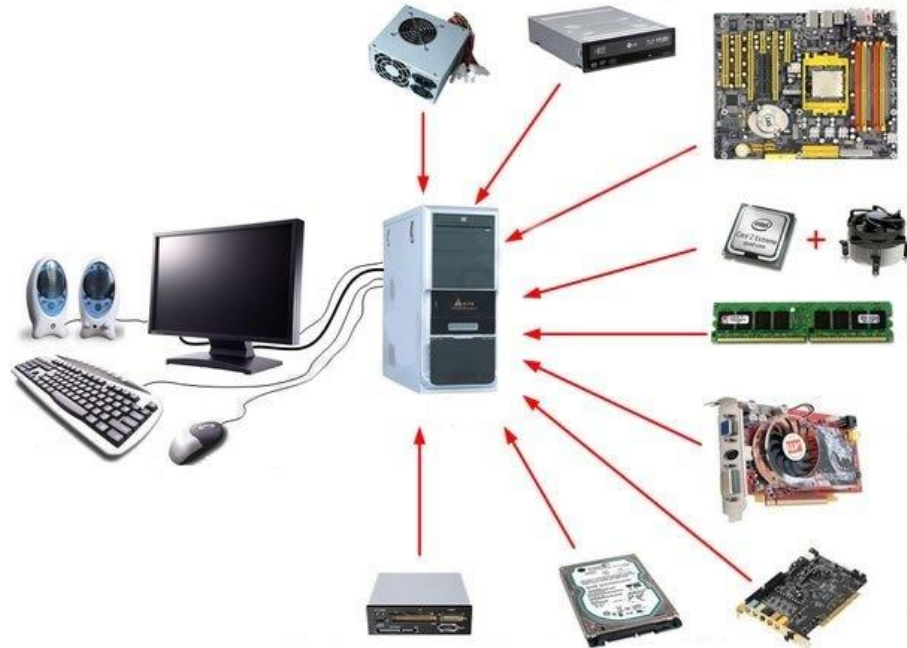


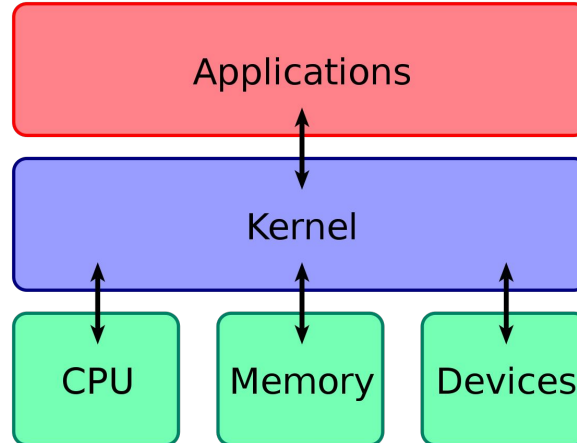
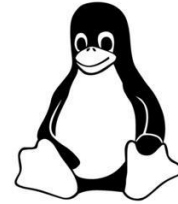
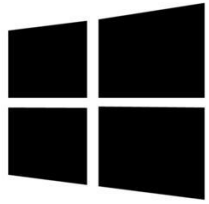
# Shell/Bash notes

Marco Mambelli  
CMS S&C, Fermilab, June 2022

# Hardware



# Operating System (OS) and Kernel



# Terminal



A terminal window with a dark background and a light gray title bar. The title bar contains the menu items: File, Edit, View, Bookmarks, Settings, Help. The terminal content shows the output of the command 'head ~/.bashrc'. The output is as follows:

```
bash-4.3$ head ~/.bashrc
TZ='Pacific/Auckland'; export TZ
if [ -e $HOME/.bash_aliases ]; then
    source $HOME/.bash_aliases
fi

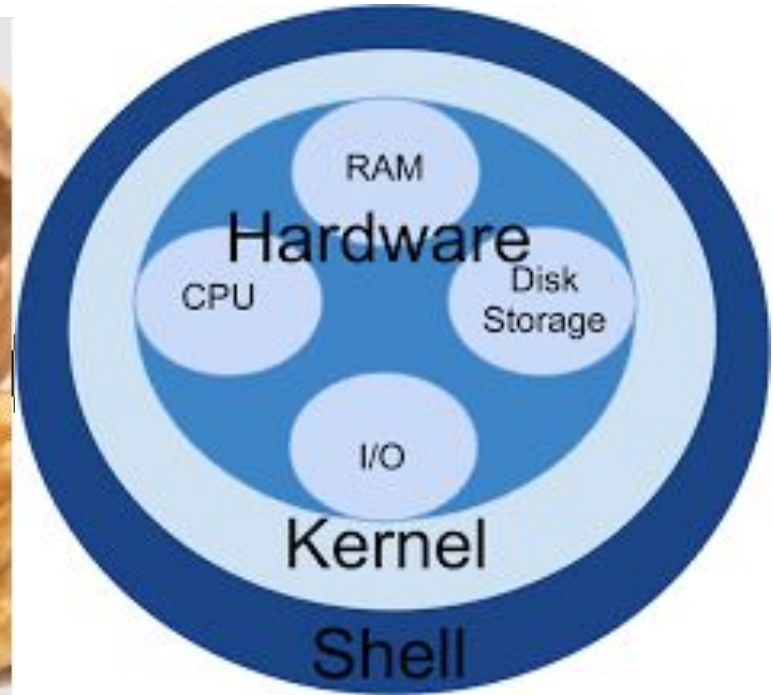
XDG_CONFIG_HOME="$HOME/.config"

# . `which env_parallel.bash`

export LESS="-XR"
bash-4.3$
```

