

Please Check In:



sh.lug.umbc.edu/checkin

UMBC LINUX USERS GROUP

Please Check In:



sh.lug.umbc.edu/checkin

Club Updates:

- Sysadmins have been onboarded, new progress on the mirror and infrastructure expected
 - Applications are still open at sh.lug.umbc.edu/sysadmin
- Upcoming Presentations:
 - Next Week: FOSS
- We still are open for presentation signups
 - sh.lug.umbc.edu/present-sp26

Who are we?



President
Alex (blahaj)



Vice President
dani (puppy)



Treasurer
Will (pptx.)

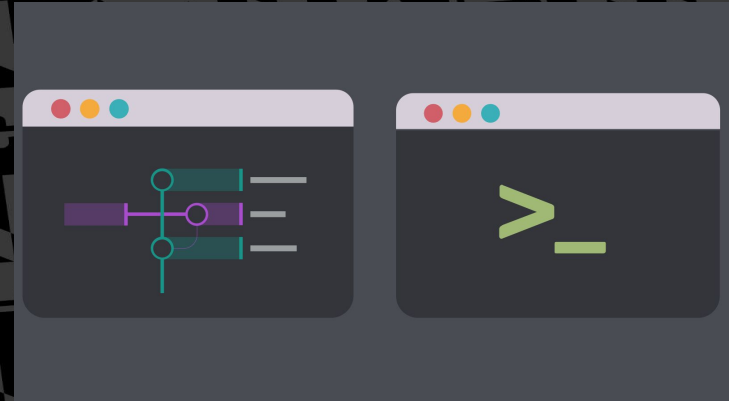


Secretary
Milk (🥛)

Today's Presentation:
Command Line Basics: Getting
Started with Linux

Why the command Line?

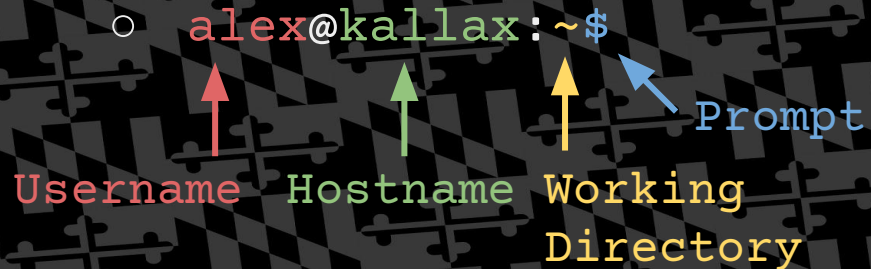
- Lightweight, powerful, and flexible
- Automates repetitive tasks
- Works on any Linux system (even servers without GUI)



The Terminal Environment

- Open the terminal (Ctrl + Alt + T) - Most systems
- Shell types (bash, zsh, fish)
- Prompt Structure

○ alex@kallax:~\$



The diagram shows the prompt 'alex@kallax:~\$' with four arrows pointing to its parts: a red arrow from 'alex' to 'Username', a green arrow from 'kallax' to 'Hostname', a yellow arrow from '~' to 'Working Directory', and a blue arrow from '\$' to 'Prompt'. Below these labels, the words 'Username', 'Hostname', 'Working Directory', and 'Prompt' are written in their respective colors.

