Linux Command Line Basics

… it’s really not that bad

Personal Information

- BS, Computer Science
- MS, Computer Science, University of Tennessee -- Knoxville
- Worked jointly with University of Tennessee and Oak Ridge National Lab
- Three (3) years at University of Tennessee
- Seven (7) years at Iowa State University
- Unix/Linux aficionado

Course Goals

- Become comfortable using a bash shell
- Access remote Linux computers
- Learn about available campus resources
- Run commands in the bash shell
- Use output redirection
- Discover scripting languages and why we use them

Course Aims

- Not to be boring
- Not to explain Operating System Theory
- Not to show Advanced Techniques for using Linux
- Not to be elitist
- Not to be unattainable

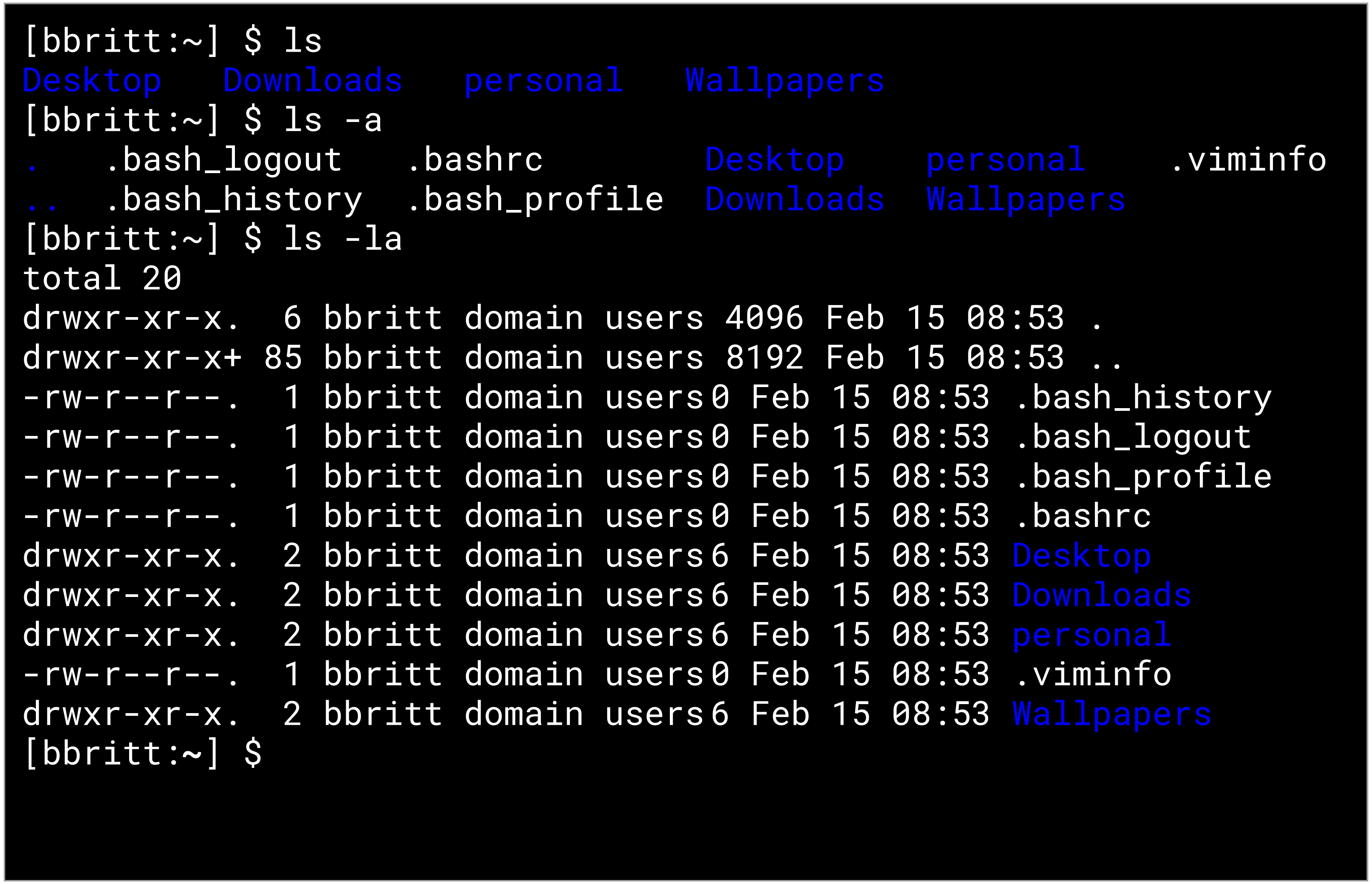
Course Identity

- Fun, or at least enjoyable
- Basic, everyday Linuxy-type stuff
- Simple
- Specific to Linux (not necessarily Windows or Mac OS X)
- A way to learn something new

Terminology

- **If you don’t understand something, please ask!!!**
- Computer Scientists use funny words sometimes:
  - Strings
  - Characters
  - Array
  - Terminal

What is the scariest thing for new Linux users?

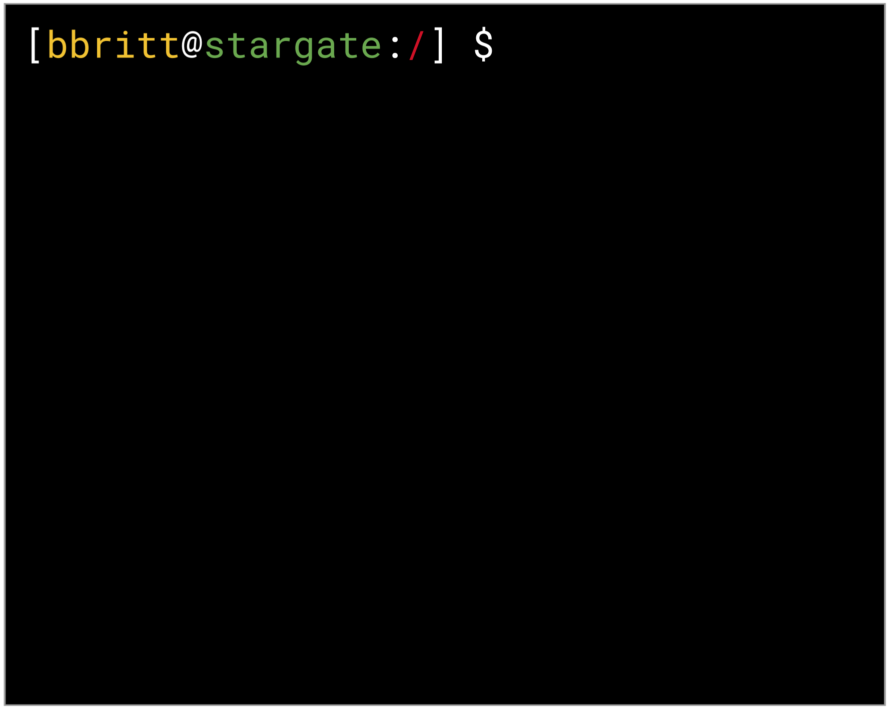


```
[bbritt:~] $ ls
Desktop   Downloads   personal   Wallpapers
[bbritt:~] $ ls -a
.    .bash_logout   .bashrc         Desktop    personal    .viminfo
..   .bash_history  .bash_profile   Downloads  Wallpapers
[bbritt:~] $ ls -la
total 20
drwxr-xr-x.   6 bbritt domain users 4096 Feb 15 08:53 .
drwxr-xr-x+ 85 bbritt domain users 8192 Feb 15 08:53 ..
-rw-r--r--.   1 bbritt domain users 0 Feb 15 08:53 .bash_history
-rw-r--r--.   1 bbritt domain users 0 Feb 15 08:53 .bash_logout
-rw-r--r--.   1 bbritt domain users 0 Feb 15 08:53 .bash_profile
-rw-r--r--.   1 bbritt domain users 0 Feb 15 08:53 .bashrc
drwxr-xr-x.   2 bbritt domain users 6 Feb 15 08:53 Desktop
drwxr-xr-x.   2 bbritt domain users 6 Feb 15 08:53 Downloads
drwxr-xr-x.   2 bbritt domain users 6 Feb 15 08:53 personal
-rw-r--r--.   1 bbritt domain users 0 Feb 15 08:53 .viminfo
drwxr-xr-x.   2 bbritt domain users 6 Feb 15 08:53 Wallpapers
[bbritt:~] $
```

Help, what do I do with this stuff???

What does it mean?

- User Name
- Computer Name
- Directory Name



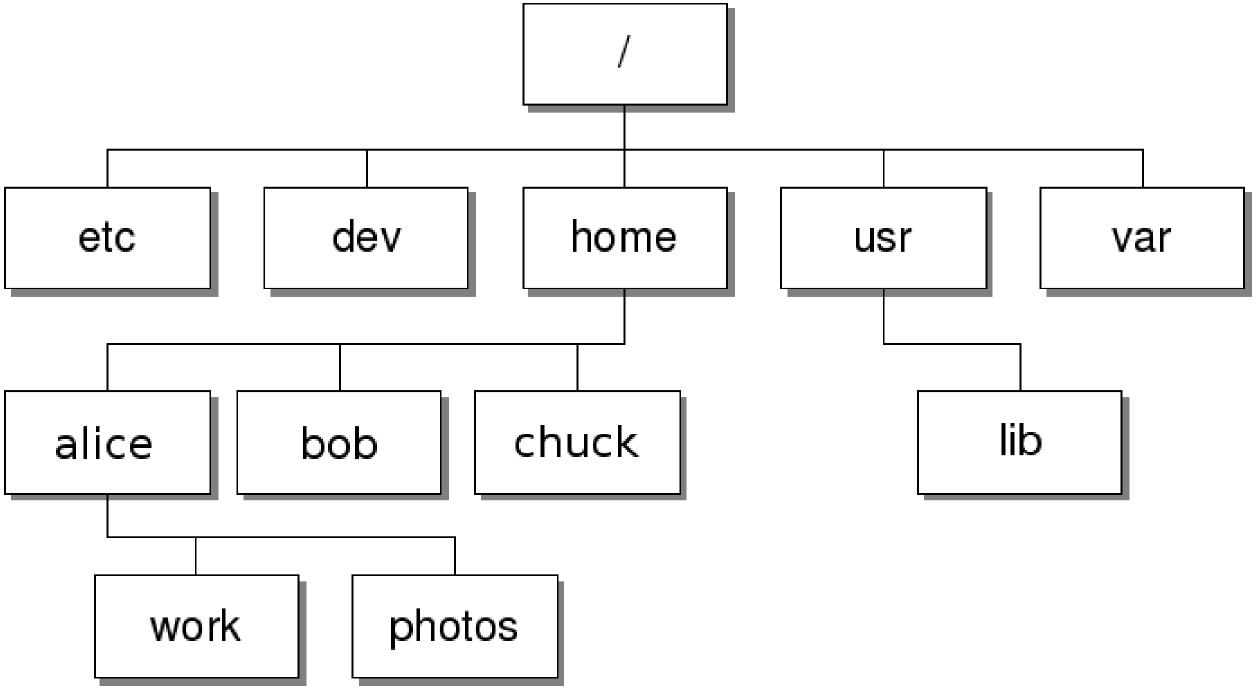
```
[bbritt@stargate:/] $
```

So what do we do?

