



Unix/Linux Cheat Sheet for CMU Students

SSH

\$ **ssh andrewid@unix.andrew.cmu.edu** - Connect to CMU's unix machines

File

\$ **ls** - List files in current directory

\$ **ls -l [subdirectory]** - List files in [subdirectory]

\$ **ls -a** - List all files including hidden files

\$ **cd** - Change directory to home

\$ **rm -r [dir]** - Remove directory [dir] & its contents

\$ **mv [file1] [file2]** - Rename or move [file1] to [file2]. [file1] will be deleted.

\$ **cp [file1] [file2]** - Copy [file1] to [file2]; creates [file2] if it didn't exist

\$ **cp -r [dir1] [dir2]** - Copy [dir1] to [dir2]; creates [dir2] if it didn't exist

\$ **cd [dir]** - Change directory to [dir]

\$ **pwd** - List current working directory

\$ **cat [file]** - Print the contents of a [file]

\$ **rm [file]** - Delete [file]

VIM (Text Editor)

\$ **vim file** - Open file in vim

i - Enter insert mode to edit file

:wq - Save and quit file

:w - Save file

:q - Quit file

/[pattern] - Search for [pattern] in file

n - Go to next search instance

? - Go to previous search instance

EMACS (Text Editor)

\$ **emacs file** - Open file in emacs

ctrl-x ctrl-s - Save file.

ctrl-x ctrl-c - Quit file.

ctrl-s [pattern] - Search for [pattern] in file; hit again to repeat search

Enter - end search

Process Management

\$ **jobs** - List programs suspended or running in the background and find job numbers

\$ **bg %[jobnumber]** - Make a suspended program resume execution in the background

\$ **fg %[jobnumber]** - Bring a program back to the foreground, resuming if suspended

\$ **[command] &** - Run a command in the background

\$ **kill %[jobnumber]** - Kill a program

\$ **ps x** - List all your programs

General

\$ **man [command]** - Open the manual page for [command]

\$ **ctrl-c** - Terminates the active program

\$ **ctrl-z** - Suspends the active program. Resume with **fg** or **bg**.

AFS

\$ **fs sa [andrewid] [permissions] [dir]** - Set permissions on directory [dir] for [andrewid]. [permissions] can be "read," "write," or "none."

\$ **fs la [dir]** - List permissions on directory [dir]

\$ **pts [creategroup] [andrewid.grouptitle]** - Create a new pts group with groupname [andrewid.grouptitle]. After running this, group can be used in place of an andrewid in permissions-setting commands.

\$ **pts adduser [andrewid] [groupname]** - Add a user to group [groupname].

\$ **pts removeuser [andrewid] [group name]** - Removes a user to group [groupname].

\$ **pts membership [group name]** - List the members of a group.