

# Introduction to the Linux Command Line

- Understand basic jargon (shell, path, file permissions, command line)
- Understand common Linux commands
- Tips on making command-line easier
- File permissions
- Text editors
- SSH/SCP/SFTP

**Eric J. Walter**

**Executive Director of Research Computing**  
**September 14th, 2023**

# Research Computing / HPC Training

1. Thursday – September 14th 5-6pm Zoom – Introduction to Linux command line
2. Thursday – September 21st 5-6pm Zoom – Introduction to HPC at W&M/VIMS
3. Thursday – September 28th 5-6pm Zoom – Linux in the HPC environment

[hpc-help@wm.edu](mailto:hpc-help@wm.edu)

- Ticket system for HPC/RC issues

<https://www.wm.edu/it/rc>

- RC website

2023 September						
SUNDAY	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY	SATURDAY
					1	2
3	4	5	6	7	8	9
10	11	12	13	14 1.	15	16
17	18	19	20	21 2.	22	23
24	25	26	27	28 3.	29	30



**Eric J. Walter**

Executive Director of  
Research Computing

*Joined in 3/13*



**Jay Kanukurthy**

Applications Analyst  
(software installs / debugging,  
batch system, user onboarding)

*Joined in 1/16*



**Matt Kennedy**

Network, Filesystems &  
Data Specialist  
(network, data science  
hardware, tape backup, RC  
storage)

*Joined in 4/20*



**Daanish Fiaz**

Systems Administrator  
(hardware, user accounts, security  
patching, tape backup)

*Joined in 9/20*



**Malcolm Slaughter**

Research Computing  
Specialist  
(RC field support, RC storage,  
Linux workstation/software,  
special projects)

*Joined in 7/23*

# Introduction to the Linux Command Line

**Windows, MacOS** have some command-line features - "*dos prompt*", "*run*", "*powershell*" for Windows or "*terminal*" for MacOS (BSD shell)

Most of you are probably familiar with Graphical User Interfaces (GUI).

Linux especially for HPC is mostly command-line. Some GUI interfaces, **but command-line interface (CLI) can be and often is superior.**

- **Command Line** – Text based interface that lets you interact with computer
- **Shell** – program that presents the command-line interface
- **Terminal** – program that runs a shell
- **Prompt** – beginning of command line which prompts you for action

**Shells** come in different flavors (tcsh, bash, zsh, ksh, fish, etc. )

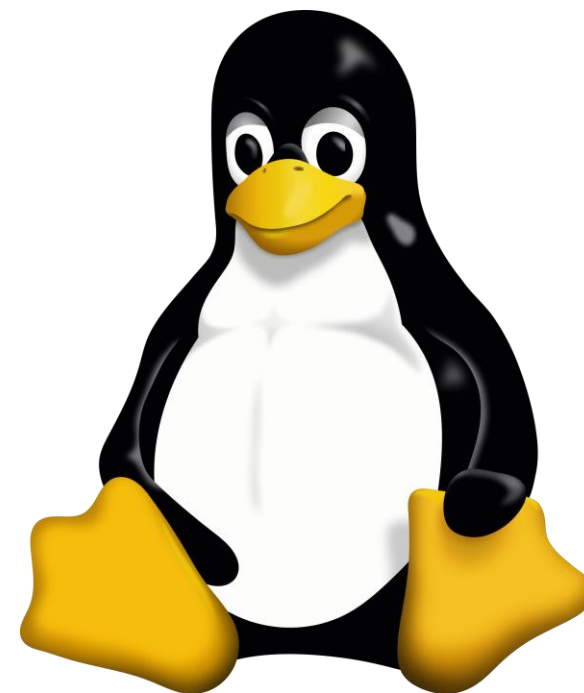
On W&M/VIMS clusters, **tcsh** is default shell, users can request **bash**

What shell am I running?:

```
1 [bora]
2 [bora] echo $0
-tcsh
```

prompt (line number increments with each command)

Everything today will be **shell agnostic**



"Tux" - Linux mascot

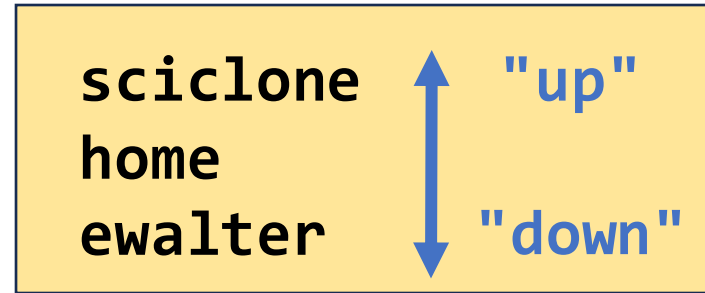
All HPC users get **tcsh** by default  
can request **bash**, experimenting with  
**zsh** for new HPC OS