- First, we need to understand the file and directory structure.
- Also, how do we navigate that structure?

So what do we do?

- First, we need to understand the file and directory structure.
- Also, how do we navigate that structure?
- Hint, it’s very similar to something you know



It's just like Windows computers, but the folders are named differently.

File and Directory Structure

Directory	Purpose	Windows
/	“root” of the file system	C:\
/dev	Devices attached to the computer	- Hidden in Windows -
/etc	Configuration files for services	Typically in C:\Program Files
/home	User home directories	C:\Users
/usr	Programs	C:\Program Files
/usr/bin	Executable files	C:\Program Files\<program>
/usr/sbin	Executable files for system tasks	C:\Windows\System32
/usr/share	Documentation	Typically in C:\Program Files or online
/var	Logs, “variable” data like databases	C:\Windows\System32\winevt\Logs C:\Program Files\<program>

File and Directory Structure

Directory	Purpose	What we care about
/	"root" of the file system	
/dev	Devices attached to the computer	
/etc	Configuration files for services	
/home	User home directories	Files go here!
/usr	Programs	
/usr/bin	Executable files	Programs go here
/usr/sbin	Executable files for system tasks	
/usr/share	Documentation	Documentation is good :)
/var	Logs, "variable" data like databases	

File / Directory Structure

- There are other directories that are important for more advanced stuff.
- We may cover those in a different class

Common Commands

Navigation:

- cd
- ls
- pwd
- whoami

Help:

- man / apropos

System Information

- ps
- top

Searching:

- grep
- find
- locate

File Manipulation:

- cat
- echo
- mkdir
- rmdir
- mv
- cp
- rm
- nano

Navigation

- pwd
- ls
- cd

Case Sensitivity

- All commands in Linux are CASE-SENSITIVE!
- This means:
  - LS is not the same as ls
  - PWD is not the same as pwd
  - etc...

pwd

- Print out my current location in the file tree.

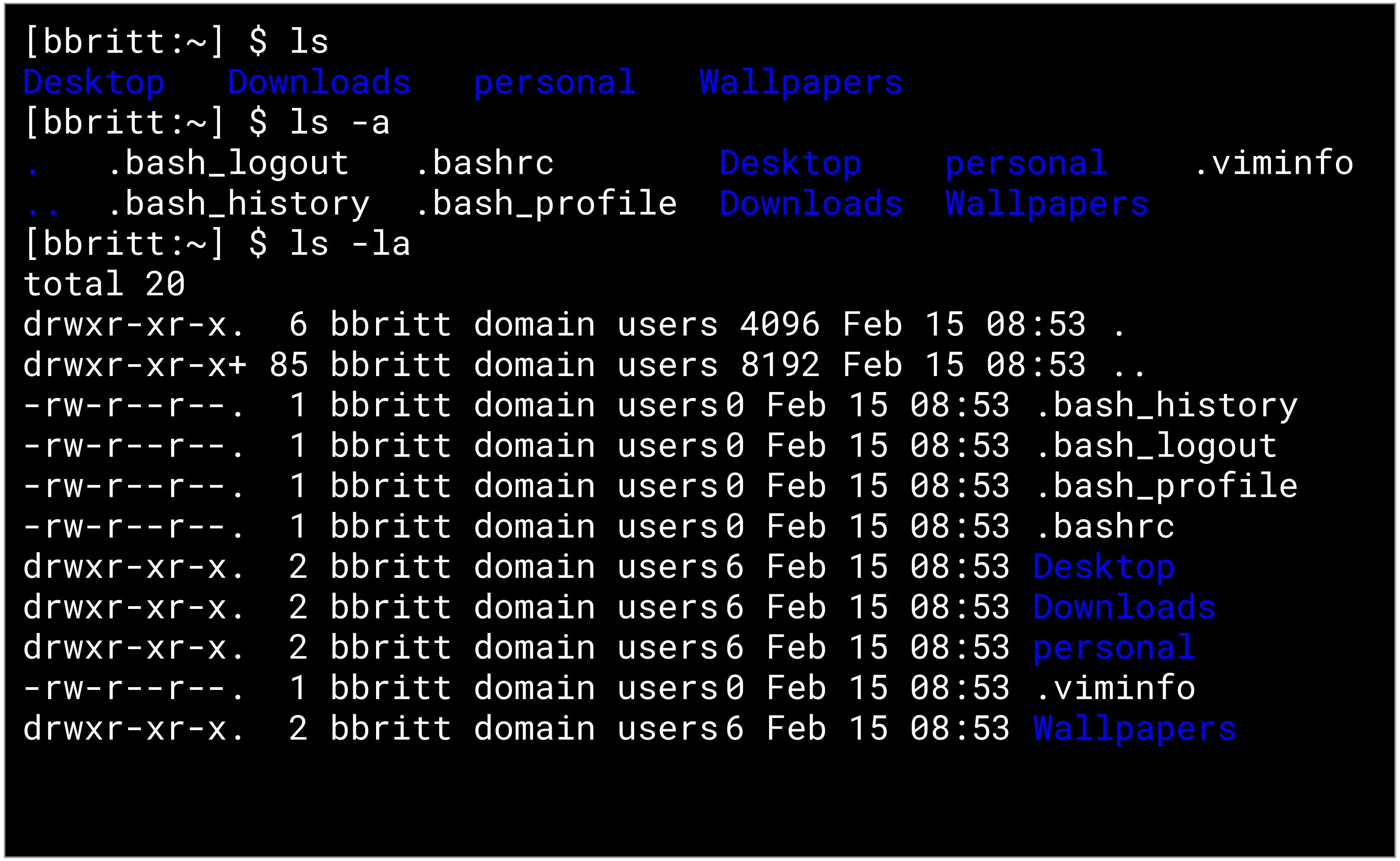
```
[bbritt:~] $ pwd
/home/bbritt
[bbritt:~] $
```

ls

- Stands for ‘list’
- Print out the files in a directory (current working directory if not specified)

Command	Effect
ls	List the contents of your current working directory.
ls /path/to/dir	List the contents of the dir directory
ls -h	List the contents of the current working directory with “human readable” sizes
ls ~	List the contents of your home directory
ls -a	List the contents of the current working directory, including hidden files
ls -l	List the contents of your current working directory in long format
ls -p	List the contents of your current working directory, but put slashes after directory names

Common LS command-line options



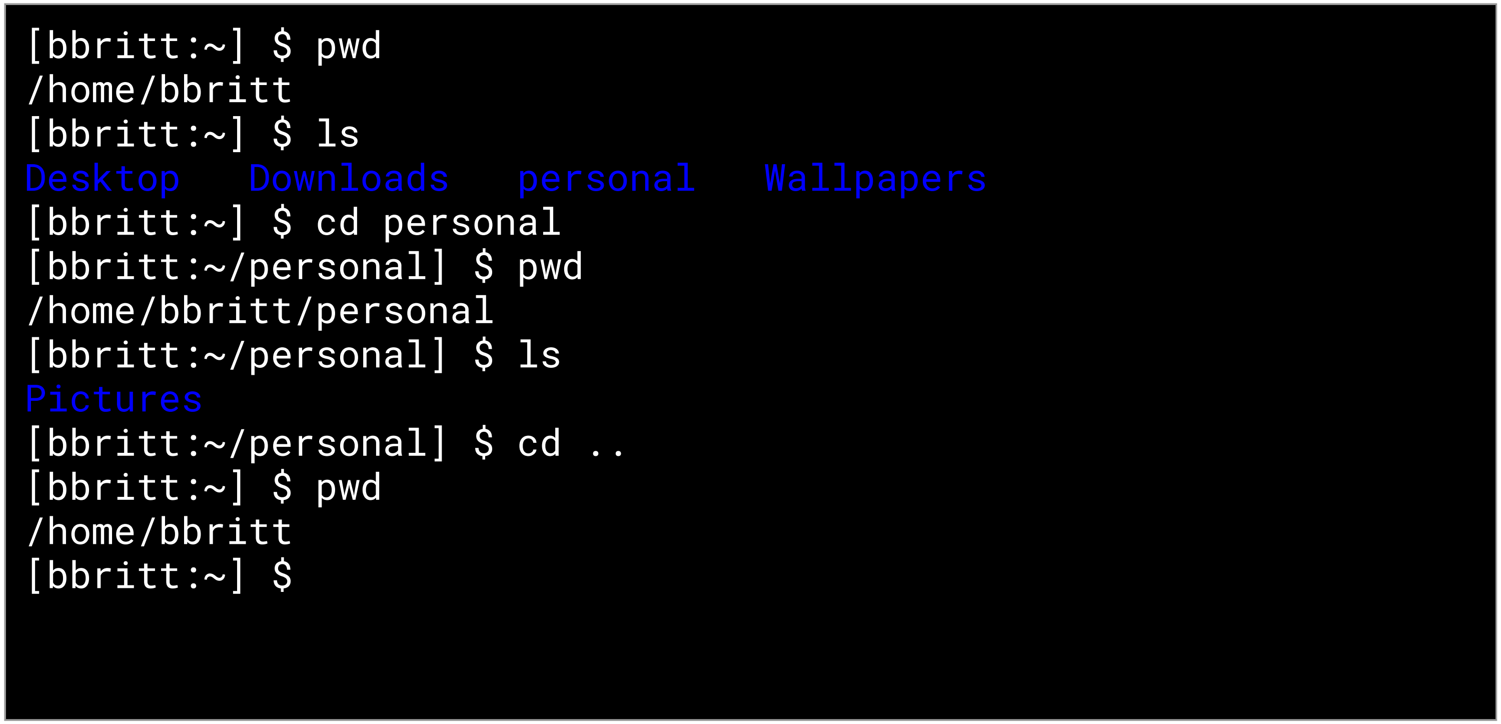
```
[bbritt:~] $ ls
Desktop   Downloads   personal   Wallpapers
[bbritt:~] $ ls -a
.    .bash_logout   .bashrc        Desktop    personal    .viminfo
..   .bash_history  .bash_profile  Downloads  Wallpapers
[bbritt:~] $ ls -la
total 20
drwxr-xr-x.  6 bbritt domain users 4096 Feb 15 08:53 .
drwxr-xr-x+ 85 bbritt domain users 8192 Feb 15 08:53 ..
-rw-r--r--.  1 bbritt domain users0 Feb 15 08:53 .bash_history
-rw-r--r--.  1 bbritt domain users0 Feb 15 08:53 .bash_logout
-rw-r--r--.  1 bbritt domain users0 Feb 15 08:53 .bash_profile
-rw-r--r--.  1 bbritt domain users0 Feb 15 08:53 .bashrc
drwxr-xr-x.  2 bbritt domain users6 Feb 15 08:53 Desktop
drwxr-xr-x.  2 bbritt domain users6 Feb 15 08:53 Downloads
drwxr-xr-x.  2 bbritt domain users6 Feb 15 08:53 personal
-rw-r--r--.  1 bbritt domain users0 Feb 15 08:53 .viminfo
drwxr-xr-x.  2 bbritt domain users6 Feb 15 08:53 Wallpapers
```

