

## DUANE'S INCREDIBLY BRIEF INTRO TO UNIX

### How to get help:

man <command-name>      Get full description of unix command  
man -k <keyword>        List unix commands mentioning keyword in title

### Logging in and out:

logout                    Terminate session  
exit                      Terminate current "shell"  
ssh amerifax.cs.williams.edu    Start a remote terminal session on amerifax

### File manipulation:

emacs <file>            Edit a text file (see "cheat sheet")  
mv <old> <new>          Rename/move <old> file to a <new> name  
rm <file(s)>            Delete file(s) from system  
cp <orig> <duplicate>    Copy <orig> to file named <duplicate>  
cat <file>              Display/catenate file contents to screen  
more <file>             Display file, page by page (but: use less)  
less <file>             Display file, page by page (avoid more)  
head <file>             Display the first few lines of a file  
tail <file>             Display the last few lines of a file  
grep <pattern> <file(s)>    Search for/display pattern within file(s)  
source <file>            Read commands from <file> (also: . <file>)

### Directory manipulation:

cd <directory>          Change focus of session to files in directory  
ls                        List files in current directory  
mkdir <name>            Make a new subdirectory, called <name>  
rmdir <name>            Remove an empty subdirectory

### Git setup (once per new machine; or use configure in starter repo):

git config --global core.editor emacs      Make emacs the git editor  
git config --global push.default simple    Simplest pushing mechanism  
git config --global user.name "<your name>"    Register you as git maintainer  
git config --global user.email "your-email@williams.edu"    Register your email.

### Git workflow:

git clone <source> <dir>      Clone (entirely copy) repository into directory  
git pull                      Pull any commits from repo into local repo  
git push                      Push any local commits to repository  
git add <file>                Add a file as part of next commit  
git commit -m "<message>"    Commit changes to added files to local repo.

### Java (always use java versions 11 or greater):

javac Sample.java          Compile java class Sample into Sample.class  
java Sample                Run main from Sample class, in Sample.class  
javap Sample                Describe the interface for java Sample class  
javadoc -d doc Sample.java    Build documentation from java in directory doc

### Printing & Mail:

enscript <file>            Print a pretty copy of file in unix lab  
enscript -d lw-cs-217a <file>    Print a pretty copy of file to mac lab

### Information about users and systems:

w                          Who's on the system  
top                        What are top cpu processes  
ps                        List processes on this system  
whoami                    Who is logged in at this window  
uptime                    Print stats on machine, also time since boot

### Web:

http://www.cs.williams.edu      CS home page  
http://www.cs.williams.edu/~cs136    CS136 home page

Names of CS unix boxes in TCL 312 (ssh to these from on-campus only!):  
amerifax bagual barzona brava charolais devon galloway guernsey kuri  
lidianiata panda rathi reina sharabi sind siri sykia tundaca zebu

### Compute servers (available from off campus):

lohani limia deoni

## Duane's Ten Ways To Make Your Unix Life More Reasonable

0. Walk away from the machine. Don't waste your time in front of a machine if you're not making any progress. Print a listing and walk away. Make and take a friend with you. Life will be better if you reconsider the situation without the pressure of a computer.

1. Read the man pages.  
Realize, if you haven't already, that you don't know everything. Learn. The world travels about 66,600 miles an hour about our Sun, and the Sun glides gracefully along its own path dragging us along. Hackers have no impact. None.
2. Learn the emacs keystrokes. It will save you when you have to use a system whose mouse is not working. Avoid the "arrow keys". Why?...ask Darwin.
3. Use emacs keystrokes in the shell. Many cursor manipulation keystrokes from emacs recall history in the "bash" shell:  
^P = previous command, ^N = next command,  
^R = search for command from the past by typing a few letters  
^A = go to beginning of command line  
^E = go to end of command line  
^B = go back one character  
^F = go forward one character  
^D = delete this character  
<del> = delete previous character  
^Y = yank cut text  
^\_ = undo  
Most of these commands work in most Mac applications, including TextEdit.

