1 Changing Your Password

The command `passwd` is used to change your password. Note that there are rules about how long your password is as well as what characters are allowed.

```
easgst26@morlock(57-11:31am)> passwd
passwd: Changing password for easgst26
Enter login(NIS) password:
New password:
pwd(System): Password too short - must be at least 6 characters.
New password:
pwd(System): The first 6 characters of the password must contain at least two alphabetic characters and at least one numeric or special character.
New password:
Re-enter new password:
```

2 Creating and Editing Files

To create a file, simply invoke an editor, passing the new file name as the argument. This is the same command you’ll use to edit an existing file. All editors on Unix will create a new file if the file you’ve specified doesn’t exist, or will edit the existing file if it exists. On Sun machines, the editors available are vi, emacs, xemacs, and pico (there may be others available). On SGI machines, jot and nedit are also available. There is plenty of online documentation available on editors. See http://wings.buffalo.edu/computing/documentation/unix/catalog.shtml for help with editors.

```
easgst26@morlock(61-11:35am)> vi newfile.cpp
```

3 Terminal Type

Some applications require that you set your terminal type. Ordinarily, this is set for you when you log in. But, if you are logging in from one machine to another, the terminal type might not be set automatically. Most of the time
applications complain about your terminal type, you can use the type vt100 to fix the problem. To do this, simply type set term vt100 at the prompt.

```
easg26@ming(51-11:40am)> vi newfile.cpp
iris-ansi-net: Unknown terminal type
I don't know what kind of terminal you are on - all I have is 'iris-ansi-net'.
[Using open mode]
"newfile.cpp" [New file]
:q!
easg26@ming(52-11:40am)> set term=vt100

4 X Windows From Remote Machines

For graphical editors, such as xemacs, it is sometimes necessary to set your X Windows parameters. This, again, is usually done for you when you log in but sometimes, especially if you’re sitting at one machine but working remotely on another, you will get error messages indicating that X is incorrectly set up. You need to do two things to fix this- first, from the machine you’re working on (the remote machine) you have to set the environment variable DISPLAY to be the machine you’re currently sitting at. Next, you have to set permissions from the machine you’re sitting at, to allow the remote machine to display to the local machine’s display. This is done with the xhost command.

You might get an error like this:

```
easg26@ming(53-11:40am)> xemacs newfile.cpp
Xlib: connection to "remes.nyscddi.buffalo.edu:0.0" refused by server
Xlib: Client is not authorized to connect to Server

Initialization error: X server not responding
: "remes.nyscddi.buffalo.edu:0.0"
easg26@ming(54-11:52am)>
```

To fix this, you would do the two steps described above:

From the remote machine:

```
easg26@ming(54-11:52am)> setenv DISPLAY remes.nyscddi.buffalo.edu:0.0

```

The ":0.0" part specifies the first monitor of the first display. You will most likely never vary from this.

From the local machine (the machine you’re sitting at):
5 Environment Variables

You'll sometimes have to modify environment variables to get certain things to work properly. There are two types of environment variables- system environment variables and application environment variables. The former are set using the `setenv` command. The latter are set using the `set` command. To see a list of the current environment variables, simply type `set` or `setenv`. To set a particular variable, the syntax is `set variable-name value`.

