

Each of the commands included here must be entered at a UNIX prompt: **bash-3.2\$**

CONTENTS: COMMAND SUMMARY

UNIX commands within this handout cover the following topics:

- Changing Your HP Unix Password
- Using Files
- Printing
- Finding Out About Your Account
- Scripting a Session
- Getting Help on a Unix Command

Changing Your HP UNIX Password

`passwd` Password Change Sequence:

1. Enter the **passwd** command.
2. The system responds:
Changing password for <your login ID>
and prompts for the
Old password:
3. Enter your current password (to confirm your identity).
For reasons of security, the password will not display on the screen. Don't be alarmed. The characters you enter are sent to the server.
If you mistype the password entry, you'll have to start over again. Backspacing does not delete mistyped characters; it adds ASCII backspace characters to the line of entry.
4. The prompt **New password:** appears.
Please refer to *Rules and Suggestions for Passwords at CCSF* on the next page.
You have less than 30 seconds to enter your new password, or the system will end the password change sequence.
5. The prompt **Re-enter new password:** appears.
Enter the new password a second time for confirmation.
6. If you get an error message such as **New password too short**, the password change sequence returns you to Step 4 and gives you another opportunity to enter a new password.
7. If you make repeated mistakes, the system will end the password sequence with the message:
Too many failures — try later.

Rules and Suggestions for Passwords at CCSF

Rules:

1. Passwords must be at least 6 characters (and no more than 10)
2. Passwords must contain at least 2 alpha and at least one numeric character.
3. *Don't use spaces or characters that carry special meaning in Unix, like * ? / \ !*
4. Remember that UNIX is case sensitive. Uppercase letters may be used, but keep in mind that any uppercase letter(s) you use in a password must be entered in upper case *at every login*.

Suggestions:

1. Don't use obvious passwords like your first name; even words found in the dictionary are not good because of their predictability.
2. Try thinking of a phrase that you can remember and use the first letter of each of the words (and at least one number).
3. Example:
From the sentence: "I don't want to do this one,"
you might create the password: idwtdt1

Using Files

Note: in the following sections, *<fn>* stands for the name of one of your saved files.

Displaying the Names of Saved Files

`ls` displays the names of files in the current directory that are not hidden.

`ls -l` or `ll` (All letter l not number 1)

displays detailed information about non-hidden files in the current directory:

- Column 1: the file type and the permissions
(read, write, and/or execute for owner, group, and other users)
- Column 2: the size of the file in little links
- Column 3: the login ID of the owner of the file
- Column 4: the group to which the owner of the file belongs
- Column 5: the size of the file in bytes
- Column 6: the date of the last modification
- Column 7: the time of the last modification
- Column 8: the filename

`ls -a` displays the names of all the files in the current directory (including hidden files).

Displaying the Contents of a Text File

```
cat <fn>
```

displays the file named *<fn>* on the screen.

Example: To see what is in the file called *prog1*, enter

```
cat prog1
```

Use *cat* on short files containing ordinary text created with a simple text editor like *pico*.

Note: if you try to display a compiled program you'll get strange characters on the screen and the responses on your terminal screen might be slow and unpredictable.



```
more <fn>
```

displays a long file one screen full of lines at a time.

When the *more* command is in effect:

- Tap the *spacebar* to move from one screen of lines to the next.
- Press the *Enter* key to advance one line at a time.
- Tap the *q* key to quit before all the lines in the file have been displayed.

```
less <fn>
```

The *less* command may also be used to pause the display of a long file after each screenful, but this command allows you to scroll both **up** and **down** through the file when you use the up and down arrow keys ( and .

Copying a File

```
cp -i <fn> <newfn>
```

Where *<fn>* is the name of an existing file, and *<newfn>* is the name of the file into which you want to duplicate the contents of *<fn>* (the previously saved file).

Example: To copy the contents of the file called *oldfile* to a file called *newfile*, enter:

```
cp -i oldfile newfile
```

Be careful to watch the screen! If the target file to which you're copying already exists, this version of the copy command (with the *-i* option) will call up messages to warn you that the file already exists and then to ask if you want to get rid of the old contents before the copy process begins. If you reply *y* for *yes*, what used to be in the file will be replaced with whatever you just copied into it.

A response of *n* for *no* just aborts the copy so that nothing happens.

Adding the Contents of One File to Another

```
cat file1 >> file2
```

This adds the contents of *file1* to the end of *file2*. It doesn't change the contents of *file1*. Again, do this only with text files, not with compiled code.

Deleting a File

```
rm <fn>
```

The *rm* (remove) command deletes a file and frees its disk space.

You can use the command *rm -i <fn>* to display messages asking you to confirm deletions by explicitly pressing *y* or *n* for *yes* or *no*.

Printing

Using the Unix Printers in the ACRC

The Unix printers in the ACRC are *text printers exclusively*.

