everything we have done has been text based. Now we're going to start using some graphical applications to view postscript, pdf and Microsoft Word files. In order to do this, we must use the graphical part of Unix, know as either “X” or “X-windows” (the name is from historical reasons- apparently the original “W”indows didn't work so well). But before we begin here, take a break, get a drink, and ask any burning questions. Don't worry if you're running out of time- you can work on this later.

Alright, now that you're back, we'll do two things in order to make sure that X is working on your PC. First, in the Secure Shell, go to the Edit menu (at the top), select Settings, then click on "Tunneling" (in the "Host Settings" list). Then make sure that "Tunnel X11 Connections" is checked. This means that you will let your PC recieve X-windows (graphical things) from Unix. To display them on your PC, you need something called X-win32. When it's running, there is a blue "X" in the lower right hand corner by the time. If it's there, you're fine, otherwise, select X-win32 from the Program menu. Ask if you have any problems.

Now you should be back from a break and have X-win32 running in the background (the small blue X in the lower right hand corner). Now we'll check out some of these files. Make sure you copy those files back from the /home/blairc/Tutorial directory.

```
stat007% cd
stat007% ls
stat007% cd Tutorial
stat007% gv compguide.ps
```

What happens now? Can you type at the prompt while gv is open? Unix only allows you to run one process at the command line at a time. This isn't as bad as you think- all you have to do is open another command line in the following way...

```
stat007% gv compguide.ps &
```

The ampersand tells it to start a new process, so now you can type at the command line while it's running. What happens if you don't want to? Exit `gv` and try this:

```
stat007% gv compguide.ps
```

Now type `ctrl-z` at the terminal (hold down the control key and press the z key quickly). You should get a message like the following, then tell it to run in the background (`bg`).

```
^Z
Suspended
stat007% bg
```

This does the same thing as running it with the ampersand (Wouldn't it just be easier to have an alias so didn't have to worry about this?) We can do the same thing with `pdf` files.

```
stat007% acroread statguide.pdf &
```

What about those Microsoft Word documents? Well, there is an "office" like program in Unix. The one installed here is Sun's StarOffice (and it used to be available free for all platforms).

```
stat007% /usr/site/staroffice/bin/soffice &
```

We have to enter the entire $PATH here. Ask if you want to know why. By the way, all installed software is in the /usr/site directory. Answer the questions, and then open the 532.doc.

SSH/SFTP in Unix, Owlnet, Stat and RUF

If you want to login to a different network, like Stat to owlnet, or stat to ruf, or if you want to transfer files between any of these networks and PCs, you need to know about SFTP (Secure File Transfer Protocol).

Now we will login to an owlnet Unix machine. You can only do this if you have your own account (training accounts won't work). To do this you need to know which machine to login to (pick one from the list at http://www.owlnet.rice.edu/docs/servers.shtml (`http://www.owlnet.rice.edu/docs/servers.shtml`), we recommend verm, jungle or forest). You also need to know your owlnet username and password (same username you logged into this PC with, but with a different password). Try this at the Unix prompt:

```
stat007% ssh verm.owlnet.rice.edu
```

You should be asked if you want to accept the host key (yes), then for your password. Type `exit` to exit. Now you are logged into the owlnet network. You can similarly login to RUF:

```
stat007% ssh kennel.ruf.rice.edu
```

Now suppose you would like to copy files from stat to owlnet (or vice versa). Note that you will end up with two copies of the file, one on each server. To put files onto the other server, we'll use `sftp`'s `put` command, and if we have any files over there that we want to get, we'll use the `get` command, like this.

```
stat007% sftp jungle.owlnet.rice.edu
```

To put files onto the other server, we'll use `sftp`'s `put` command, and if we have any files over there that we want to get, we'll use the `get` command, like this.

```
stat007% sftp jungle.owlnet.rice.edu
blairc@jungle.owlnet.rice.edu's password:
sftp>
sftp> get statguide.tar
statguide.tar | 710 kB | 710.0 kB/s | TOC: 00:00:01 | 100%
sftp> put 532.doc
532.doc | 41 kB | 41.5 kB/s | TOC: 00:00:01 | 100%
sftp> quit
```

Here's a trick- if you want to ssh into a network, but you have a different username, then you'll want to use the `-l` flag, like this: `stat007% ssh -l othername jungle.owlnet.rice.edu`

It also works with `sftp`.