The LS Command

cd

- Stands for ‘change directory’.
- Alters your current working directory context.
- In other words, it moves you around!

Command	***Effect***
cd	***Change to your home directory (/home/username)***
cd ~	***Change to your home directory (/home/username)***
cd ~/Pictures	***Change to /home/username/Pictures***
cd /path/to/foo	***Change to the foo directory***
cd ..	***Change to the parent directory (i.e., move up one directory)***



```
[bbritt:~] $ pwd
/home/bbritt
[bbritt:~] $ ls
Desktop   Downloads   personal   Wallpapers
[bbritt:~] $ cd personal
[bbritt:~/personal] $ pwd
/home/bbritt/personal
[bbritt:~/personal] $ ls
Pictures
[bbritt:~/personal] $ cd ..
[bbritt:~] $ pwd
/home/bbritt
[bbritt:~] $
```

The CD Command

Tips

- The '**~**' (tilde) character, when used in a PATH, always means “my home directory”. This is a **very common** shortcut.
- Use the arrow keys:

  If you type something in wrong, you can use the left and right arrow keys (on your keyboard) to move the cursor and fix your mistakes!
- If you press the up arrow, you can see previous commands that you have typed, starting with the most recent.

Useful *expansions*

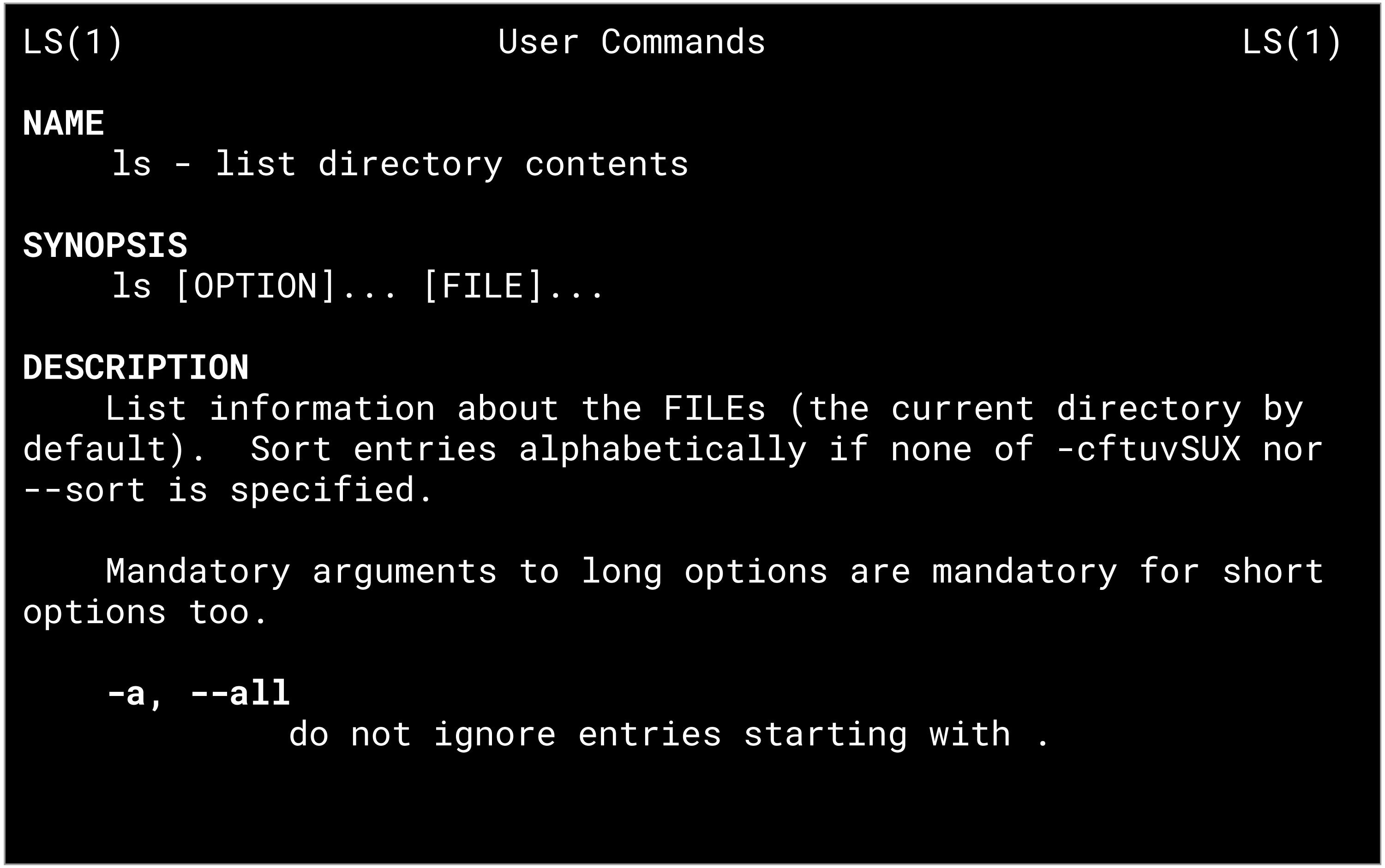
String	Meaning	Example
*****	Match everything!	**_file*_** would match **file1**, **file2**, **filename**, **files**, **filed**, **filer**, **filet** but it would not match **username.txt** ...
?	Match any 1 character	**_fi?e1_** would match **fire1**, and **file1**, but it would not match **file2**
[Aa]	Match one of the included letters	**[Aa]file** would match Afile and afile but not Thisfile or Thatfile