# Paths / Streams / Files

*pwd* – print working directory  
aka current **path**

```
24 [bora]$ pwd  
/sciclone/home/ewalter
```

<--- current path



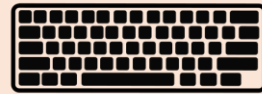
In Linux there are 3 streams:

Input (stdin) < (rarely used)

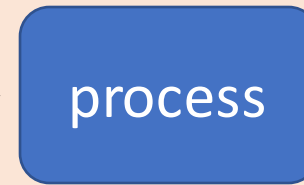
Output (stdout) > (>> append)

Error (stderr) &

Both stderr and stdout >&



stdin



stdout

stderr



*echo* – display text or value of variable

*echo hello > file1* – echo word "hello" into file named "file1"

*ls* – list current directory

*cat / more / less* – dump contents of file (hit "q" to exit less).

**Output redirection is very useful to capture the output of commands or programs.**

# Directories (folders) / Files / History

**mkdir** – make directory

**cd** – change directory

**rmdir** – remove directory

**rm** – remove file

**mv** – move / rename file

**cp** – copy file

**ls** – list contents of  
current directory

**history** – CLI history (!)

**clear** – clear the screen

**\*** – wildcard symbol

**.** – current directory

**..** – up one directory

**~** – home directory

**-** – previous directory

relative vs. absolute paths

**cd** – by itself means cd to  
my home

```
31 [bora] pwd                print working directory
/sciclone/home/ewalter/LinuxPres
32 [bora] mkdir newdir1      make directory "newdir1"
33 [bora] mkdir newdir2      make directory "newdir2"
34 [bora] ls                 list contents of current dir
newdir1 newdir2
35 [bora] cd newdir1         go into "newdir1"
36 [bora] ls                 list current content (it is empty)
37 [bora] cd ..             go up one directory
38 [bora] ls                 list current directory
newdir1 newdir2
39 [bora] ls > dirlist        ... send stdout of ls to file "dirlist"
40 [bora] cat dirlist        dump "dirlist" to screen
dirlist
newdir1
newdir2
41 [bora] rmdir newdir2      remove newdir2
42 [bora] mv newdir1 newdir  rename newdir1 to "newdir"
43 [bora] history            list command history
 31 20:09 pwd
 32 20:09 mkdir newdir1
 33 20:09 mkdir newdir2
 35 20:09 cd newdir1
 37 20:09 cd ..
 38 20:09 ls
 39 20:10 ls > dirlist
 40 20:10 cat dirlist
 41 20:10 rmdir newdir2
 42 20:10 mv newdir1 newdir
 43 20:10 history
44 [bora] !38                repeat command #38
ls                             ! - reports what command was
dirlist newdir
```

```
45 [bora] cp dirlist newdir  copy file "dirlist" into newdir
46 [bora] ls                 list contents (nothing changed)
dirlist newdir
47 [bora] ls newdir         list contents of "newdir"
dirlist
48 [bora] cp dirlist dirlist2 make copy of dirlist named dirlist2
49 [bora] ls                 list
dirlist dirlist2 newdir
51 [bora] cp dir* newdir     cp all files starting with dir to "newdir"
52 [bora] cd newdir         go into newdir
53 [bora] ls                 list
dirlist dirlist2
54 [bora] cd ..             go up one directory
55 [bora] cd ~              cd to home directory
56 [bora] pwd               print current path
/sciclone/home/ewalter
57 [bora] cd ~ewalter       cd to home directory (another way)
58 [bora] pwd               print current path
/sciclone/home/ewalter
59 [bora] cd ~ewalter/LinuxPres go to LinuxPres (in my home directory)
60 [bora] pwd               print current path
/sciclone/home/ewalter/LinuxPres
61 [bora] cd                 go to home directory (yet another way)
62 [bora] pwd               print current path
/sciclone/home/ewalter
63 [bora] cd -               go to previous directory
64 [bora] pwd               print current path
/sciclone/home/ewalter/LinuxPres
65 [bora] cd ~               go home
66 [bora] cd /sciclone/home/ewalter/LinuxPres absolute path
67 [bora] cd ~               go home
68 [bora] cd LinuxPres      relative path
```

# Switches/man/hidden files

## Many Linux commands have options called switches

<code>ls -l</code>	list with "long" option
<code>ls -ltr</code> or <code>ls -l -t -r</code>	"long" and sort by reverse modification time
<code>rm -r</code> testdir	remove "recursively" <b>CAREFUL!</b>
hidden files <code>ls -l</code> vs. <code>ls -la</code>	files/folder that being with '.' "dot" files
long vs. short switches <code>ls -a</code> vs. <code>ls --all</code>	

## Presence of trailing slash indicates directory

**Note:** directories can be referred to as "<directory>" or "<directory>/". The presence of the trailing slash (/) can change some commands behavior. e.g.