Username Hostname Working Directory Prompt

Navigating

- `pwd` - print working directory
- `ls` - list files
- `cd` - change directory

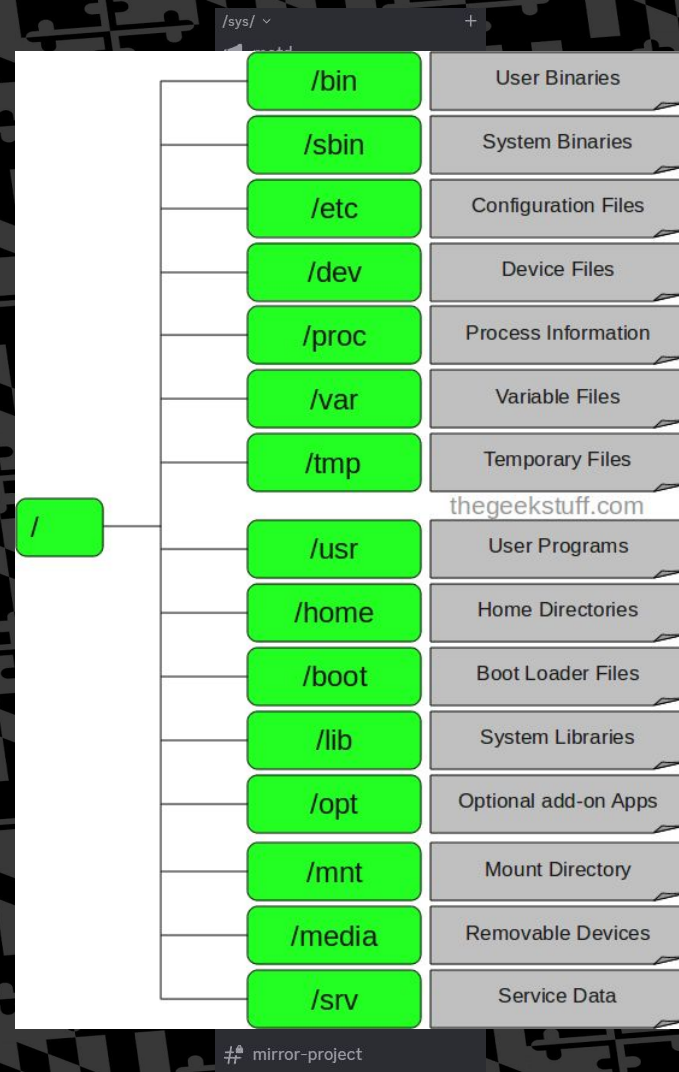
Options & arguments

- `ls -la` (uses the options `-l` & `-a`)
 - Shows all files (hidden files)
 - List format
- After the options are the actuals arguments



Linux File Structure

- Files & Directories
- Hierarchical file structure
 - Root Directory '/'
 - Subsequent directories within
 - Path: '/home/alex/LUG/umbx.txt'
 - Absolute path
 - Relative path
 - LUG/umbc.txt
- Standard Linux directory structure



File operations

- touch - create empty file
- cp - copy files
- mv - move/rename files
- rm - remove files
- mkdir /rmdir - create/remove directories



Viewing and Editing Files

- cat, less, head, tail - view contents
- nano - beginner friendly text editor
- vim - powerful text editor

How do I exit Vim?

Asked 13 years, 1 month ago Modified 8 months ago Viewed 3.2m times



I am stuck and cannot escape. It says:

5495

type `:quit<Enter>` to quit VIM



But when I type that it simply appears in the object body.