Help

- man

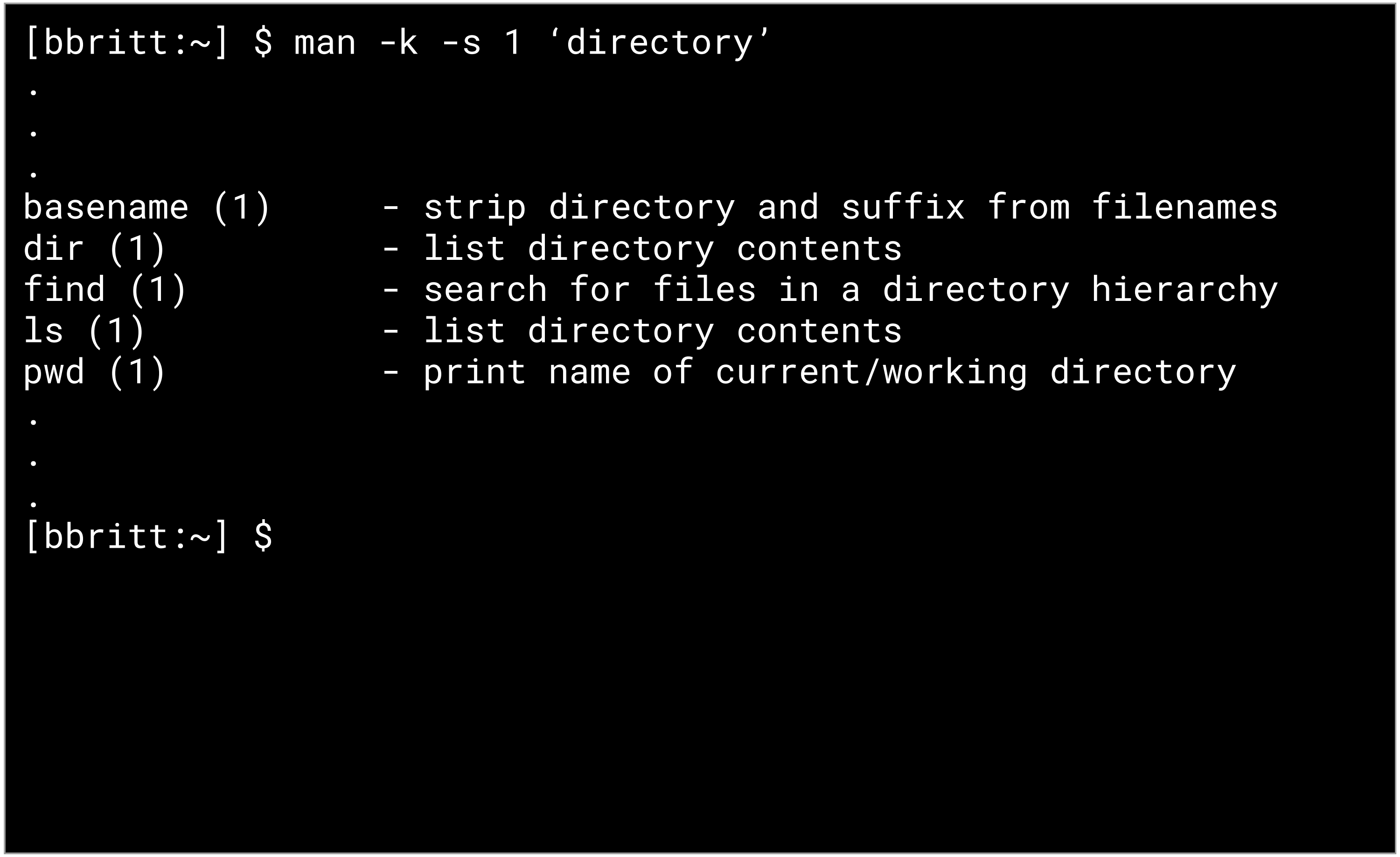
man & apropos

- Software documentation in your terminal application!
- Also mostly available online: https://www.kernel.org/doc/man-pages/
- Details how to use a program
- Can search through man pages to find the topic you want.
  - Use ‘apropos’ or ‘man -k’ to search
  - Use the ***-s*** option to limit your search to certain sections. You will typically only want to search **Section 1**



```
LS(1)                     User Commands                    LS(1)

NAME
    ls - list directory contents

SYNOPSIS
    ls [OPTION]... [FILE]...

DESCRIPTION
    List information about the FILEs (the current directory by
default).  Sort entries alphabetically if none of -cftuvSUX nor
--sort is specified.

    Mandatory arguments to long options are mandatory for short
options too.

    -a, --all
            do not ignore entries starting with .
```

Example man page for the LS command



```
[bbritt:~] $ man -k -s 1 ‘directory’
.
.
.
basename (1)     - strip directory and suffix from filenames
dir (1)          - list directory contents
find (1)         - search for files in a directory hierarchy
ls (1)           - list directory contents
pwd (1)          - print name of current/working directory
.
.
.
[bbritt:~] $
```

Searching through man pages

System Information

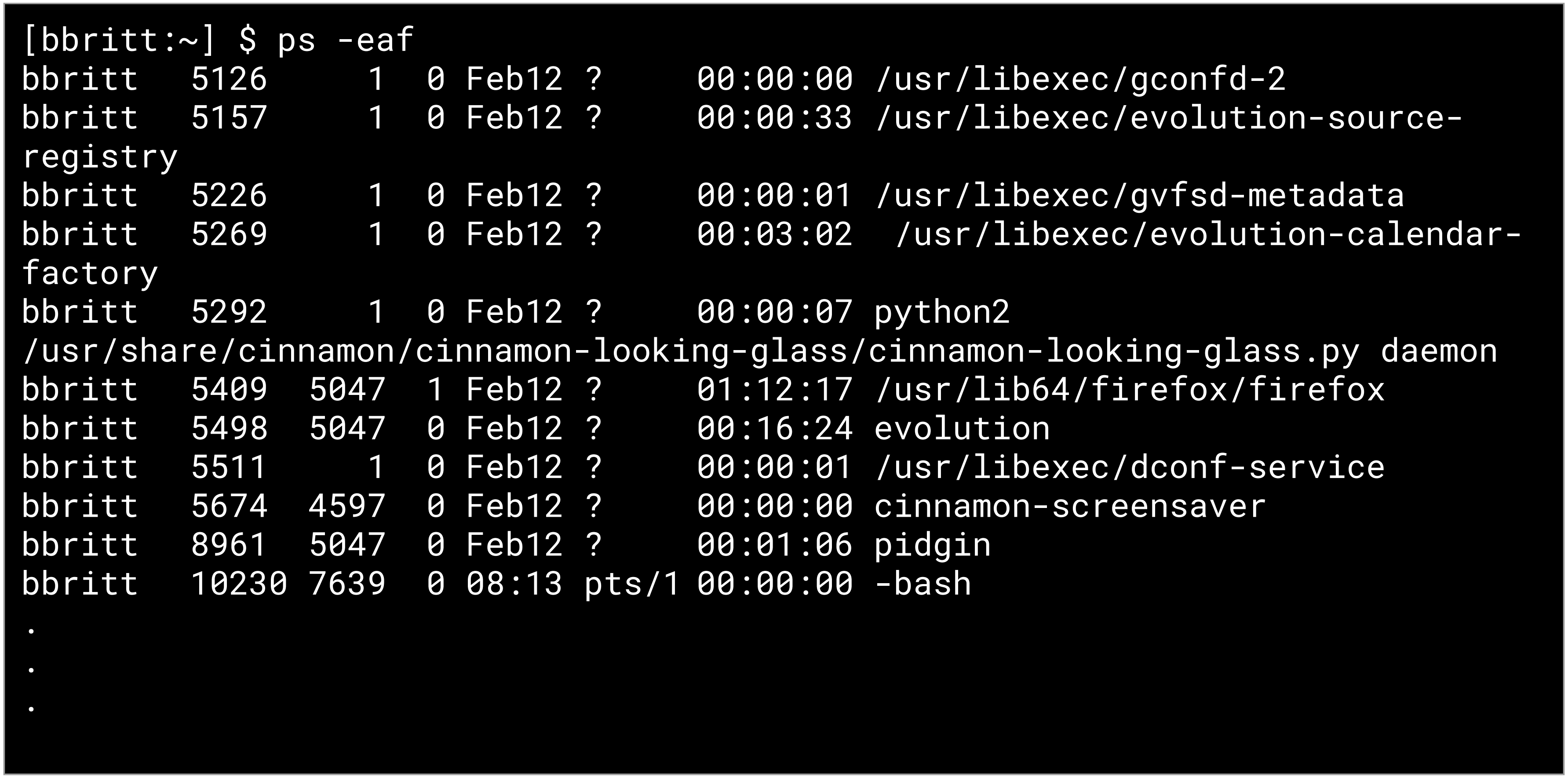
- ps
- top
- who

ps

- List Processes
  - **Processes**: programs that are executing
- Like the **ls** command, this one can take a LOT of options.
- You could call this **ls** for processes.
- My personal favorite:

  ps -eaf

  Means:
  - **-ea** : show me all the processes
  - **-f** : list it with all the extra columns

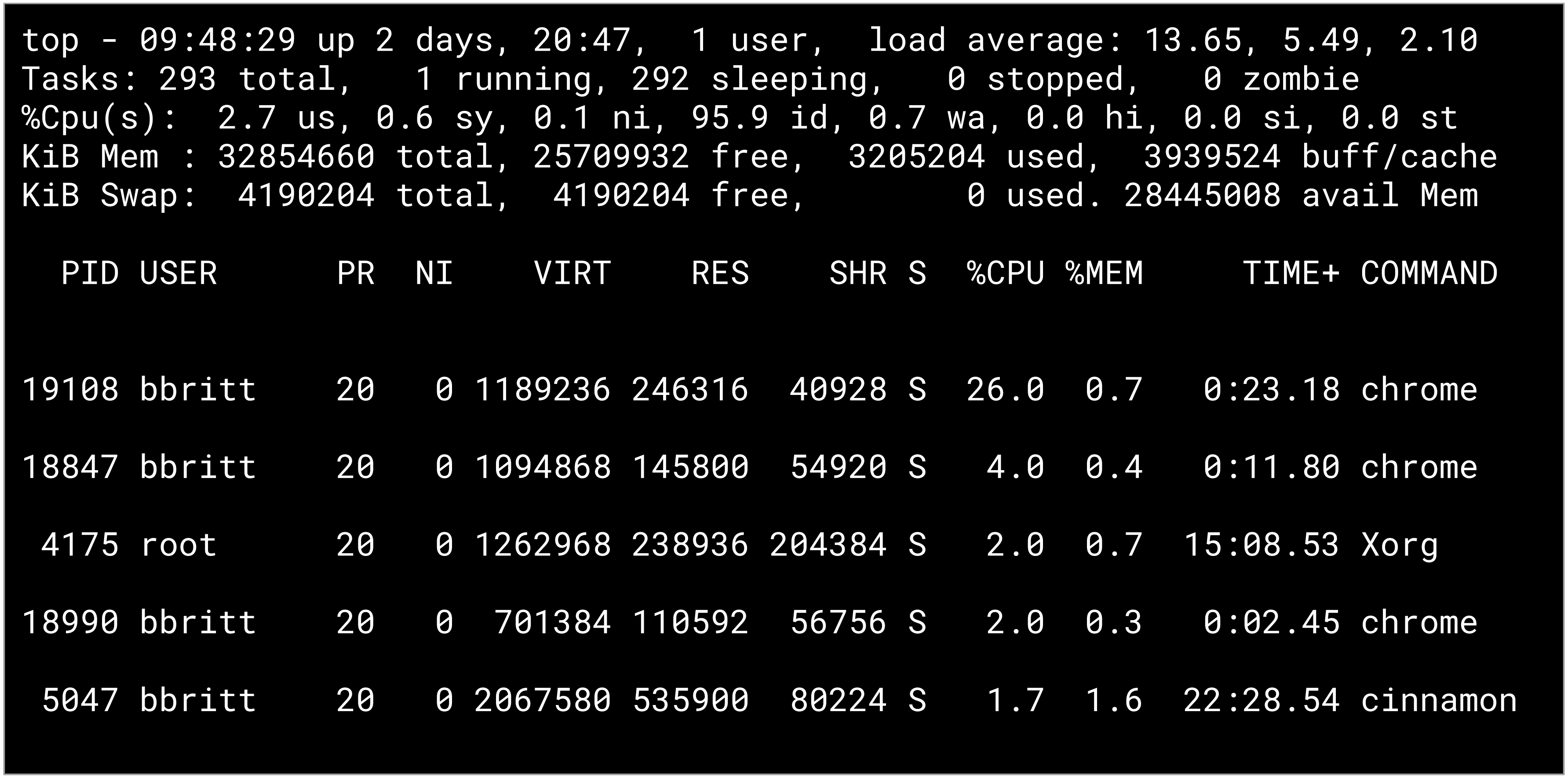


```
[bbritt:~] $ ps -eaf
bbritt   5126      1  0 Feb12 ?     00:00:00 /usr/libexec/gconfd-2
bbritt   5157      1  0 Feb12 ?     00:00:33 /usr/libexec/evolution-source-
registry
bbritt   5226      1  0 Feb12 ?     00:00:01 /usr/libexec/gvfsd-metadata
bbritt   5269      1  0 Feb12 ?     00:03:02  /usr/libexec/evolution-calendar-
factory
bbritt   5292      1  0 Feb12 ?     00:00:07 python2
/usr/share/cinnamon/cinnamon-looking-glass/cinnamon-looking-glass.py daemon
bbritt   5409   5047  1 Feb12 ?     01:12:17 /usr/lib64/firefox/firefox
bbritt   5498   5047  0 Feb12 ?     00:16:24 evolution
bbritt   5511      1  0 Feb12 ?     00:00:01 /usr/libexec/dconf-service
bbritt   5674   4597  0 Feb12 ?     00:00:00 cinnamon-screensaver
bbritt   8961   5047  0 Feb12 ?     00:01:06 pidgin
bbritt   10230 7639   0 08:13 pts/1 00:00:00 -bash
.
.
.
```