Consider you want to mv the file "newfile" into the directory "newdir" **but** you forget to make the directory "newdir" ...

```
1 [bora] mv newfile newdir          renames newfile to newdir
2 [bora] mv newfile newdir/
mv: failed to access 'newdir/': Not a directory    complains
```

Important when using "rsync"

```
90 [bora] pwd
/sciclone/home/ewalter/LinuxPres
91 [bora] ls -l
total 8
-rw-----. 1 ewalter hpcf 24 Sep 10 20:50 dirlist
-rw-----. 1 ewalter hpcf 24 Sep 10 20:11 dirlist2
drwx-----. 2 ewalter hpcf 10 Sep 10 20:30 newdir
92 [bora] ls -ltr
total 8
-rw-----. 1 ewalter hpcf 24 Sep 10 20:11 dirlist2
drwx-----. 2 ewalter hpcf 10 Sep 10 20:30 newdir
-rw-----. 1 ewalter hpcf 24 Sep 10 20:50 dirlist
93 [bora] mkdir .hidden
94 [bora] ls -l
total 8
-rw-----. 1 ewalter hpcf 24 Sep 10 20:50 dirlist
-rw-----. 1 ewalter hpcf 24 Sep 10 20:11 dirlist2
drwx-----. 2 ewalter hpcf 10 Sep 10 20:30 newdir
95 [bora] ls -la
total 56
drwx-----.  4 ewalter hpcf      86 Sep 10 20:51 .
drwxr-xr--. 173 ewalter hpcf 86016 Sep 10 20:08 ..
-rw-----.  1 ewalter hpcf      24 Sep 10 20:50 dirlist
-rw-----.  1 ewalter hpcf      24 Sep 10 20:11 dirlist2
drwx-----.  2 ewalter hpcf     10 Sep 10 20:51 .hidden
drwx-----.  2 ewalter hpcf     10 Sep 10 20:30 newdir
96 [bora] ls -a
.  ..  dirlist  dirlist2  .hidden  newdir
97 [bora] ls --all
.  ..  dirlist  dirlist2  .hidden  newdir
98 [bora]
99 [bora] rmdir newdir
rmdir: failed to remove 'newdir': Directory not empty
100 [bora] rm -r newdir
101 [bora] ls
dirlist  dirlist2
```

print working directory

list with "-l" switch (*long*)

list with long and reverse by modification time

make a "hidden" directory  
list -l

list with -l and -a (for all)

the hidden directory

Short list all files with -a

Short list with -all

Try removing non-empty dir

remove -r recursively

# du, df, find

**df** – get filesystem usage (**df -h**)

**du** – get current directory disk usage

**find** – find a file

**df** and **du** are important in HPC since you share file-systems with other users.

**du** lets you identify large files/folders  
**df** tells you how full that filesystem is

**find . -name <file name>**

'.' refers to where to search

**-name** means match the filename

**<file name>** is the file you are  
searching for

**find . -type d -name <directory name>**

Will search for directory

```
28 [bora] df -h
Filesystem                                Size  Used Avail Use% Mounted on
.
.
.
192.168.56.208@o2ib,192.168.56.209@o2ib:/pscr 273T  102T  172T  38% /sciclone/pscr
po00-ib:/sciclone/scr10                    219T   38T  181T  18% /sciclone/scr10
cm00:/sciclone/home                        16T   12T   3.9T  76% /sciclone/home
tp00-ib:/sciclone/data10                   360T  334T   27T  93% /sciclone/data10
tw00-i8:/sciclone/scr20                     73T   25T   48T  35% /sciclone/scr20
.
.
.
28 [bora] pwd
/sciclone/home/ewalter/examples
29 [bora] du -hs .
22G    total
30 [bora] du -hs *
4.0K CODE_OF_CONDUCT.md
4.0K CODEOWNERS
4.0K CONTRIBUTING.md
436K cpp
.
.
.
31 [bora] cd ..
32 [bora] find . -name CODEOWNERS
examples/CODEOWNERS
```

df with -h switch ("human readable")

pwd

-hs "human-readable" + "summarize" examples is 22 GB

-hs \* usage of each file and folder in this directory

cd up one level  
find file named "CODEOWNERS" in or below this directory

# tab completion, up arrow, man pages

Some really helpful commands...