vim

vi

LINUX TERMINAL FOR BEGINNERS

head

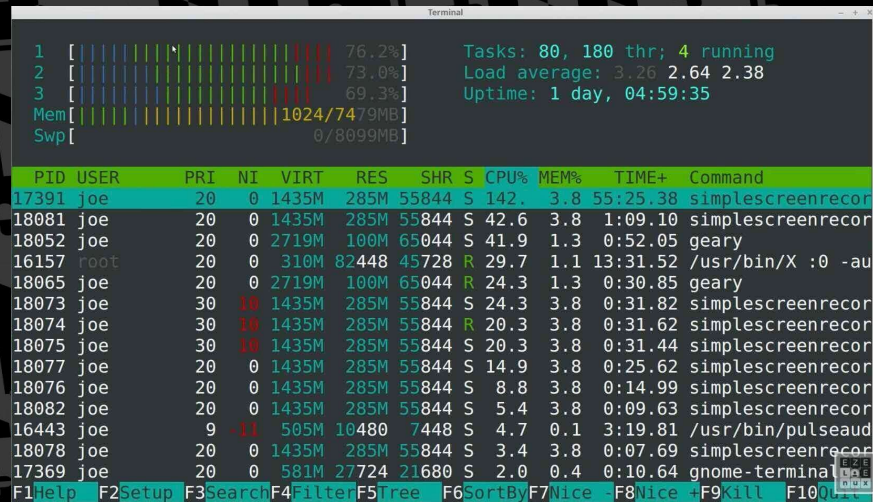


tail

cat

System Info & Management

- Whoami - current user
- Uname -a - kernel/system info
- Df -h - disk usage
- top/htop - monitor processes
- Free -h - memory usage
- ps - list processes
- kill - terminate process



The screenshot shows a terminal window with a dark background. At the top, it displays system status information including CPU usage (76.2%, 73.0%, 69.3%), memory usage (1024/7479MB), and swap usage (0/8099MB). To the right, it shows system statistics: Tasks: 80, 180 thr; 4 running; Load average: 3.26 2.64 2.38; and Uptime: 1 day, 04:59:35. Below this, a table lists running processes with columns for PID, USER, PRI, NI, VIRT, RES, SHR, S, CPU%, MEM%, TIME+, and Command. The processes listed include simplescreenrecorder, geary, and gnome-terminal.

PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%	MEM%	TIME+	Command
17391	joe	20	0	1435M	285M	55844	S	142.6	3.8	55:25.38	simplescreenrecor
18081	joe	20	0	1435M	285M	55844	S	42.6	3.8	1:09.10	simplescreenrecor
18052	joe	20	0	2719M	100M	65044	S	41.9	1.3	0:52.05	geary
16157	root	20	0	310M	82448	45728	R	29.7	1.1	13:31.52	/usr/bin/X :0 -au
18065	joe	20	0	2719M	100M	65044	R	24.3	1.3	0:30.85	geary
18073	joe	30	10	1435M	285M	55844	S	24.3	3.8	0:31.82	simplescreenrecor
18074	joe	30	10	1435M	285M	55844	R	20.3	3.8	0:31.62	simplescreenrecor
18075	joe	30	10	1435M	285M	55844	S	20.3	3.8	0:31.44	simplescreenrecor
18077	joe	20	0	1435M	285M	55844	S	14.9	3.8	0:25.62	simplescreenrecor
18076	joe	20	0	1435M	285M	55844	S	8.8	3.8	0:14.99	simplescreenrecor
18082	joe	20	0	1435M	285M	55844	S	5.4	3.8	0:09.63	simplescreenrecor
16443	joe	9	-13	505M	10480	7448	S	4.7	0.1	3:19.81	/usr/bin/pulseaud
18078	joe	20	0	1435M	285M	55844	S	3.4	3.8	0:07.69	simplescreenrecor
17369	joe	20	0	581M	27724	21680	S	2.0	0.4	0:10.64	gnome-terminal