ps example

top

- Shows:
  - a dynamic, real-time view of the running system.
  - processor usage
  - memory usage
  - total running time
  - total computer “load”
  - amount of free memory.
- You can also press the number **1** to show all CPU cores
- Can sort and filter output.

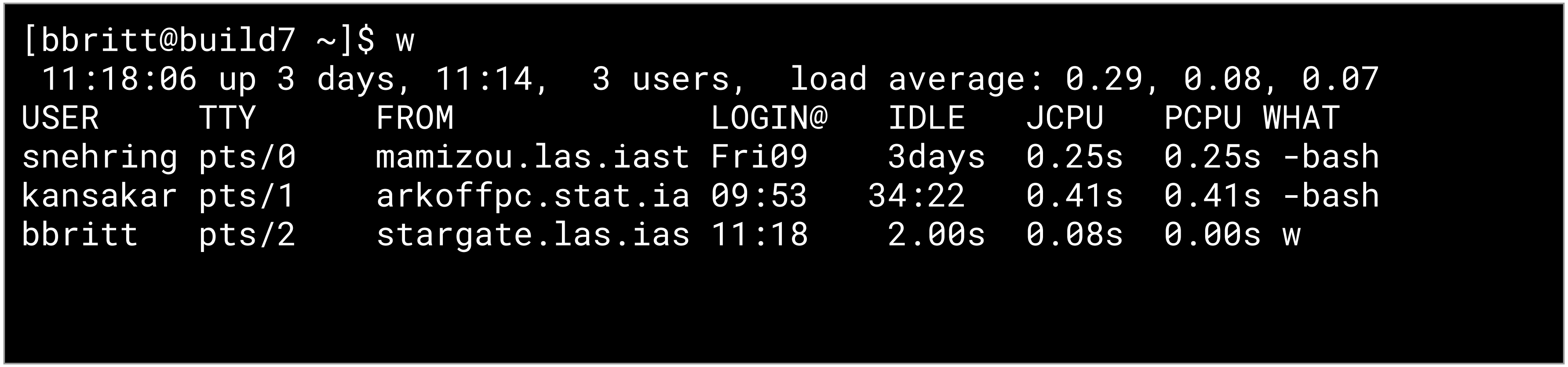


```
top - 09:48:29 up 2 days, 20:47,  1 user,  load average: 13.65, 5.49, 2.10
Tasks: 293 total,   1 running, 292 sleeping,   0 stopped,   0 zombie
%Cpu(s):  2.7 us, 0.6 sy, 0.1 ni, 95.9 id, 0.7 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 32854660 total, 25709932 free,  3205204 used,  3939524 buff/cache
KiB Swap:  4190204 total,  4190204 free,        0 used. 28445008 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND


19108 bbritt    20   0 1189236 246316  40928 S  26.0  0.7   0:23.18 chrome

18847 bbritt    20   0 1094868 145800  54920 S   4.0  0.4   0:11.80 chrome

 4175 root      20   0 1262968 238936 204384 S   2.0  0.7  15:08.53 Xorg

18990 bbritt    20   0  701384 110592  56756 S   2.0  0.3   0:02.45 chrome

 5047 bbritt    20   0 2067580 535900  80224 S   1.7  1.6  22:28.54 cinnamon
```

top example

who / w

- Shows very basic system information
  - time of last system boot
  - list of logged-in users
  - what processes the users are running
- Also shows system load



```
[bbritt@build7 ~]$ w
 11:18:06 up 3 days, 11:14,  3 users,  load average: 0.29, 0.08, 0.07
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
snehring pts/0    mamizou.las.iast Fri09    3days  0.25s  0.25s -bash
kansakar pts/1    arkoffpc.stat.ia 09:53   34:22   0.41s  0.41s -bash
bbritt   pts/2    stargate.las.ias 11:18    2.00s  0.08s  0.00s w
```

Example for the ‘w’ command

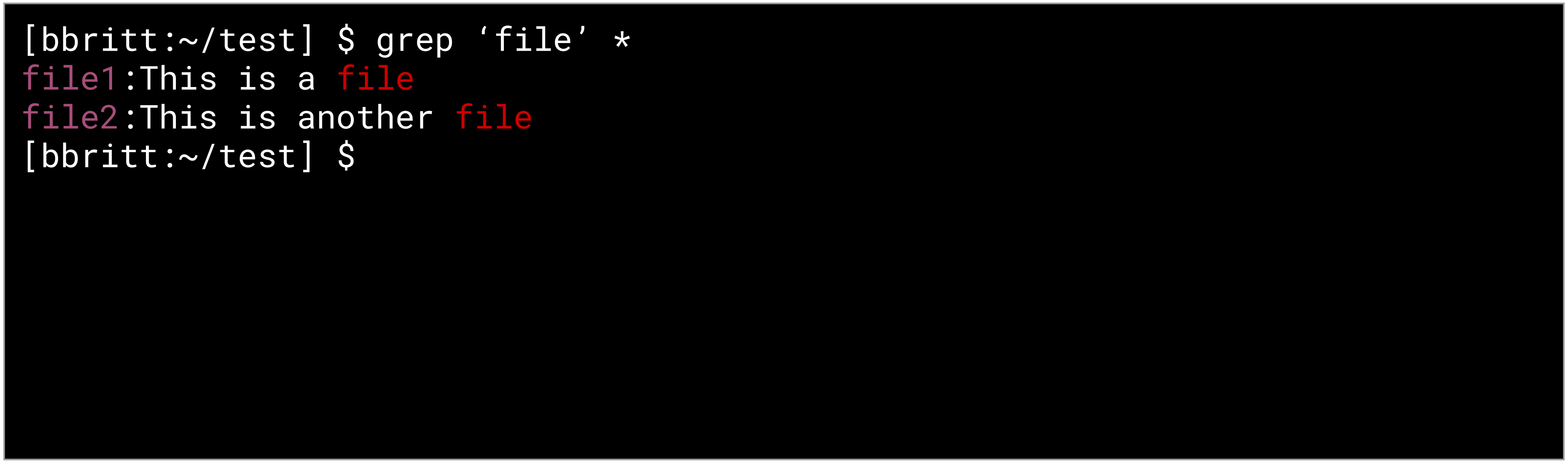
Searching

- grep
- find
- locate

grep

- Find information within files.
- Usage:
    - **grep** *‘word’* filename
    - **grep** *‘word’* file1 file2 … filen
    - **grep** *‘string1 string2’* filename
    - **<command>** | grep *‘something’*
- This is one of the most useful commands we go over today.

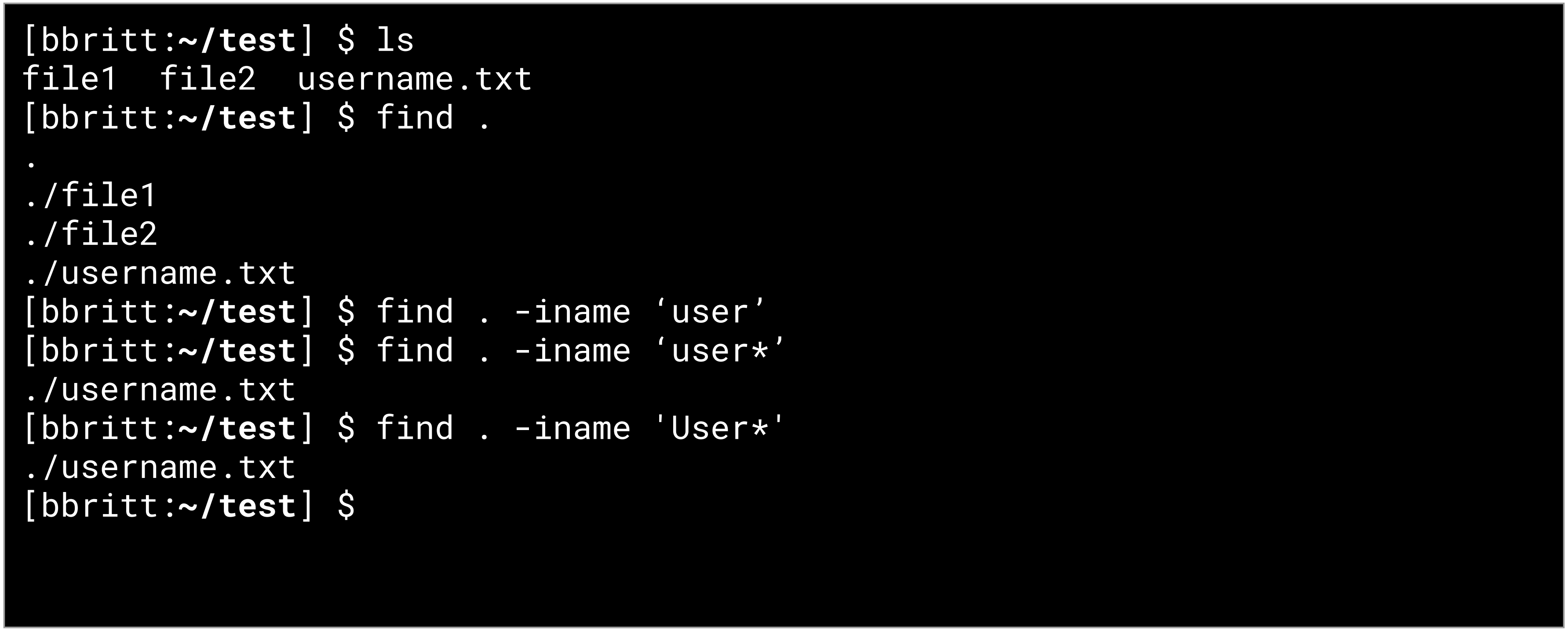
Grep example



```
[bbritt:~/test] $ grep ‘file’ *
file1:This is a file
file2:This is another file
[bbritt:~/test] $
```

find

- Find files by their attributes (filename, etc…).
- Usage:
  - **find** <location>
  - **find** <location> -iname ‘name’
  - … a lot more, also
- This is also one of the more useful commands we go over today.
- *-iname* means ‘case insensitive’ match, which is also useful.