## tab completion

*ls file<tab>*

This will complete the word "file" with whatever files/folders in your current directory match this name.

## up arrow

Hitting the up-arrow key will cycle back through your current history

**man pages** exist for almost all Linux commands

*man ls*

man pages, tab completion, up arrow, history all useful

Google also helpful!!

```
25 [bora] ls
data1 data2 runscript
26 [bora] ls d<TAB> -> ls data
data1 data2
26 [bora] ls data2
data2
29 [bora] ls ru<TAB> -> ls runscript
runscript

30 [bora] <up arrow> -> ls runscript
30 [bora] <up arrow> -> ls data2
30 [bora] <up arrow> -> ls

31 [bora] man ls

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 .

    -A, --almost-all
        do not list implied . and ..

    --author
        with -l, print the author of each file

.
```



# tar and gzip/bzip2

**grep** – globally search for a regular expression and print matching lines  
"|" (pipe) - pipe output of one command into another

## Tar and gzip/bzip2 files:

**tar** – tape archive

**gzip** – GNU zip

**bzip2** – Another zip program

tar – combine files into single file collection  
does not compress

Often combined with gzip or bzip2 to  
compress a tarfile (tarball)

**gzip** – compresses a file -> file.gz

**bzip2** – also compresses a file > file.bz2

```
37 [bora] grep pytorch README.md
![Run Examples](https://github.com/pytorch/examples/workflows/Run%20Examples/badge.svg)
https://pytorch.org/examples/ `pytorch/examples` is a repository showcasing examples of using
.
.
.
38 [bora] grep pytorch README.md |grep !
![Run Examples](https://github.com/pytorch/examples/workflows/Run%20Examples/badge.svg)
39 [bora]

115 [bora] pwd
/sciclone/home/ewalter/LinuxPres
116 [bora] ls
FILES
117 [bora] ls FILES
file1 file2 file3 otherfiles
118 [bora] tar czvf FILES.tgz FILES/
FILES/
FILES/otherfiles/
FILES/otherfiles/file4
FILES/otherfiles/file5
FILES/otherfiles/file6
FILES/file1
FILES/file2
FILES/file3
119 [bora] ls
FILES FILES.tgz
120 [bora] tar tzvf FILES.tgz
drwx----- ewalter/hpcf 0 2023-09-11 17:50 FILES/
drwx----- ewalter/hpcf 0 2023-09-11 17:49 FILES/otherfiles/
-rw----- ewalter/hpcf 0 2023-09-11 17:49 FILES/otherfiles/file4
-rw----- ewalter/hpcf 0 2023-09-11 17:49 FILES/otherfiles/file5
-rw----- ewalter/hpcf 0 2023-09-11 17:49 FILES/otherfiles/file6
-rw----- ewalter/hpcf 0 2023-09-11 17:49 FILES/file1
-rw----- ewalter/hpcf 0 2023-09-11 17:49 FILES/file2
-rw----- ewalter/hpcf 0 2023-09-11 17:49 FILES/file3
```

# tar and gzip/bzip2 #2

```
tar cvf mytarball.tar <dir>
gzip mytarball.tar
Mytarball.tar.gz
```

## Can do in one step

```
tar czvf file.tgz <dir>
tar cjvf file.tbz2 <dir>
```

**Always** list tar file first (with **-t**)

**Always** tar up a directory

file.tgz = file.tar.gz - file that is a gzipped tarball

file.tbz2 file.tar.bz2 - file that is a bzip2'd tarball

Three main operations

**c** - create, **t** - list, **x** - extract

<b>tf</b>	- list tar ball
<b>tvf</b>	- list tar ball verbosely
<b>tzvf</b>	- list gzipped tar ball verbosely
<b>tjvf</b>	- list bzip2'd tar ball verbosely
<b>xf</b>	- extract tar ball
<b>cf</b>	- create tar ball

```
47 [bora] pwd /sciclone/home/ewalter/examples
48 [bora] ls
CODE_OF_CONDUCT.md distributed gat mnist regression super_resolution
CODEOWNERS docs gcn mnist_forward_forward reinforcement_learning time_sequence
CONTRIBUTING.md examples.tgz imagenet mnist_hogwild run_python_examples.sh vae
cpp fast_neural_style legacy mnist_rnn runtime.txt vision_transformer
dcgan fx LICENSE README.md siamese_network word_language_model

49 [bora] tar czf examples.tgz *
50 [bora] ls -l examples.tgz

51 [bora] ls -lh examples.tgz
-rw----- . 1 ewalter hpcf 453M Sep 12 12:04 examples.tgz

55 [bora] tar tzf examples.tgz
CODE_OF_CONDUCT.md
CODEOWNERS
CONTRIBUTING.md
cpp/
cpp/.clang-format
cpp/autograd/
cpp/autograd/CMakeLists.txt

56 [bora] cp examples.tgz ~
57 [bora] cd
58 [bora] tar xvf examples.tgz
```

This is a bad idea since this tarball extracts many files to top level! Will make a mess that is hard to clean up.