File Permissions

You certainly don't want anyone on the network to read your mail, delete or move your files or access you classwork. Read this tutorial for more information... http://www.mcsr.olemiss.edu/Unixhelp/tasks/access_permissions.html (`http://www.mcsr.olemiss.edu/Unixhelp/tasks/access_{p}ermissions.html`). The two commands you'll use here are `ls -l` to view the permissions and `chmod` to change them.

Optional Exercises

If you have any pictures you'd like to convert (from say, a jpg file to an eps file so you can put it in a latex document), there are two programs, `xv` and `gimp`. The latter is more flexible. Try to convert the Sparks1.jpg file to an eps file. (ask if you want to know what this means).

Finally, here are some really great tricks that I use frequently, they include `grep` and `find`. The former lets you search through a file for a particular word, and find lets you recursively search through directories to find a filename. You can use them together (Oh, I know I used that "key" command (a better legend) in an Splus script, I just can't remember which one...)

```
stat007% cd Tutorial
stat007% grep -n 'key' *.q
stat007% find .  -name '*.q' -print
stat007% find .  -name '*.q' -print | xargs grep -n 'key' /dev/null
```

Optional File Compression (tar, gzip, compress)

During your time here, you may come across files with the following file types: `Z, gz, tar, tgz, tar.gz`. Compressed files have the `Z` extension, gzipped files have the `gz` extension, and tarred files have the `tar`. Files that have been tarred and zipped have the extension `tar.gz` or `tgz`. Tarring makes one file out of entire directories. Try the following examples:

```
stat007% cd /home/username/Tutorial/Src
stat007% ls
stat007% uncompress ashRegScripts.q.Z
stat007% ls
```

```
stat007% gunzip ashRegExamples.q.gz
stat007% tar -xvzf ashRegression.tgz
stat007% ls
```

Note that the 'z' flag in the tar command means the files are zipped. If you have a plain `.tar` file, then just use the flags `xvf`, which stand for eXtract, View on the File that follows. Or, to create these files...

```
stat007% ls
stat007% compress ashRegScripts.q
stat007% ls
stat007% gzip ashRegExamples.q
stat007% tar -cvf ashRegression.tar ashRegression.tar ashRegExamples.q.gz
ashRegScripts.q.Z
stat007% ls
```

That's not quite the same, but you can figure it out. The new `tar` flag stands for Create a new file

Printing

Printing is not the easiest thing in Unix. For information on all of the glories of printing, open the compguide.ps or compguide.pdf file, and check out Chapter 3, http://www.stat.rice.edu/~helpdesk/compguide/ (`http://www.stat.rice.edu/~helpdesk/compguide/`).

Being Nice (Job Priorities)

Suppose you're running Splus on a very large script, and it's taking up lots of processor time, and could run for a while. Well, you should be nicing this process (lowering it's priority so people doing things like checking their email or using emacs don't get slowed down by your process. Here's how it works. You can start a job "niced" like this (we highly suggest this for any job over 5 minutes):

```
stat007% nice +10 Splus
```

However, if you forget, go through these steps (try this now):

```
stat007% Splus
Welcome to Splus...
>
```

Press `ctrl-z`, then

```
Suspended
stat007% ps
PID TTY TIME CMD
28170 pts/12 0:00 Slmclien
```

```
27489 pts/12 0:18 emacs
28168 pts/12 0:01 Sqpe
27416 pts/12 0:00 tcsh
```

Splus runs under the name Sqpe. So what we do is "renice" it, then bring it to the foreground (unsuspend it), like this:

```
stat007% renice +10 28168
stat007% fg
Splus
```

To exit Splus, type `q()`.