NEVER print or attempt to print:

- any part of a log-in session using the *FILE* menu *PRINT* option
- compiled code (like an a.out file or a [filename].class file)
- a script file with a *pico*, *vi*, or *pine* session in it
- a formatted text file (like a document in MS Word)
- a binary file (like an image, a graphic, or a browser page)
- a GUI interface (like printing from a Linux station)

`lp` The names of the Unix printers in the lab are `iclpr1` and `iclpr2`

If you want to print to one of the main Unix printers, you have to specify one of these names as the destination of your print job, using the Unix command `lp` with the option `-d` (for destination) followed by the name of the printer you want to use. If you want to print a file called `UnixFile1` on printer `iclpr1`, enter

`lp -d iclpr1 UnixFile1`

Note that the space used after `-d` is optional; the other spaces in the command line, after `lp` and after the name of the printer, are required.

`pr` The `lp` command prints the file without page breaks or interpretation of tab characters. In order to print page breaks and tabs, the `lp` command must be piped through the `pr` command with an `-e` option. Enter

`pr -e UnixFile1 | lp -d iclpr2`

`expand` To avoid lines printing beyond the righthand perforated edge of the paper, use `expand` to expand tabs, fold (or wrap) long lines, and paginate.

`expand Unixfile1 | fold | pr | lp -d iclpr1`

You might prefer the next command: it makes the text more readable by expanding tabs to only four characters (instead of the default eight characters) and by folding (or wrapping) the line on spaces between words instead of in the middle of words.

`expand -t 4 Unixfile1 | fold -s | pr | lp -d iclpr2`

Printing Long Jobs on Unix Printers

`wc` Unix printing in the ACRC is limited to 9 pages per job.

If you want to check the number of lines in the file to be printed, use the command `wc` with the line option:

`wc -l UnixFile1`

If the number of lines is greater than 594 (9 pages of 66 lines), then the file must be printed in smaller parts.

`head` If the file is 750 lines long, for instance, you won't be able to print the file in one job in the ACRC. However, you can print the first 500 lines using the `head` command if you enter

`head -500 UnixFile1 | lp -d iclpr1`

`tail` Then, you can print the last 250 lines using the `tail` command. Enter


`tail -250 UnixFile1 | lp -d iclpr2`

Using Laser Printers

ASCII text files created in your HP Unix account may be printed on laser printers *Pay1*, *Pay2*, or *Pay3* after the files have been downloaded to your Windows home directory.

To Download the File:

From the desktop, double-click the *SSH Secure Shell Client for FTP* icon (yellow in color). Please refer to the handout *Using SSH Secure Shell Client for FTP* to familiarize yourself with the details of its use.

- Login to your HP Unix account. (Click the *Quick Connect* button or select the *Quick Connect* option from the *File* menu.)
- Your Unix home directory should appear on the right (remote) side of the window. Move to the directory that has the file you want to download. Highlight the name of the file.
- Click the  button to download the file.
- Go to your Windows home directory. (Select *H:* from the *Look in:* menu.) Find the destination path for the file.
- Click the *Download* button.
- If the file is in (unformatted) ASCII text, rename the file so that it has a *.txt* extension.
- Select *Disconnect* from the *File* menu to log off your Unix account.
- Select *Exit* from the *File* menu to close the *SSH Secure Shell Client FTP*.

To Print the Downloaded File:

From the desktop, double-click the *My Computer* icon. Open the home directory for your Windows account (H:).

- Open the file you want to laser print.
 - Remember, unformatted text files must be renamed with a .txt extension.*
- Select the *Print* option from the *File* menu.
- In the *Print* window:
 - Select a printer from the *Print* window *Name* box.
 - Specify the page(s) you want to print.
 - Click the *OK* button. (Striking the *ENTER* key will *not* accept the selections.)
- From the *Print Control For The Desktop* window:
 - We recommend that you type your login ID in the *Client ID* box.
 - The *Client ID* does not have to be your login or user id, but your own unique login will not be confused with anyone else's name or nickname.
 - Please note **the number of pages** and the **total cost of printing** for this job.
 - Click the *Print* button.

Please refer to the *Laser Printing from Windows* handout for further assistance with printing, paying for, and retrieving laser output.

Finding Out About Your Account

Check Your Disk Quotas

The space that you are given to save files is limited, and so is the number of files that you are allowed to save. If you go over the allotted amount of space or the number of files, you won't be able to save anything new. At present, the limit is set at 50000 kilobytes and 4000 files (subject to change). It's good practice to use the `quota -v` command regularly to check how close you are to your limits. Delete files if it looks as though you might fill your disk space soon.

```
quota -v
```

Displays the following format (Your *usage* and *files* numbers will vary.):

File system	usage	quota	limit	timeleft	files	quota	limit	timeleft
/students	223	200000	200500		32	20000	20000	

Columns 2 and 3 show how many kilobytes you have stored and how many you are allowed. If you have exceeded your limit, the column labeled "timeleft" will indicate how long you have before the system administrator deletes some of your files.

The last 4 columns display similar statistics for the number of files you have saved.

Both numbers will be higher than you expect because there are hidden files that are necessary for Unix to configure programs like *mail* and your text editor. Don't delete these files or your Unix account might not work reliably.

The DISK LIMIT REACHED Message

If you get this error message when trying to perform an operation, you have bumped up against one of your quotas, and probably need to remove some files. (See item #2 on page 7.)