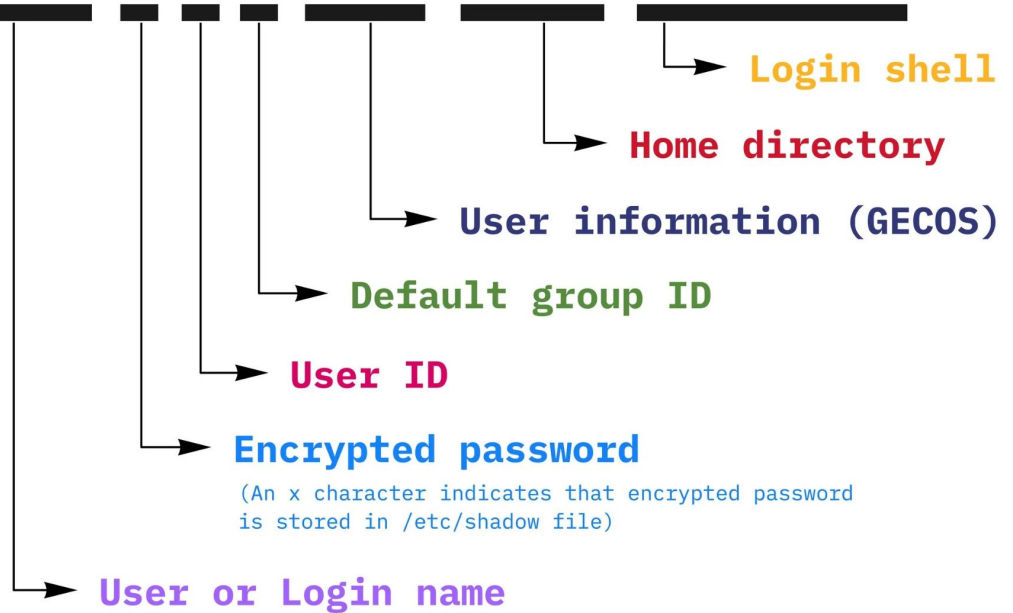
At the bottom of the terminal window, there is a row of function key shortcuts: F1Help, F2Setup, F3Search, F4Filter, F5Tree, F6SortBy, F7Nice, F8Nice, F9Kill, F10.



Users and Permissions

/etc/passwd

root:x:0:0:root:/root:/bin/bash



- UID of zero always means root (carries those perms)

File Permissions

```
drwxr-xr-x  5 alex alex 4096 Aug 12 21:37 .vscode-server
-rw-r--r--  1 alex alex  183 Aug 12 21:36 .wget-hsts
drwxr-xr-x  5 alex alex 4096 Sep 18 21:53 SamsungHealth
-rw-r--r--  1 alex alex 3541 Sep 21 22:27 jeopardy.py
drwxr-xr-x  3 alex alex 4096 Aug 12 21:39 shelf
```