4. Learn about your environment. Shells like "bash" have survived evolution by helping their users do complex things. Type:  
man bash  
Good things to keep an eye out for are "aliases" and "shell scripts". Other things to read about: find, tar, awk, re\_format.
5. Stay organized.  
Create directories to organize your belongings. Delete temporary files that you no longer need. Besides taking up space, they add friction to your life.
6. Use a version control system, like git.  
It's very easy to get started using a version control system. Making frequent use of commits will keep you from replacing good code with bad. Follow the workflow every session: pull, add, commit, push.
7. Use the facilities we provide.  
Using our labs allows us to help you if you have problems. They're also a good place to meet others that are suffering the same project. Leaving your room to do your work makes it a nicer place to return.
8. Practice. Yes, even more.
9. Write. Good writing is hard, and computer scientists write far too little real prose. A good, small place to start: comments on your code. Another place: write your name on everything you do. If it's really yours, copyright it (it's free)!

(c) 2000-2021 duane a. bailey  
See?

# DUANE'S INCREDIBLY BRIEF INTRO TO EMACS

C-z means: hold Control and z at same time.

M-x means: type escape \*then\* x, or: Meta \*and\* x.

## Starting Emacs

start emacs            emacs

## Exiting Emacs

Suspend emacs        C-z  
exit emacs            C-x C-c

## Files

read file            C-x C-f  
visit file other windw C-x C-v  
save file            C-x C-s  
insert file          C-x i  
write buffer to file C-x C-w

## Getting help

first time users      C-h t  
second time users     C-h ?  
help on keystroke     C-h k  
help on function      C-h f  
man page              M-x manual-entry

## Error recovery

abort command        C-g  
recover lost file     M-x recover-file  
undo                  C-\_   
restore buffer        M-x revert-buffer  
redraw screen        C-l

## Motion

Entity	back	forth
character	C-b	C-f
word	M-b	M-f
line	C-p	C-n
end of line	C-a	C-e
sentence	M-a	M-e
buffer	M-<	M->
screen	M-v	C-v

## Marking (building regions)

set mark            C-spc  
exchange point & mark C-x x  
mark buffer        C-x h

## Registers

copy region to reg    C-x x  
get region from reg   C-x g

## Killing and Deleting

Entity	back	forth
character	Delete	C-d
word	M-Del	M-d
end of line	M-0 C-k	C-k
sent	C-x Del	M-k
region	C-w	
yank back	C-y	
zap to <char>	M-z <char>	

## Transpose

characters            C-t  
words                M-t  
lines                C-x C-t

## Searching

forward	C-s
backward/reverse	C-r
forward expression	C-M-s
reverse expression	C-M-r
exit search	Return
undo last search char	Delete
abort search	C-g

## Query replace

start query replace	M-%
query replace word	C-u M-%
Within query replace...	
replace & search	Space
replace & stay here	,
backup to prev. match	^
don't replace, go on	Delete
replace remaining	!
exit	Return

## Multiple Windows

keep just this window	C-x 1
split window	C-x 2
switch to other window	C-x o

## Buffers

select another buffer	C-x b
list other buffers	C-x C-b
kill this buffer	C-x k
minibuffer	M-x

## The minibuffer

complete	Tab
show completions	?
complete and execute	Return
previous input	M-p
next input	M-n
abort	C-g

## Keyboard Macros

start defining	C-x (
stop defining	C-x )
execute macro	C-x e

## Compiling something

Compile window	M-x compile
(e.g. gcc -o x x.c)	
Find next error	C-x `

## Binding keys

To make control-x-' compile, add  
(global-set-key "\C-x'" 'compile)  
to ~/.emacs (or create file, as necessary)

## Things you should never know about

dungeon	M-x dunnet
tetris	M-x tetris
hide & seek	M-x blackbox
psychotherapy	M-x doctor
gomoku	M-x gomoku
robot game	M-x landmark
pong	M-x pong
the snake game	M-x snake
peg solitaire	M-x solitaire