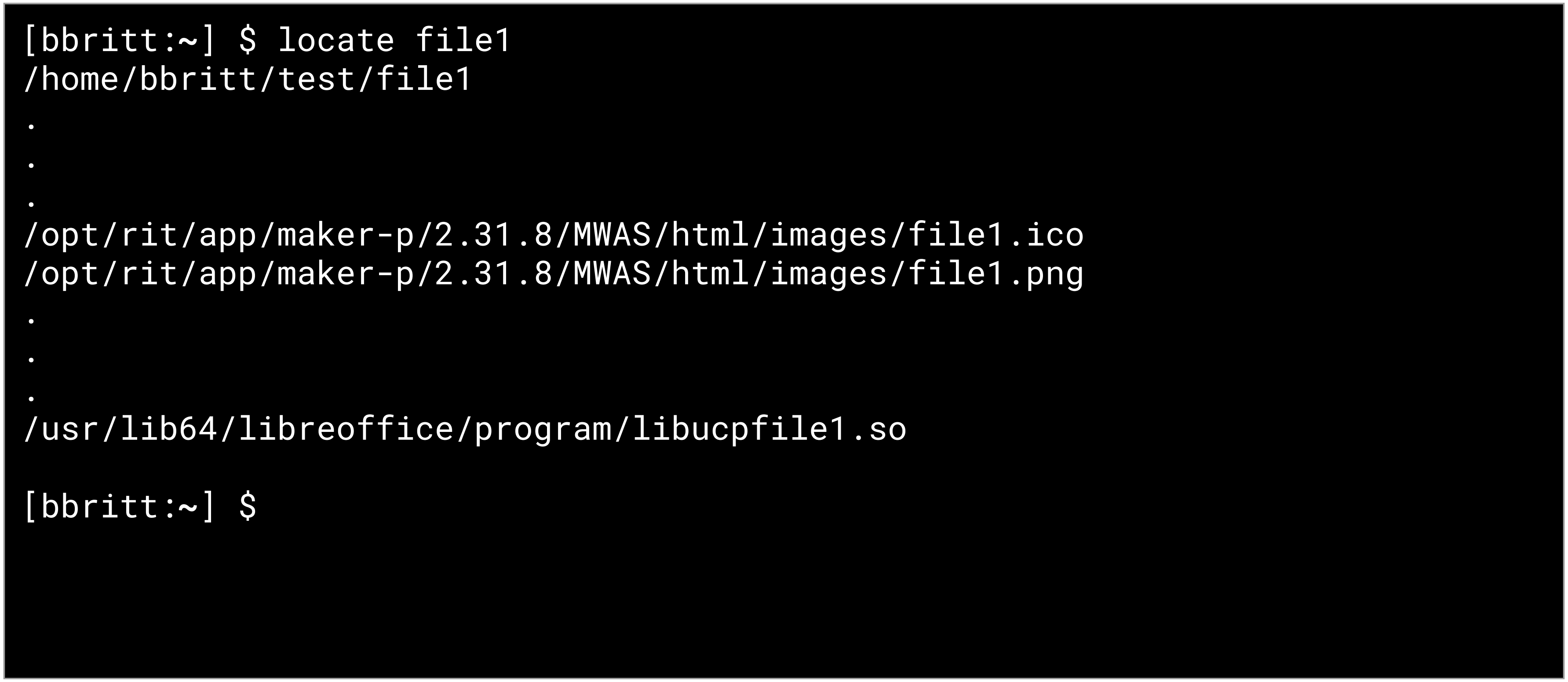


```
[bbritt:~/test] $ ls
file1  file2  username.txt
[bbritt:~/test] $ find .
.
./file1
./file2
./username.txt
[bbritt:~/test] $ find . -iname ‘user’
[bbritt:~/test] $ find . -iname ‘user*’
./username.txt
[bbritt:~/test] $ find . -iname 'User*'
./username.txt
[bbritt:~/test] $
```

Find example

locate

- Find files by name
- This requires that the locate database be installed on the computer (it usually is).
- Usage:
  - **locate** *filename*
- The files are not available immediately in the locate database after file creation. This job typically runs at night.



```
[bbritt:~] $ locate file1
/home/bbritt/test/file1
.
.
.
/opt/rit/app/maker-p/2.31.8/MWAS/html/images/file1.ico
/opt/rit/app/maker-p/2.31.8/MWAS/html/images/file1.png
.
.
.
/usr/lib64/libreoffice/program/libucpfile1.so

[bbritt:~] $
```

Locate example

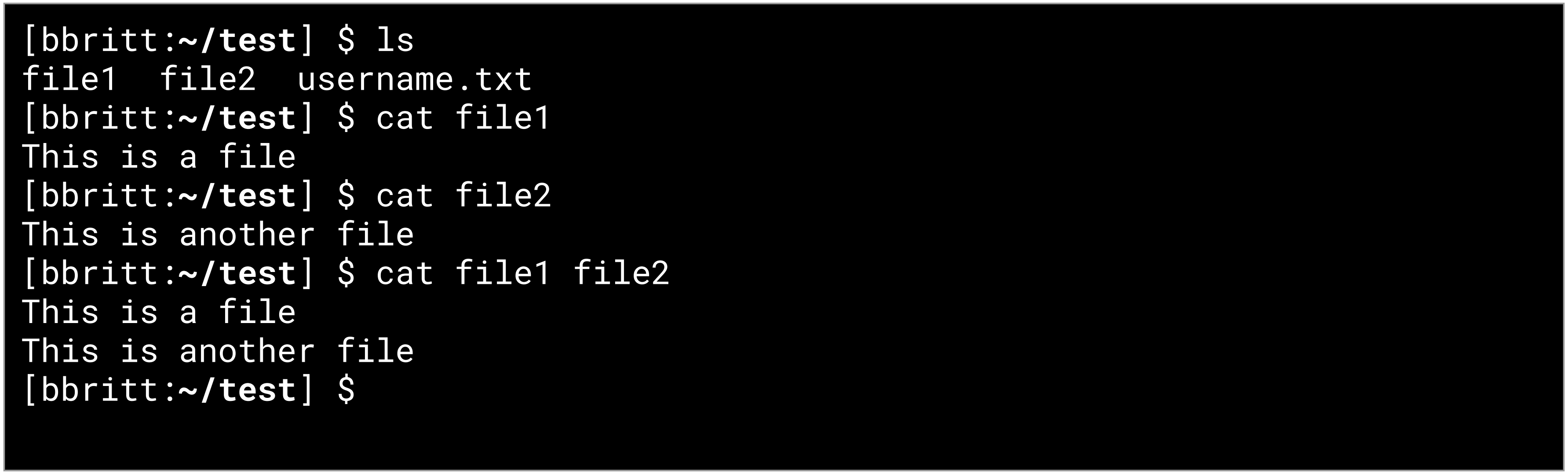
File Manipulation

- cat
- echo
- mkdir
- rmdir
- mv
- cp
- rm
- nano

cat

- Stands for ‘concatenate’
- Does the following:
  - Displays text files on a screen
  - Copies text files
  - Combines text files
  - Make new text files
- Usage:
  - **cat** *filename*
  - **cat** file1 file2

Cat example

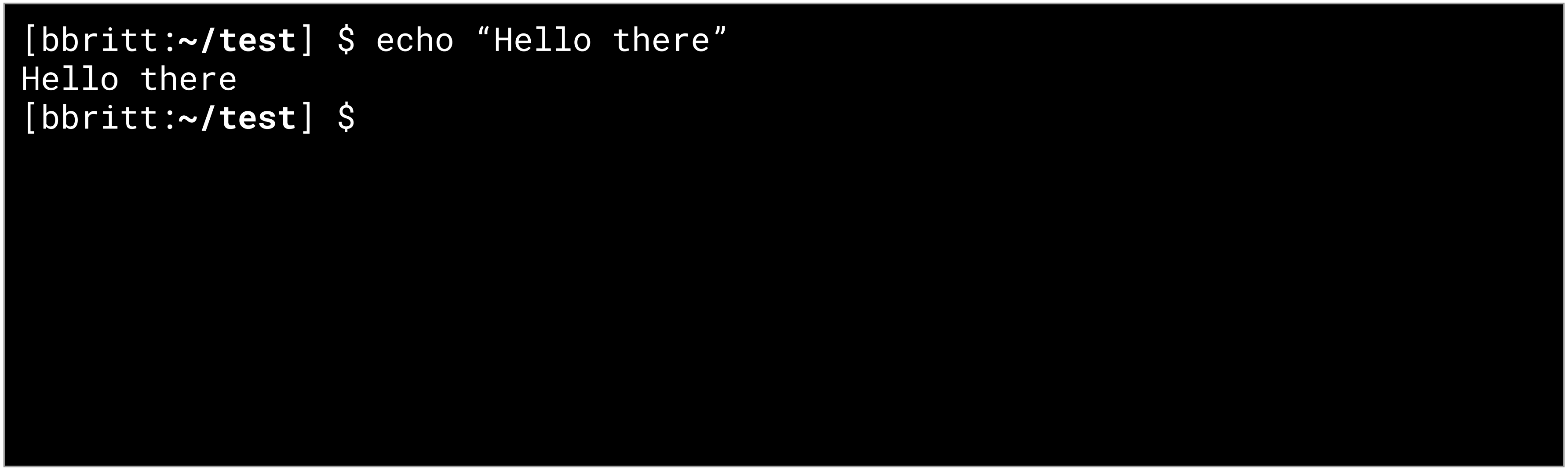


```
[bbritt:~/test] $ ls
file1  file2  username.txt
[bbritt:~/test] $ cat file1
This is a file
[bbritt:~/test] $ cat file2
This is another file
[bbritt:~/test] $ cat file1 file2
This is a file
This is another file
[bbritt:~/test] $
```

echo

- Prints something to the screen
- Usage:
  - **echo** *“Something”*
  - **echo** *“Lots of somethings”*