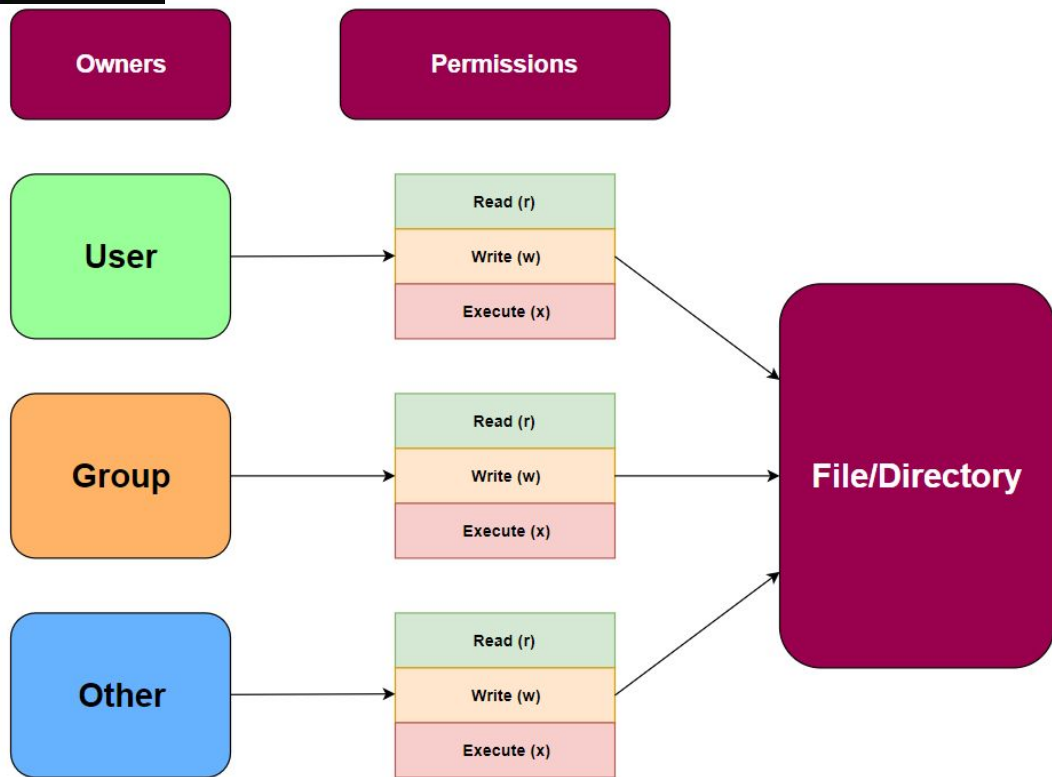
User Group

-rwxrwxrwx

USER (U) GROUP (G) OTHERS (O)

- = file
d = directory
l = Symbolic Link

r = read (4)
w = write (2)
X = execution (1)



Changing file permissions

- `chmod`
 - `sudo chmod -R 755 /home/alex/Documents`
 - `sudo chmod 777 /etc/passwd`
 - `chmod u=rw,go=r document.odt`
 - `chmod u+x script.sh`
- `chown`
 - `sudo chown dev script.sh`
 - `sudo chown docker:docker script.sh`

Sudo

Sudo is an application that handles having multiple admins.

Allows users root access with their own credentials

Membership defined in `/etc/sudoers`

```
alex@LAPTOP-AJ8EBTVQ:/$ getent group sudo  
sudo:x:27:alex
```

Demo

(This could theoretically mess with things so be careful)

- Create a new user '\$ adduser dev'
- Modify the UID bit of the user to be zero
- Switch to the test account

SUID/SGID bit

```
chmod u+s <file>
```

- **rws** **r--** **r--** 1 **user** **admin**

SUID bit set! This will run as **user**

```
chmod g+s <file>
```

- **rwX** **r-s** **r--** 1 **user** **admin**

SGID bit set! This will run with the **admin** group

Changing SUID Bits

- Remove the SUID bit from passwd
 - `chmod u-s /bin/passwd`
- Add SUID bit to /bin/bash
 - `chmod u+s /bin/bash`
- Remove the SUID bit from sudo
 - ?

Package Management

- Responsible for installing, updating, and managing packages
- Debian/Ubuntu - apt install
- Fedora/RHEL - dnf install
- Arch: pacman -S

yum
yellowdog updater modified

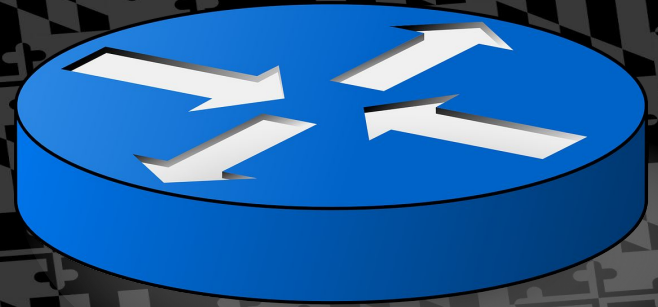
Debian package
management

dpkg
apt
aptitude

PAC-MAN

DNF

Networking



- ping - test connectivity
- ifconfig / ip addr - network info
- ip route - show routing table
- nslookup - DNS record lookup
- Traceroute - trace path to destination
- netstat -tuln - show open connections
- curl /wget - download files
- ssh - connect to remote systems