A lot of software and datasets are distributed as tarballs (usually compressed in some way)

# Permissions

```
11 [bora] pwd
/sciclone/home/ewalter
12 [bora] ls -l
total 12
-rw----- . 1 ewalter hpcf 6 Jun 7 18:40 file1
-rw----- . 1 ewalter hpcf 6 Jun 7 18:41 file2
-rw----- . 1 ewalter hpcf 6 Jun 7 18:41 file3
drwx----- . 2 ewalter hpcf 6 Jun 7 18:41 testdir
```

directory? user group other user group file/folder

**u** – user (ewalter)

**g** – group (hpcf)

**o** – other (everyone else)

**r** - read permission

**w** – write permission

**x** – execute / traverse permission

File/folder permissions important for sharing files with others withing the HPC clusters.

e.g. Want to allow everyone to copy file1 out of my home directory:

```
77 [bora] ls -l file1
-rw----- . 1 ewalter hpcf 6 Jun 7 18:40 file1
78 [bora] ls -ld /sciclone/home/ewalter
drwx----- . 173 ewalter hpcf 86016 Sep 12 12:17 /sciclone/home/ewalter
79 [bora] chmod go+rx /sciclone/home/ewalter
80 [bora] ls -ld /sciclone/home/ewalter
drwxr-xr-x. 173 ewalter hpcf 86016 Sep 12 12:17 /sciclone/home/ewalter
81 [bora] chmod go+r file1
82 [bora] ls -l file1
-r--r--r--. 1 ewalter hpcf 6 Jun 7 18:40 file1
```

Permissions discussed more in Linux for HPC (coming soon)

# Text Editor

To create / edit files, need to use a **text editor**. All HPC systems have: *vi/vim*, *emacs*, and *nano*

*Nano* is most useful for beginners <https://www.howtogeek.com/42980/the-beginners-guide-to-nano-the-linux-command-line-text-editor/>

```
GNU nano 2.0.9      File: txt_files/testfile      Modified
Learn how to use nano to boost your terminal confidence!
Edit config files like a pro!
Make easy to-do lists and notes in a text-only format!
Do it via SSH from a smartphone or other computer!
#
# /etc/fstab: static file system information.
#
# Use 'blkid -o value -s UUID' to print the universally unique identifier
# for a device; this may be used with UUID= as a more robust way to name
# devices that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point> <type> <options> <dump> <pass>
proc /proc proc defaults 0 0
# / was on /dev/sdb1 during installation
[ Read 17 lines ]
^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 Text ^T To Spell
```

## How to launch

1 [bora] nano

or

1 [bora] nano <filename>

List of commands at bottom of screen: i. e. **cntrl-X** = exit, **cntrl-O** = write file

Also, can set up visual studio code to connect to files on HPC via ssh.

# SSH/SCP/SFTP

**ssh** – is the standard app for connecting to a remote computer with Linux

**scp** – is the standard app for copying files from one site to another

**sftp** – more convenient for multiple hops

Both are easy within Linux. Mac also supports both via **terminal**. Windows users can use **powershell** to do this or use ssh client program (**putty**) and a separate scp client (**WinSCP**).

ssh to bora

- **ssh ejwalt@bora.sciclone.wm.edu**

ssh to bora through bastion host

- **ssh -J ejwalt@bastion.wm.edu ejwalt@bora.sciclone.wm.edu**

scp to bora

- **scp file1 ejwalt@bora.sciclone.wm.edu:**

scp from bora

- **scp ejwalt@bora.sciclone.wm.edu:file1 .**

sftp double hop

- **sftp -J ejwalt@bastion.wm.edu ejwalt@bora.sciclone.wm.edu**

See <https://www.wm.edu/offices/it/services/researchcomputing/using/connecting/> - connecting/logging in to HPC  
<https://www.wm.edu/offices/it/services/researchcomputing/using/xfers/> - transferring files to/from HPC

[hpc-help@wm.edu](mailto:hpc-help@wm.edu)

- Ticket system for HPC/RC issues

<https://www.wm.edu/it/rc>

- RC website

Other Linux Tutorials - <https://www.wm.edu/offices/it/services/researchcomputing/using/prereqs/index.php>