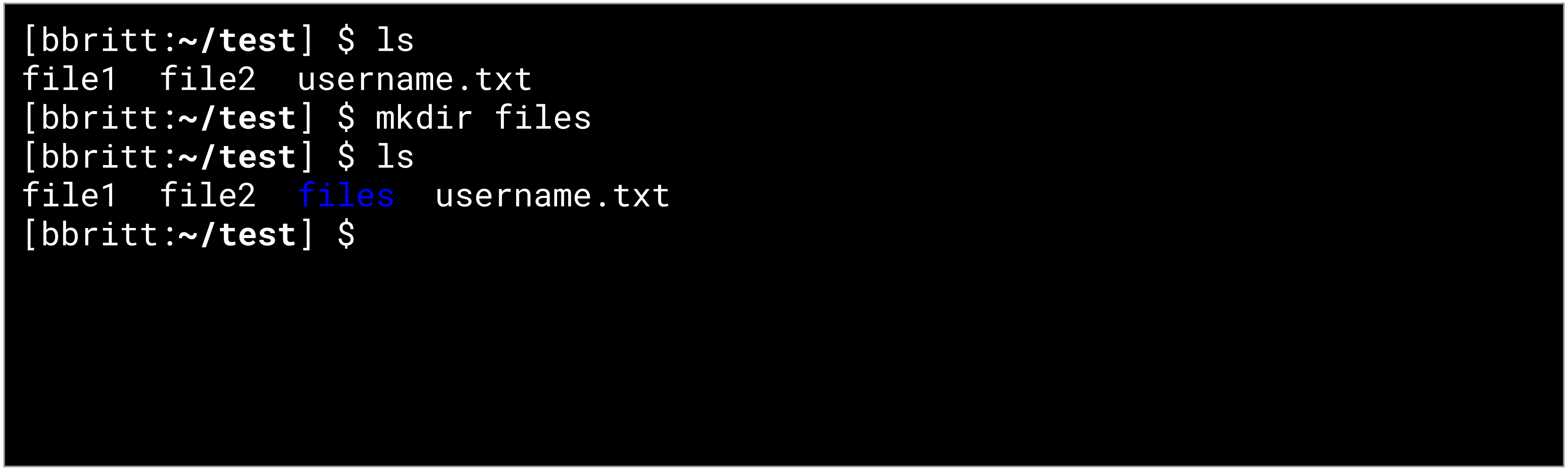
echo example



```
[bbritt:~/test] $ echo “Hello there”
Hello there
[bbritt:~/test] $
```

mkdir

- Short for ‘make directory’
- Can do the following:
  - Create directories, if they don’t already exist
- Usage:
  - **mkdir** *directoryname/*

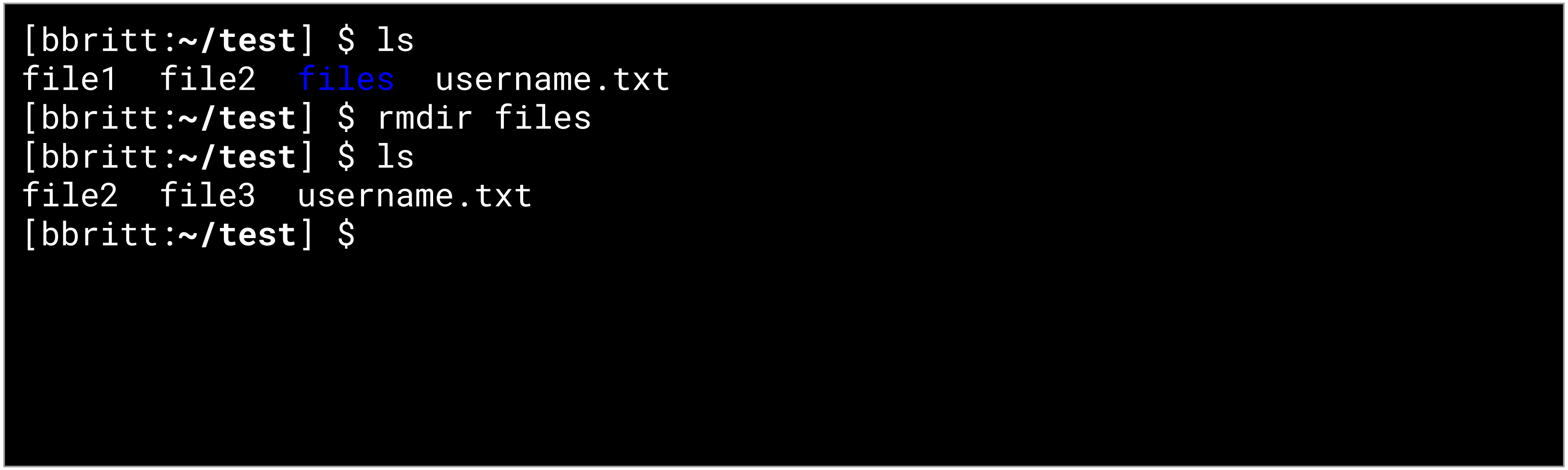


```
[bbritt:~/test] $ ls
file1  file2  username.txt
[bbritt:~/test] $ mkdir files
[bbritt:~/test] $ ls
file1  file2  files  username.txt
[bbritt:~/test] $
```

mkdir example

rmdir

- Short for ‘remove directory’
- Can do the following:
  - Delete empty directories
- Usage:
  - **rmdir** *directoryname/*

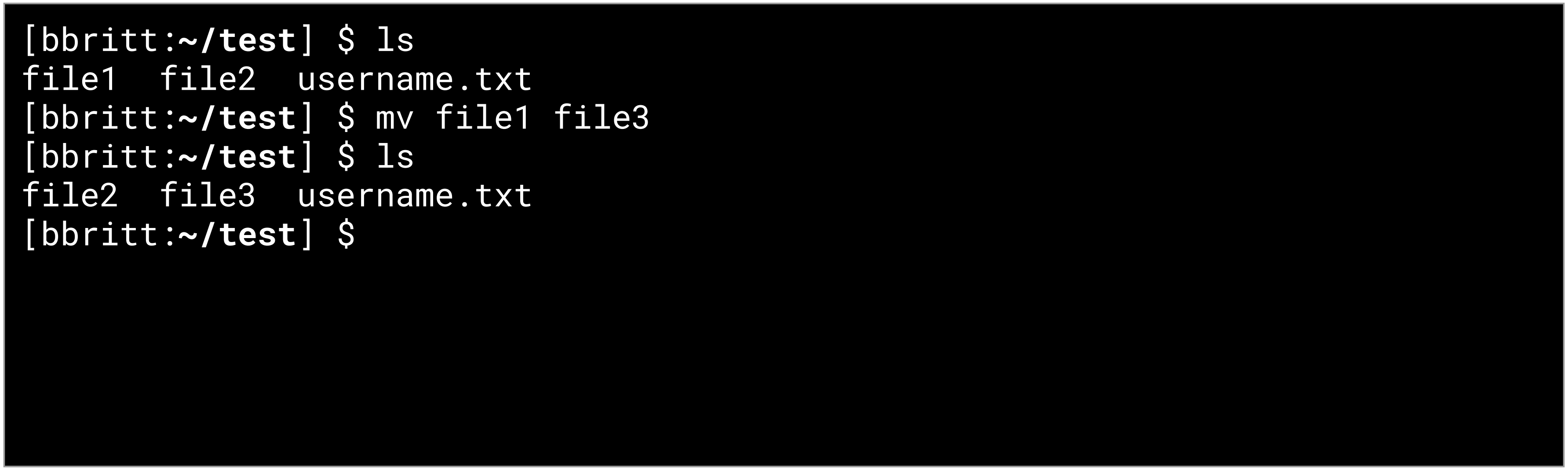


```
[bbritt:~/test] $ ls
file1  file2  files  username.txt
[bbritt:~/test] $ rmdir files
[bbritt:~/test] $ ls
file2  file3  username.txt
[bbritt:~/test] $
```

rmdir example

mv

- Short for ‘move’
- Can do the following:
  - Moves one or more files / directories from one place to another
  - Rename a file
- Usage:
  - **mv** *file destination/*
  - **mv** *oldfilename newfilename*



