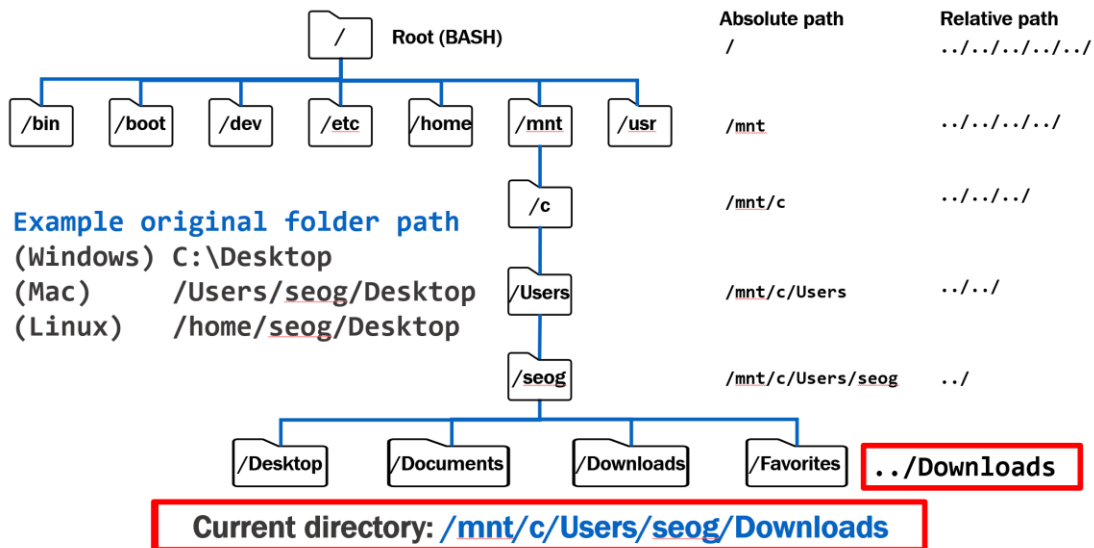


Cheatsheet for: Introduction to BASH session (MMID Coding Workshop)

Folder structure for: Windows Subsystem for Linux, Mac and Linux OS

Windows Subsystem for Linux	/mnt/c/Users/ USERNAME /Desktop
Mac	/Users/ USERNAME /Desktop
Linux	/home/ USERNAME /Desktop

Example Windows Subsystem for Linux BASH folder structure



2022-01-12 Introduction to BASH by Grace E. Seo

24

File information and permission

	File size	Owner	Group	Date & time created	File name
total 0					
-rwxrwxrwx 1	seog	seog	0	Jan 9 23:14	test0001.txt
-rwxrwxrwx 1	seog	seog	0	Jan 9 23:14	test0002.txt

Permission (\$ chmod ###)

No permission	---	---	---
Symbol:	rwx	rwx	rwx
Bit:	421	421	421
Triad:	Owner	group	others

r: readable
w: writable
x: executable

\$ **chmod 755** test0001.txt
Owner(rwx) group(r-x) others(r-x)

\$ **chmod a+wrx** test0001.txt
Owner(rwx) group(rwx) others(rwx)

BASH prompt:

USERNAME@COMPUTER_NAME:/Desktop\$

Useful commands

Commands	Explanation
\$ pwd	Print current working directory
\$ cd PATH	Change directory
\$ cd /PATH/	Change directory (depends on terminal you use)
Creating folder or files	
\$ mkdir FOLDER	Create a new folder
\$ touch FILE	Create a new file
\$ nano FILE.sh	Create a shell script and open text editor
List folder contents	
\$ ls	List files in the folder
\$ ls -l	List files in the folder with long details
Move or renaming or copying a folder or file	
\$ mv OLDNAME NEWNAME	Rename file or folder
\$ mv OLDLOCATION NEWLOCATION	Move file or folder to a new location
\$ cp FILE NEWFILE	Copy FILE and give it a NEWFILE name
Removing folder or file	
\$ rm FILE	Delete a file - irreversible
\$ rm -r FOLDER	Delete a folder recursively - irreversible
\$ rmdir FOLDER*	Delete empty folders that matches to FOLDER name and pattern that matches where wildcard * assigned - irreversible

Text printing, editing or viewing	
\$ echo "TEXT"	Print message inside ""
\$ echo "TEXT" >> FILE	Print message inside "" and store output into a file (>>)
\$ cat FILE	Concatenate files or printout file content
\$ less FILE	View file interactively, press q to quit viewer
\$ nano FILE	Create or edit file in the terminal
Program dependency update and install	
\$ sudo apt-get update	Install updates on Ubuntu program packages
\$ sudo apt-get install build-essential	Install essential program packages on Ubuntu
Copying source files from GitHub repository	
\$ git clone GITHUBLINK	Clone GitHub repository to your local folder i.e. \$ git clone https://github.com/lh3/minimap2
\$ make	Build a program from source file copied to local folder using git clone.
Locating default system bin folders for program execution	
\$ echo \$PATH	Print \$PATH variable to locate system binary folders. Copying executable program files into one of these bin folders will allow you to access program in any directory without using relative path.
Symbols and pattern matching	
>>	Redirect and append output
*	Wildcard is used to define pattern for search and matches
;	Semicolon - command separator i.e. \$ echo "Hi" ; cd Desktop/ Printout "Hi" message first and then move into Desktop folder

Complete BASH script

```
C: > Users > seog > Desktop > intro_to_bash > script.sh
#!/usr/bin/env bash

# Anything after the pound sign become comments
# shebang can also be written as: #!/bin/bash

echo "Welcome to MMID Coding Workshop!" # echo function prints message inside ""

# set variables - use absolute path for directories!
BASE_DIR="/mnt/c/Users/seog/Desktop/intro_to_bash/2022-01-12-Introduction-to-BASH"
DATA_DIR="/mnt/c/Users/seog/Desktop/intro_to_bash/2022-01-12-Introduction-to-BASH/data"
OUTPUT_DIR="$BASE_DIR/script_output"
INPUT="1_control_18S_2019_minq7.fastq"
REFERENCE="reference_MN385595_1.fasta"

# echo $BASE_DIR # print out variable
cd $BASE_DIR # you can also use ${BASE_DIR}
mkdir -p "script_output" # create folder if it doesn't exist -p
echo "hi" # just random word to verify script output
pwd # print script's current working directory

# Run minimap2 and save output result into $OUTPUT_DIR
# Copy and paste previous command used but use variables for input files
# minimap2 -d ../data/reference_MN385595_1.mmi ../data/reference_MN385595_1.fasta
minimap2 -d $DATA_DIR/reference_MN385595_1.mmi $DATA_DIR/$REFERENCE

# minimap2 -a ../data/reference_MN385595_1.mmi ../data/1_control_18S_2019_minq7.fastq > alignment.sam
minimap2 -a $DATA_DIR/reference_MN385595_1.mmi $DATA_DIR/$INPUT > $OUTPUT_DIR/alignment.sam

echo "the end!"
exit
```

2022-01-12 Introduction to BASH by Grace E. Seo

54