The terminal window has a vertical scrollbar on the right side. At the bottom, there are two tabs: 'code : bash' and 'media : bash', both with a right-pointing arrow icon.

# Shell



# Linux shells

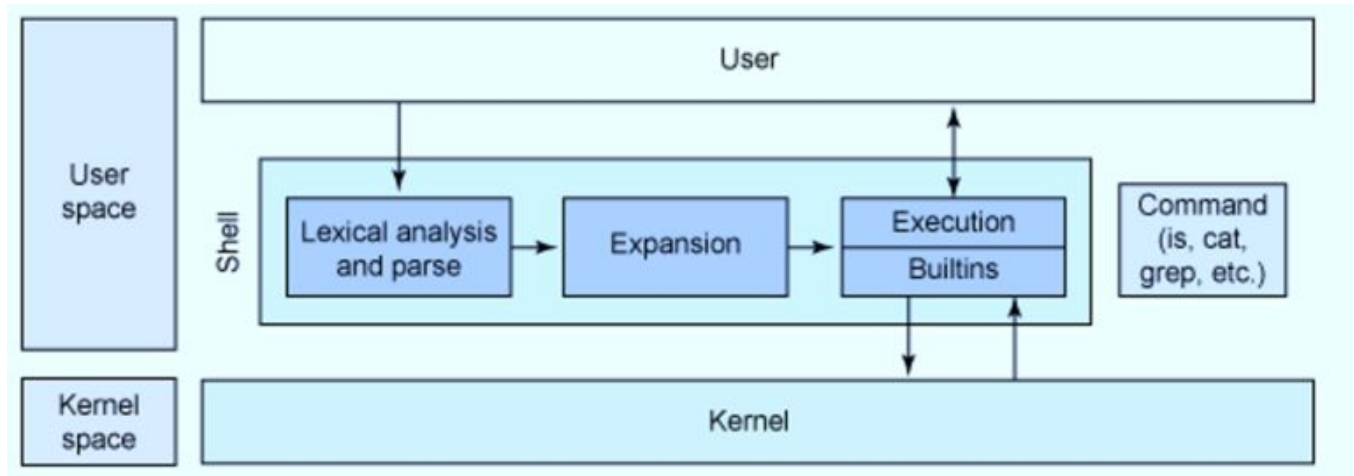
bash (sh)

zsh

ksh

tcsh (csh)

fish



edureka!

<https://www.tecmint.com/different-types-of-linux-shells/>

<https://www.edureka.co/blog/types-of-shells-in-linux/>

# Unix File System

Everything is a file

Regular file

Directory

Link

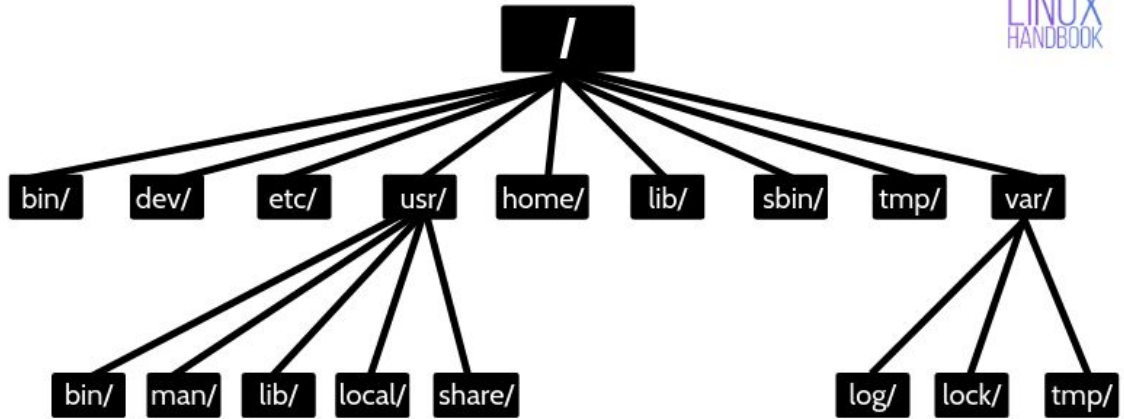
Devices, ...