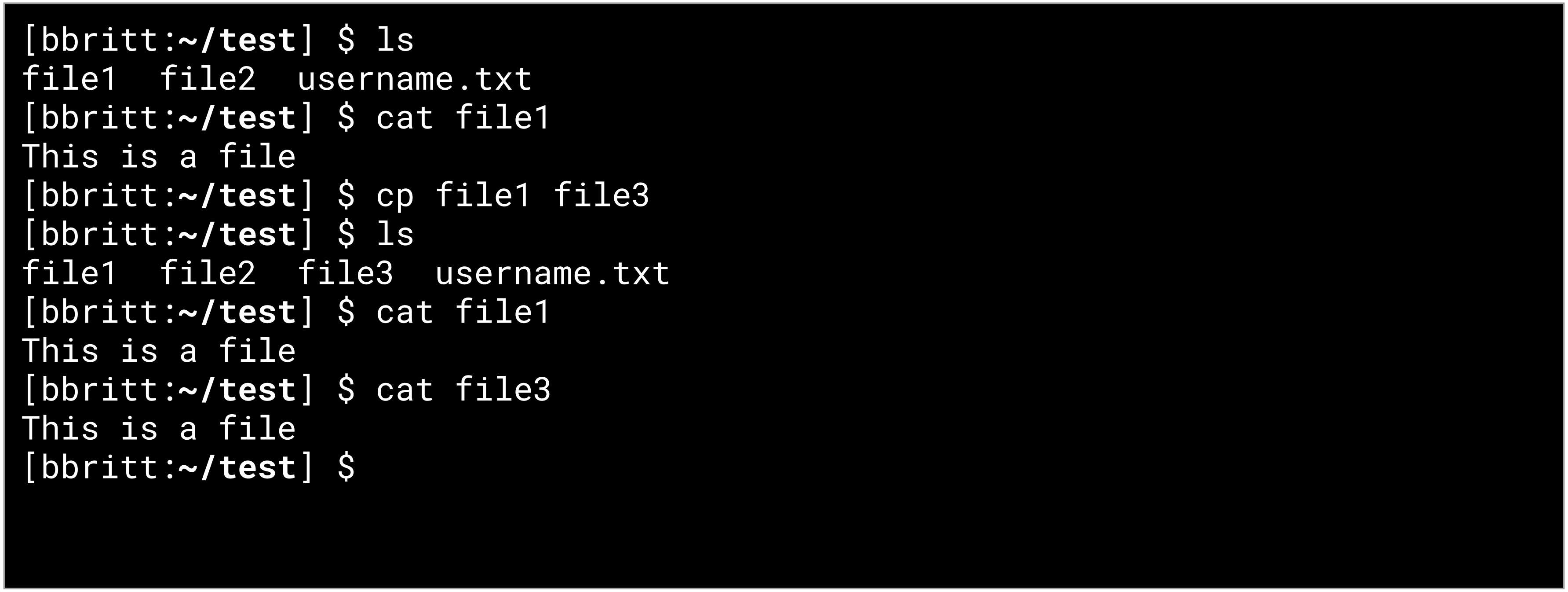
```
[bbritt:~/test] $ ls
file1  file2  username.txt
[bbritt:~/test] $ mv file1 file3
[bbritt:~/test] $ ls
file2  file3  username.txt
[bbritt:~/test] $
```

Cat example

cp

- Short for ‘copy’
- Can do the following:
  - Copies one or more files / directories from one place to another
  - Creates new files with a different name, but the same file contents
- Usage:
  - **cp** *file destination/*
  - **cp** *oldfilename newfilename*
  - **cp** -r olddirectory newdirectory

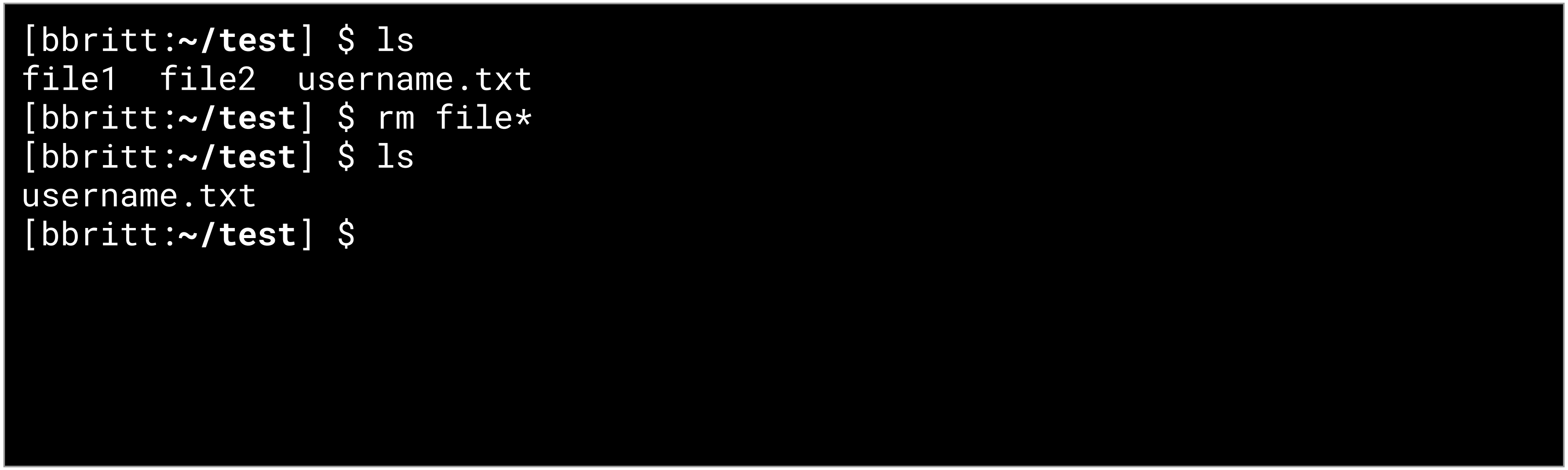
Cp example



```
[bbritt:~/test] $ ls
file1  file2  username.txt
[bbritt:~/test] $ cat file1
This is a file
[bbritt:~/test] $ cp file1 file3
[bbritt:~/test] $ ls
file1  file2  file3  username.txt
[bbritt:~/test] $ cat file1
This is a file
[bbritt:~/test] $ cat file3
This is a file
[bbritt:~/test] $
```

rm

- Stands for ‘remove’
- Usage:
  - **rm** somefile
  - **rm** file1 file2 file3
- Other useful flags
  - **-f** : force removal (don’t ask me if I want to remove stuff)
  - **-r** : **recursive** removal. In other words, remove the file/directory listed and everything in it!

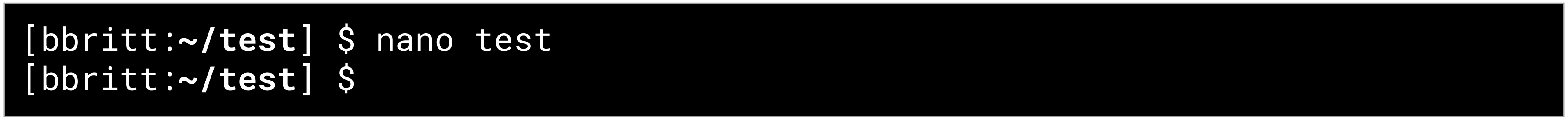


```
[bbritt:~/test] $ ls
file1  file2  username.txt
[bbritt:~/test] $ rm file*
[bbritt:~/test] $ ls
username.txt
[bbritt:~/test] $
```

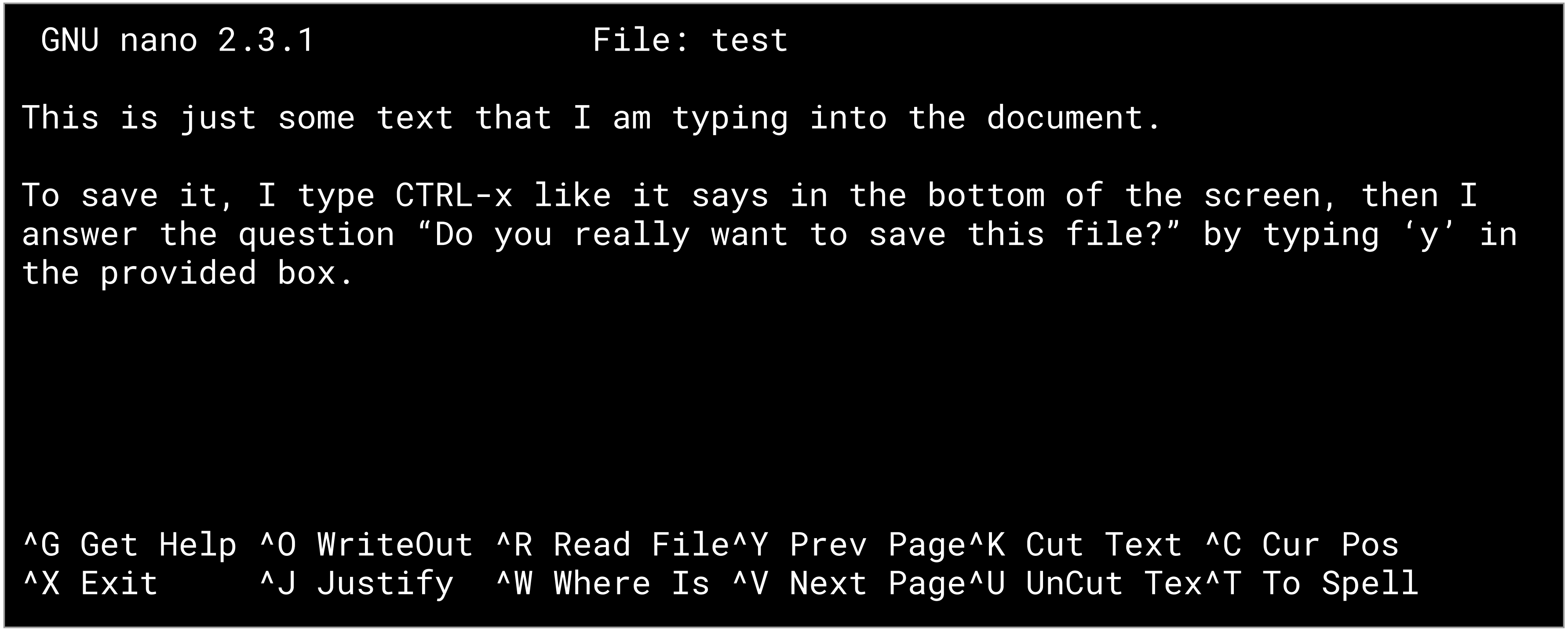
rm example

nano

- Used for:
  - Text editing
  - File creation
- Usage:
  - **nano** *filename*



```
[bbritt:~/test] $ nano test
[bbritt:~/test] $
```



```
 GNU nano 2.3.1              File: test

This is just some text that I am typing into the document.

To save it, I type CTRL-x like it says in the bottom of the screen, then I
answer the question “Do you really want to save this file?” by typing ‘y’ in
the provided box.






^G Get Help ^O WriteOut ^R Read File^Y Prev Page^K Cut Text ^C Cur Pos
^X Exit     ^J Justify  ^W Where Is ^V Next Page^U UnCut Tex^T To Spell
```

Nano example