- Check your quotas with the `quota -v` command
- List your files and their sizes with `ll` or `ls -l`
 - To check on the contents of a file before deleting it, enter the command: `more <fn>`
- Any files you want to keep should be downloaded to diskette before they are removed from the Unix directory. See the separate handout on *Using SSH Secure Shell Client for FTP*.
- Delete files using the `rm` command.
- Retry the operation that gave you the DISK LIMIT REACHED error.

How You Reach Your Limits

You may encounter file space limitations in unexpected ways. Some commands create temporary files. An operation like compiling or linking a program creates new files, some of them temporary.

Any command that is trying to make new files may be aborted (stopped) due to your assigned space quota. File space may not seem to be a limitation when you check your quota because you are not seeing the temporary or other files that the operation is attempting to create.

If you get messages showing that an operation can't proceed, however, try deleting or downloading files and trying again. Please refer to the separate handout on *Using SSH Secure Shell Client for FTP*.

Scripting a Session

`script <fn>`

The `script` command captures (to the file that you name after the command) whatever comes up on the screen between the time you type in the `script` command and the time that you stop the process with `exit`. Uses include capturing the interactive input and the screen responses of one of your programs that writes to the screen and gets user input.

At times you will want to have a record of what has come up on the screen during part of the time you are in the lab.

Scripting procedure:

- Enter the `script` command, followed by a space and the name of the file to which you want to save the visual recording of the session.
- Start whatever you want to copy (a common example would be to display a program's source code onscreen with the `cat` command, and then compile and run the program).
- When you have finished, stop the scripting process with the `exit` command.
- You can then use the `lp` command to print the file which holds the results of the `script` command.

Warning: The `script` command does not always work the same way when you are running multiple processes, especially spawned shells.

`ps -u <your login id>`

1. This expression of the `process` command will remind you if you have one or more script sessions in progress.

PID	TTY	TIME	COMMAND
12158	ttyp5	0:00	bash
29275	pts/28	0:00	bash
12157	pts/28	0:00	script
12161	ttyp5	0:00	ps

If a `script` command is in progress, the command name `script` will appear in the command column on the right. For every `script` command displayed in the right-hand column, you will have to enter the `exit` command at the UNIX prompt in order to discontinue each of the scripts in progress.

2. This expression of the `process` command will also allow you to cancel a `script` command whose file has reached its disk limit.

If the DISK LIMIT REACHED message occurs when a `script` command is in progress, for example, your Unix connection will be 'stuck' in a `script` command it can neither continue nor complete.

The only way to break the command in progress is to login to your Unix account by way of another connection and cancel the `script` command in progress. (Connect by clicking the *Quick Connect* button in the upper left portion of the *SSH Secure Shell Client* window.)

Enter the command **`ps -u <your login ID>`** to check the processes currently running.

- Find the `ps` command in the command column.
- Look at the TTY # on the same line.
- Find the `bsh` in the command column that is NOT running on the same TTY#
- Note the PID for that `bash`.
- Use the command **`kill -9 <the PID #>`**

For the example shown above, the command would be: **`kill -9 29275`**

- Do *NOT* try to view or edit the extremely large file that the `script` command created; delete it: **`rm <the name of the script file whose disk limit was reached>`**

Samples Script Command Procedure (Using a C++ Program):`script hwk1`

Starts copying what will appear on the screen to a file named `hwk1` in your current directory

NEVER name the script with the same name as the C++ program.
The script file immediately destroys the program by *writing over* it.

`cat prg1.cpp`

Displays the contents of the C++ source file `prg1.cpp` on the screen. The script process sends everything that appears on the screen to the file `hwk1`

When `script` command is in progress, commands like `pine` or editors like `pico` or `vi` are prohibited because the changes these commands create in screen displays cannot be interpreted by the printers.

NEVER `cat` the script file while it is in progress.
It will duplicate itself in the script in progress and never reach the end of the script file.
Use `Ctrl/C` to break the `cat` command in progress.

`aCC prg1.cpp`

Compiles the source code in `prg1.cpp`

There are different arguments that your instructor might recommend.

Any error messages that appear on the screen would be captured to the `hwk1` file by the script process.

`a.out`

Runs the compiled C++ code. Any screen output sent by the program will be captured to the `hwk1` file by the script process.

NEVER `cat` an `a.out` file.
Object code is not a readable representation of your program.

`exit`

Ends the script (copying) process.

`lp -d iclpr2 hwk1`

Prints everything captured by the script process in the `hwk1` file on one of the lab's Unix printers.

Getting Help on a Unix Command`man <cmd>`

You can read the pages of the Unix manual on your screen by using the `man` command followed by a space and the name of the command you want. Entries are long and contain many options that you might not be interested in, but you can get a lot of information about the command and the system environment in this way. The manual pages are displayed with the `more` command, so you can:

- see the next page by pressing the *spacebar*
- quit manual command in progress and get back to the prompt by pressing `q`

`man -k <cmd>`

If you just want to find out whether there is a manual entry for a particular command, or if you want to see whether you have the correct spelling of a command, use this form of `man`. It gives a one-line response if the manual page for the command exists on the system, or an error message if it doesn't.