Standard streams

stdin (0)

stdout (1)

stderr (2)

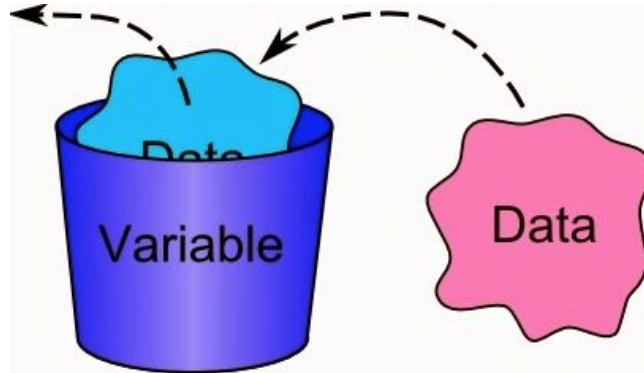


[https://en.wikipedia.org/wiki/Unix\\_filesystem](https://en.wikipedia.org/wiki/Unix_filesystem)

<https://linuxhandbook.com/linux-directory-structure/>

# Variables

used to store information to be referenced and manipulated in a computer program. Variables have a **name**, **value**, **representation**, a **type**



also have a **scope** which is the part of the program where you can access it and a **lifetime** or a duration for which a variable exists



# Bash variables

Untyped \*

\*define -[aif...] var

Names are case sensitive

Parameters substitution

Global environment variables

Shell variables

Local function variables

varname=value

\${var}

define -i numvar

\${varname%.\*}

export varname=value

PATH

SHELL

USER

PWD

HOME

TMPDIR

<https://ostechnix.com/bash-variables-shell-scripting/>

[https://tldp.org/LDP/Bash-Beginners-Guide/html/sect\\_03\\_02.html](https://tldp.org/LDP/Bash-Beginners-Guide/html/sect_03_02.html)

<https://www.cyberciti.biz/tips/bash-shell-parameter-substitution-2.htm>

# Script

File with (bash) code

Shebang (#!)

Variables

Functions

Options and Arguments

Flow control

Output

stdout, stderr

Exit code

```
#!/bin/bash
NAME=""
TIMES=1
usage() {
    echo "Usage: $0 [ -n NAME ] [ -t TIMES ]" 1>&2
}
exit_abnormal() {
    usage
    exit 1
}
while getopts ":n:t:" options; do
    case "${options}" in
        n)
            NAME=${OPTARG}           # If the option is n,
                                     # set $NAME to specified value.
            ;;
        t)
            TIMES=${OPTARG}          # If the option is t,
                                     # Set $TIMES to specified value.
            re_isanum='^[0-9]+$'    # Regex: match whole numbers only
            if ! [[ $TIMES =~ $re_isanum ]]; then # if $TIMES not whole:
                echo "Error: TIMES must be a positive, whole number."
                exit_abnormal
                exit 1
            elif [ $TIMES -eq "0" ]; then # If it's zero:
                echo "Error: TIMES must be greater than zero."
                exit_abnormal          # Exit abnormally.
            fi
            ;;
        :)
            echo "Error: -${OPTARG} requires an argument."
            exit_abnormal            # If expected argument omitted:
                                     # Exit abnormally.
            ;;
        *)
            exit_abnormal           # If unknown (any other) option:
                                     # Exit abnormally.
            ;;
    esac
done
if [ "$NAME" = "" ]; then          # If $NAME is an empty string,
    STRING="Hi!"                  # our greeting is just "Hi!"
else
    STRING="Hi, $NAME!"          # Otherwise,
                                     # it is "Hi, (name)!"
fi
COUNT=1                          # A counter.
while [ $COUNT -le $TIMES ]; do
    echo $STRING
    let COUNT+=1
done
exit 0
```

<https://github.com/glideinWMS/glideinwms/blob/master/factory/tools/gwms-logcat.sh>

<https://www.computerhope.com/unix/bash/getopts.htm>

# Command line

Command, options, [subcommand, subcommand options,] arguments

-o short option

--option long option

--help to get a list of valid options and what they do

man command or info command will also provide information

Tokenization separating the element in the command line

Use quotes (single or double) to avoid separating on spaces

Expansion substituting variables, wildcards, ...

