

Smith/FCAD Unix Cheat Sheet

James Lowenthal*

What is Unix?

Unix is an *operating system* that provides a set of commands to help operate the computer. When you open a terminal window (xterm, decterm, etc.) you will see the *unix prompt* (usually: %).

Where is my account, and how do I log on?

Your account is either (1) on the Smith College Novell network. Log on to any Smith Macintosh computer using the username/password given to you by the User Support office; (2) the class account, “ast337”, only Smith Astronomy computers; or (3) a personal account on one or more individual Smith Astronomy computers.

From the mac finder, go to applications > utilities and double-click on x11.app. An xterm window should appear

You may also have an account on **urania**, the Smith Astronomy student research computer. To access an account on urania from other computers, open an xterm window (on a Windows machine, you’ll need an X-window emulator such as cygwin or xwin32). In the xterm window type the “secure shell” login command:

```
ssh -Y urania.ast.smith.edu and login using your username and password
```

How does Unix recognize the location of my files?

Unix recognizes a file by its *pathname* and its *filename*. The pathname consists of all the directories leading to the file, and can be specified in several ways. The organization on the computer is like a tree, with branches, sub-branches, sub-sub-branches, etc. For example, for the file “filehum” in the directory “Documents” and subdirectory “proj2” of the user “sophia”,

- ⇒ the *filename* is filehum
 - ⇒ the *pathname* is /Users/sophia/Documents/proj2/
 - ⇒ the *fullname* is /Users/sophia/Documents/proj2/filehum
- [above examples for Macintosh; for linux, replace “/Users” with “/home”]

Here is some unix shorthand notation for specifying path names:

- current directory
- .. directory one level above the current one
- ~jlowenth jlowenth’s home directory
- * wildcard that can stand for any string of characters not including “/”
- ? wildcard that can stand for any single character not including “/”

Examples of unix commands *cd*, *ls*, and *cp* with different specifications for pathnames

- | | |
|-------------------------------|--|
| cd ~sophia /hw1 | change to sophia’s hw1 directory |
| ls | list the current directory |
| cd coords | change from there to sophia’s coords directory |
| ls *x | list files ending in x in the current directory |
| ls .. | list the directory above this one (i.e. hw1) |
| cd ../time | move to sophia’s time directory |
| cp /Users/ally/proj2/filehi . | copy ally’s filehi file to the current directory (Mac) |
| cp /home/ally/proj2/filehi . | copy ally’s filehi file to the current directory (linux) |
| cp filea ../coords | copy filea to the coords directory one dir level up |

* Adapted from Kim McLeod’s sheet at Wellesley College

Which Unix commands will get me started? (Words in *italics* are arguments)

Information:

whoami	--figure out who's account this is!
man <i>commandname</i>	--show me the manual (help) pages for <i>commandname</i> (e.g. <i>man ls</i>)
du	--check disk space used by this directory and its subdirectories
df	--check to see how much space is left on various disks
cat <i>filename</i>	--print a file to the screen all at once
history	--show me the commands I've recently typed
passwd	--let me change my password
page <i>filename</i>	--print a file to the screen, one page at a time (hit <i>q</i> to quit, <i>enter</i> to advance 1 line, <i>space</i> to advance 1 page, <i>?</i> for help)

Directories and files:

pwd	--print working directory (i.e. show me the path for the current dir)
ls <i>dirname</i>	--list the contents of the directory called <i>dirname</i>
cd <i>dirname</i>	--change to directory <i>dirname</i>
cp <i>source destination</i>	--copy a file (works for various combinations of files and directories)
mv <i>source destination</i>	--move a file (works for various combinations of files and directories)
rm <i>filename</i>	--remove (delete) a file
rm -r <i>dirname</i>	--careful! Delete a directory and all of its subdirectories ("recursive")
mkdir <i>dirname</i>	--make a new subdirectory called <i>dirname</i>
locate <i>string</i>	-- find file(s) with <i>string</i> in their name, anywhere on computer
grep <i>string filename*</i>	-- search in files called <i>filename*</i> for text string <i>string</i>

Printing:

lpr <i>filename</i>	--print a file to the printer
lpq	--look at the printer queue (will show jobnumber, owner, filename)
lprm <i>jobnumber</i>	--remove a job from the printer queue
enscript <i>filename</i>	--print a file to the printer in pretty or paper saving formats e.g. <i>enscript -2r filename</i> for two-to-a-page rotated
duplex <i>filename</i>	--print double-sided

Editors

emacs <i>filename</i>	--full screen editor
vi <i>filename</i>	--another full screen editor

How do I get fancy?

Redirect output with ">"

e.g. **ls *.txt > txtlist** --list files ending in ".txt" and write the output to a file called "txtlist"

Send output to a program with "|" (called "pipe")

e.g. **ps aux | grep jlowenth** --look at all current processes and search for jlowenth's

Use the command line editor to save typing

e.g. up arrow, down arrow, ... (exact keys will depend on your own setup and editor)

Use shortcuts to save typing

e.g. **^m** --repeat the last command I typed that started with "m"

Run a job in the background so that you can still type in your unix window while it is running

e.g. **ds9 &** --start up the program ds9 but leave my unix window functional