Other commands that you should know about include `top/prstat` (the former on some machines, the latter on others). This tells you who is using the processors at the moment (you can tell who is slowing everything down). Also, if you have a job that isn't behaving, you can `kill -9 PID`, where the `PID` comes from the `ps` command (like above).

Email

Pine is the easiest and most commonly used email client in Unix. It's easy enough to use without documentation, but if you want to do things like have different folders, sort old mail, change your signature or more, you should check out there webiste, "Using Pine". Try it out with `pine`. This won't work on the training accounts (since your mail won't be delivered there, and these accounts will be deactivated soon). For more information, see http://www.washington.edu/pine/ (`http://www.washington.edu/pine/`).

You might also be interested in using Netscape's mail client, or using mutt (advnaced) or another supported mail client (like the one in staroffice).

Definitely read Chapter 4 of the Computing Guide for more information on email, http://www.stat.rice.edu/~blairc/stat/guides.html (`http://www.stat.rice.edu/~blairc/stat/guides.html`).

Special Keys, Tricks

If you want to change the font in emacs, you can hold down shift and then hold down the left mouse button.

When you're at a terminal, hold down control and press the right mouse button to change the font size.

Here are some keys that can save you time (in emacs and at the tcsh command line). Remember that the meta key is the same as escape on a windows keyboard.

ctrl-d	delete forwards	meta-d	delete words forward
ctrl-w	kill previous word	backspace	deletes backwards
ctrl-k	cuts current line of text	ctrl-y	pastes the line of text
ctrl-a	go to beginning of line	crtl-e	go to end of line
meta-f	move forward one word	meta-b	move backwards one word
ctrl-l	clear screen	ctrl-t	transpose characters

Note that you can change these key bindings in several ways (like using `cua` in emacs) to get windows-style key bindings or other key bindings. Also, if you are part way though a command, `ctrl-d` will tell you possible completions.

```
stat007% nets

ctrl-d
netscape netscape.old netstat
```

Disk Space, Quotas and File Backups

As another sidenote, there are quotas on your RUF and Owlnet accounts (see the compguide for more info, http://www.stat.rice.edu/~blairc/stat/guides.html (`http://www.stat.rice.edu/~blairc/stat/guides.html`)). There is no quota on the stat network (it is intended for research, read the Computing Agreement you signed when you got your Unix account for more info on proper computer use, see Chapter 11, or mail problem@rice.edu). To check how much disk space you're using, try `df` from your home directory, or `ds -sk` for a summary of disk usage in kilobytes (tells you which files/directories are taking up space). Also note that the only time your files are backed up is when you're on a Unix system. If you use a PC, and your computer dies, there is nothing the dept can do for you if you haven't backed up your data to the Unix servers.

A check list

Here is a list of commands (skills) that you should definitely know to start using Unix system(s).

1. log in to the servers from a Unix terminal and/or using windows+ssh shell.
2. concepts: shell, absolute/relative path, `$path`, .cshrc, file permissions (optional)
3. directory/file manipulation: `cd`, `mkdir`, `rmdir`, `ls`, `mv`, `cp`, `rm`
4. text editor: `pico`, `emacs/vim` (learn one of them later, preferably emacs).
5. get help: `man`, `siteman`, `which`
6. ssh/sftp: `ssh`, `sftp`
7. file viewer: `cat`, `more`, `head/tail` (optional) (for plain text file), `gv` (for `.ps` files), `acroread` (for `.pdf` files), `xdvi` (for `.dvi` files)
8. print: lpr or through acroread, gv.
9. email and web: pine, netscape, mozilla,
10. (optional) alias, `df`, `du`, `gzip`, `tar`, `ps`, `top`, `nice`, `grep`, `find`, ...