```
chugh@rennes(114-11:58am)>> setenv
_/=/usr/sbin/xsh
MANPATH=/usr/share/catman:/usr/share/man:/usr/man:
LANG=C
PAGER=less
VISUAL=vi
  _SGI_DTLAUNCH_EFFECT=1
PATH=/nyscedi/home/chugh/bin:/usr/local/bin:/usr/freeware/bin...
NOMSGLABEL=1
WEBBROWSER=/usr/bin/X11/netscape
BOOKVIEWER=/usr/sbin/infosearch -books
MAILBOXPROG=/usr/bin/X11/netscape mailbox:
XUSERFILESEARCHPATH=/nyscedi/home/chugh/.desktop-rennes.ny...
EDITOR=vi
LOGNAME=chugh
...
chugh@rennes(115-11:58am)>>
chugh@rennes(117-11:59am)>> setenv EDITOR emacs
chugh@rennes(118-11:59am)>>
chugh@rennes(118-11:59am)>> setenv
...
EDITOR=emacs
...
chugh@rennes(118-11:59am)>>
```

6 Creating Directories

To keep things organized, you'll want to store your files in directories. To create a directory, use the `mkdir` command. To remove a directory, use the `rmdir` command. Note that you can't remove a directory if it contains any files. To set your current directory, use the `cd` command.
7 File Management

You will undoubtedly need to rename, delete, copy and move files. To move and rename files, the command `mv` is used. It performs both renaming and moving functions. The syntax is `mv source destination`. Both `source` and `destination` can be files or directories. To delete files, use the `rm` command. The argument can be either a file or a `wildcard`. A `wildcard` is an argument that means "all the files that match this pattern". The wildcard "*" means "all files in the current directory", and is the most commonly used wildcard. To copy files, use the `cp` command, which has the syntax `cp source destination`. Here, `source` can be a file or a wildcard, and `destination` can be either a directory or a file (or wildcard). You'll also need to use the `ls` command, which lists files. Use the `-l` option to get a more verbose listing.
8 Class Directory

Every person registered for the class has a directory where you can store your projects and homework. The directory is /eng/class/mae473-573/ username where username is your login name. This directory has 150 megs disk space for your work. To see how much space you have, use the quota command, with the -v option.

```bash
easgst26@ming(73-2:29pm)> quota -v
Disk quotas for easgst26 (uid 85173):
... /eng/home/easgst26
    7470  100000  100050
    752  10000   10050
```
This command may give you quota information for other users as well as your own depending on how it has been configured on the machine you are logged on to.

9 Less, More, Cat, and Tail

Sometimes you'll want to quickly look at the contents of a file. To do this you can use the commands less, more, cat and tail. The commands less and more show you the contents of a file, a screen at a time. You press space bar to go to the next page. The command cat simply prints out the contents of the file, so if it's more than one screen long, you will see the whole file fly by the screen until it gets to the end. The program tail prints out the last few lines of a file.

```
$ asgst26@ming(82-2:34pm)> tail .cshrc
#  
# These settings are very useful:
#  
set autoexpand correct=cmd
endif
#
#
# Source the user’s .aliases file if it exists; this file is where all # custom commands should go:
#
if (! $?NOSETALIASES) if (-f ~/.aliases ) source ~/.aliases
$ asgst26@ming(83-2:34pm>
```

Printing Files To print out files, use the lp command. The syntax is `lpr -Pprinter filename`. To check on the status of your print job, use the `lpq` command.

```
$ asgst26@ming(92-3:10pm)> lpr -Pbell .cshrc
easgst26@ming(93-3:10pm)> pwd
/eng/home/easgst26
easgst26@ming(94-3:10pm)> lpq -Pbell
bellxban is ready and printing
Rank  Owner   Job Files    Total Size
active  easgst26  75  .cshrc  3421 bytes
```

10 Managing Processes

To see what processes of yours are running, use the `ps` command. This will show you your current processes as well as their process ID. Sometimes your programs will freeze up and it will be necessary to kill the processes. Use the `kill` command with the `-HUP` option to kill a process.
11 Compiling and Running Programs

There are two compilers available on most Unix workstations. They are cc and gcc. They have almost identical option syntaxes and are extremely sophisticated. The most simple form of using these compilers is gcc sourcefile -o executablename.

```
chugh@morlock(58-3:50pm)> cat hello.c
main()
{
    printf("Hello Cruel World.\n");
}
chugh@morlock(59-3:50pm)> gcc hello.c -o hello
chugh@morlock(60-3:50pm)> hello
Hello Cruel World.
chugh@morlock(61-3:50pm)>
```

12 Logging Out and Locking Your Workstation

When you’re done working, you can log out of your station with the logout or exit command. Alternatively, on most workstations, there is a graphical logout procedure as well. If you’re going to be away from your workstation for a few minutes, you can use the xlock program to secure it. Please note that if you’re going to be gone for more than a few minutes, you should log out rather than lock your workstation, as this is against UB computing policies.

13 Man Pages

Help for all of these commands is available with the command man. To use it, simply type man followed by the command you’d like help with.
asgst26@ming(123-3:15pm)> man cp
Reformatting page. Please Wait... done

User Commands cp(1)

NAME
   cp - copy files

SYNOPSIS
   /usr/bin/cp [ -fip ] source_file target_file
   /usr/bin/cp [ -fip ] source_file ... target
...