Essential Applications

- Grep - search files in text
- Find - locate files
- Tar / zip / unzip - archive/compression
- History - see command history
- Man - read manual pages

linuxhandbook.com

grep = g/re/p

- ↗ g - global search
- re - regular expression
- ↘ p - print

grep = global search for regular expression and print the result

```
SED(1)                                User Commands                                SED(1)

NAME
sed - stream editor for filtering and
transforming text

SYNOPSIS
sed [OPTION]... {script-only-if-no-other-
script} [input-file]...

DESCRIPTION
Sed is a stream editor. A stream editor is
used to perform basic text transformations
on an input stream (a file or input from a
pipeline). While in some ways similar to an
editor which permits scripted edits (such as
ed), sed works by making only one pass over
the input(s), and is consequently more
efficient. But it is sed's ability to
filter text in a pipeline which particularly
distinguishes it from other types of
editors.

-n, --quiet, --silent

    suppress automatic printing of pattern
    space

--debug

    annotate program execution

-e script, --expression=script

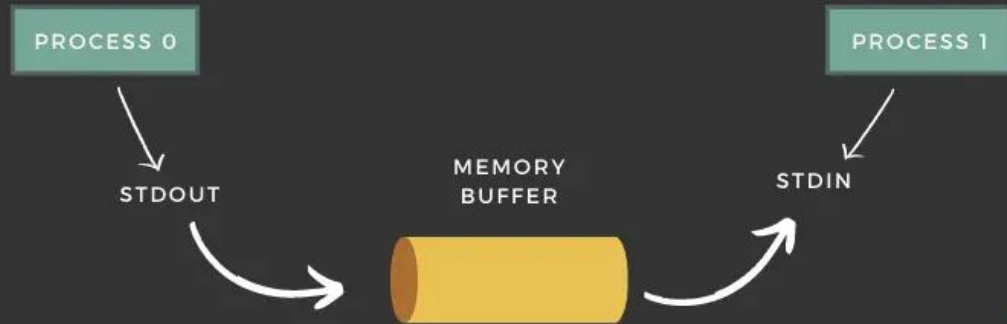
    add the script to the commands to be
    executed

:|
```

ESC → CTRL ALT — ↓ ↑

Redirection & Pipes

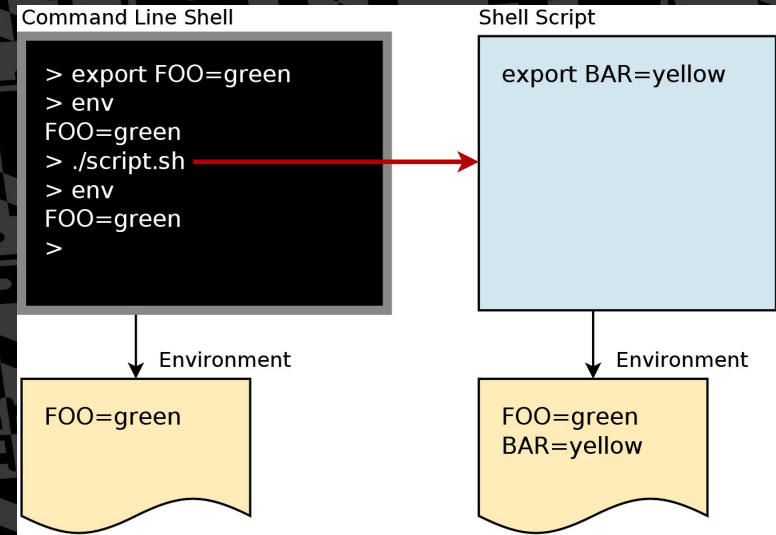
- STDIN & STDOUT
- > redirect to file
- >> append
- | pipe



**PIPE
REDIRECTION
IN LINUX**

High Level concepts

- Variables
 - Local Variables
 - Global (Environment) Variables
 - PATH, HOME, SHELL, USER, PWD
 - Viewing
 - printenv
 - echo \$VARIABLE
 - export \$VAR=123
 - unset \$VAR
- Commands are Programs



Processes

- What about processes?
- How do applications run?
- How are they managed?
- Daemons? What's a daemon



Much more to learn

- Process management (systemd)
- Security practices
- Customizing the environment
- Advanced Networking (network manager, resolved)
- Shell Scripting
- Any Questions?