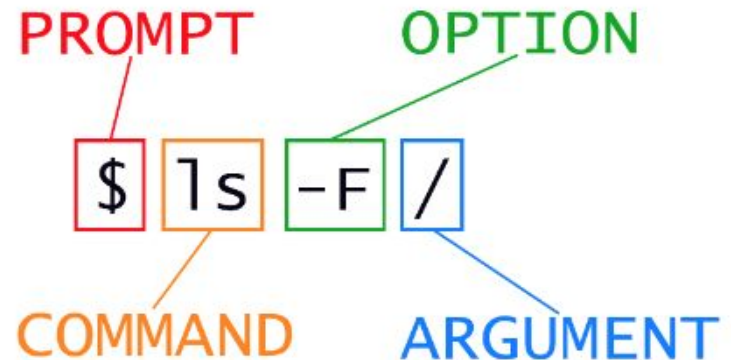
\${varname} variable value

\$(command) output of the command

\$((expression)) output of the math expression

\* ? many or one character

Use single quote to avoid expansion



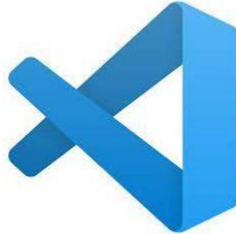
# Text editors



Sublime Text



Brackets



Visual Code  
Editor



Bluefish



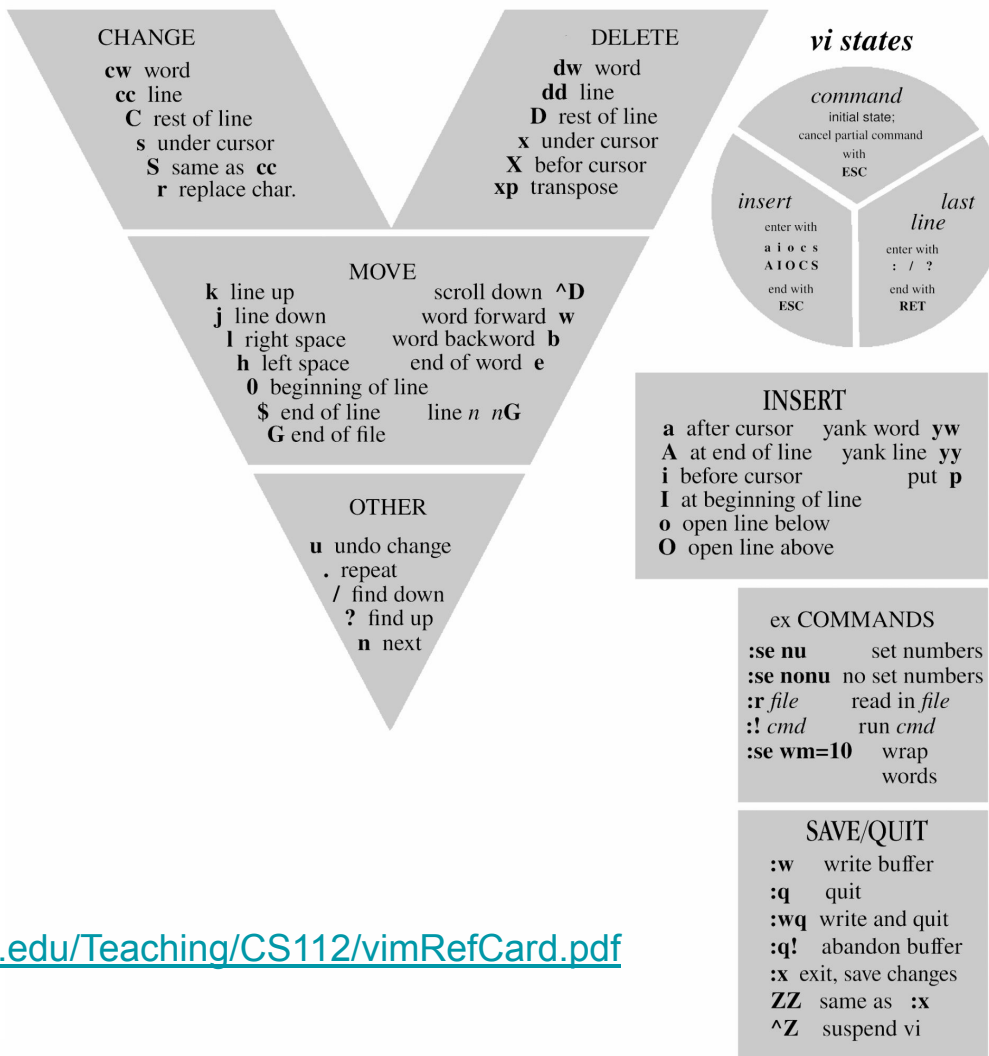
TextMate



TextWrangler

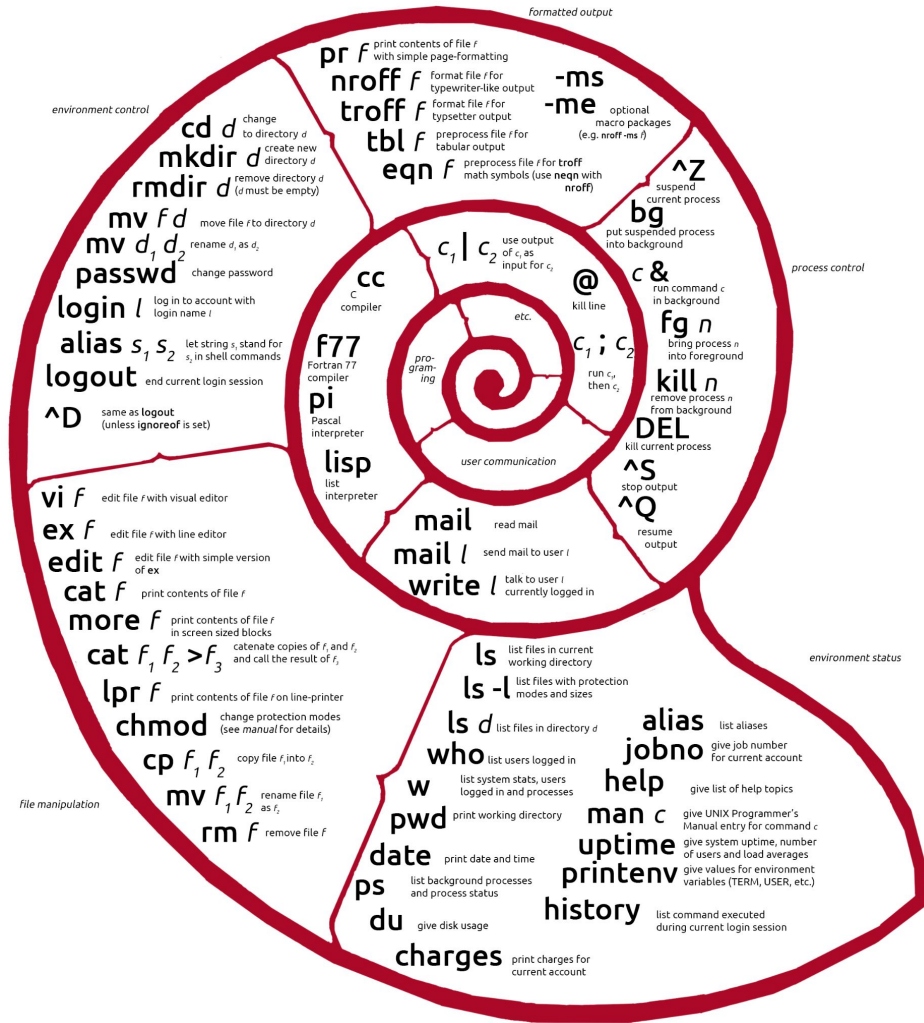
# vi (vim)

## Commands summary



<http://marvin.cs.uidaho.edu/Teaching/CS112/vimRefCard.pdf>

# Unix Shell Commands



Based off a marvelous Unix Poster, if you own the rights to this poster Please Contact me:

Remixed by JaceTheSaltSculptor

UNIX is a trademark of The Open Group

# More about bash

Here a couple of cheat sheets with shell commands:

- [https://appletree.or.kr/quick\\_reference\\_cards/Unix-Linux/Linux%20Command%20Line%20Cheat%20Sheet.pdf](https://appletree.or.kr/quick_reference_cards/Unix-Linux/Linux%20Command%20Line%20Cheat%20Sheet.pdf)
- <https://github.com/LeCoupa/awesome-cheatsheets/blob/master/languages/bash.sh>

And a few more references:

- Classic: <http://www.tldp.org/LDP/abs/html/>
- Break it down: <http://explainshell.com>
- Great guide: <http://wiki.bash-hackers.org/>
- Good to know: <https://mywiki.woledge.org/BashPitfalls>