Department of Data Science
University of the Punjab

SUMMARY OF LINUX SHELL COMMANDS

Internal commands: Part of shell (`cd, exit, type, help, alias`)
External Commands: Code resides on disk and executed after fork with exec (`clear, ls, passwd, man`)

Basic Shell Commands

Command	Description
echo	Displays text on stdout **-n** don't append \n **-e** enables escape sequences **-E** disable interpretation of backslash escapes (default) **-c** don't produce any more output
help	Provides detail of internal commands
clear	Clears terminal screen.
exit	To close shell
type	Display information about command type (external/built-in)
logout	To close login shell. **Login Shell:** when we login, a particular shell starts execution known as login shell
bash	Bash shell
sh	Bowne shell
csh	C-Shell
kch	Korn Shell
tcsh	Tc-Shell
env	Display environment variables
pwd	Shows absolute address of present working directory
passwd	To change user password
man	To view manual pages of different external commands for better understanding. It has 9 sections. **-k** To search string in all available man pages
who	Shows who is logged in (can be multiple), it also displays "terminal name".
w	Show who is logged on and what they are doing.
whoami	Prints effective username (currently active)
users	Print the user names of users currently logged in to the current host
whatis	Displays command basic purpose (one line description)
whereis	Tells *source*, *binary files* and *man page* file location of external command
which	Gives path of binary file of external and internal command
history	Output the last part of the history list. ***history [-n]***
info	Reads info document of external and internal command
column	Columnize input text **-c** Specify number of columns **-w** Specify columns width (1 to 2048)
ls	List directory contents. **-a** To view hidden files as well **-A** To view hidden files except '.' and '..' **-i** Displays inode number **-h** Displays size in human readable forms in K, M, G instead of bytes **-s** Shows block count before name (in disk files are saved in blocks) **-S** Sort all the files and directory w.r.t their sizes and the first file is largest files in all **-1** List one entry in a line **-f** List files without sorting as they are stored in directory (it also shows '.' & '..' hidden files) **-d** List directories themselves not their contents **-l** Displays files in long listing (7 columns) sorted by names **-t** Sort by modification time (latest first) **-c** Sort by status change time (latest first)

	-u Sort by access time (latest first)
ll	Same as "ls -ls"
touch	Creates 1 or more empty files by touching (updating) modification and access timestamps. If file already exists it updates timestamps: **-m** For updating modification time only **-a** For updating access time only **-c** For updating status change time only
file	display type of file
local	used to declare a variable to be local to a bash function ***local [-OPTION] [name[=value]]***
read	read a single line from stdin
set	sets shell variables ***set [-OPTIONS] [arguments]***
test	Evaluate condition(s) or make execution of actions dependent on the evaluation of condition(s) ***test['condition']['condition']***
expr	Evaluate arguments as an expression: **expr `arguments`**
cat	To view contents of a simple file on stdout **-n** To print line numbers as well **-s** To suppress repeated blank lines **-b** To number only non-empty lines (overrides -n)
tac	To view contents of file in reverse (last line 1st)
more	To view contents of large files one screen at a time. It also displays % of file displayed and we can't move back up in it. **ENTER** To move down line-by-line **SPACE** To move down one screen **/str** To search *"str"* in file. Press 'n' to find next Press 'N' to find previous
less	To view contents of large files one page at a time but much better than *"more"*. **Navigation** Arrow keys, Pgup, Pgdwn, ENTER, SPACE (acts as Pgdwn), HOME, END **/str** To search *"str"* in file. Press 'n' to find next Press 'N' to find previous **g, G** 'g' moves to start and 'G' moves to end
head	Displays 1st ten lines **-n** To view 1st n lines
tail	Displays last ten lines **-n** To view last n lines **-f** Output appended data as the file grows **-c** specifies that we want to read n characters not lines
alias	User defined names for commands (arguments are also allowed in alias) *alias cls="clear"* #makes "cls" an alias for clear command *alias* #list all aliases
unalias	To remove alias **-a** Remove all alias definitions
cal	To display calender **-h** Won't highlight current-date. **-m** Display the specified month **yyyy** Display a calendar for the specified year (e.g. *cal 2017)* **-1** Displays only the current month. **-3** Display the previous, current and next month.
date	To display and change (only for root) date ***[day MON dd mm:hh:ss PKT yyyy]***
shutdown	To shut-down or restart **shutdown now** Shut-down immediately **shutdown -r now** Restarts immediately **shutdown +0** Shut-down immediately **shutdown +m** Shutdown after m minutes ('+' is optional) **shutdown 22:30** Shut-down at 22:30

cp	To copy files/directories **-p** Preserve permissions while copying (by default permissions can change) **-r** For directories **e.g:** *cp f1 f2* #f1 is source file and f2 is target-file
rm	To remove files/directories **-f** ignore non-existent files and arguments, never prompt **-r** For directories **-i** For confirmation prompt **e.g:** *rm f1 f2* #will delete both f1 & f2
mv	To move files/directories **-i** For confirmation prompt **e.g:** *mv f1 f2* #will move f1 to f2 (it is also used to rename file)
mkdir	To make directory file **-m** set file mode (as in chmod) **-p** no error if existing, make parent directories as needed
rmdir	To remove directory file **-p** remove DIRECTORY and its ancestors; e.g., 'rmdir -p a/b/c' is similar to 'rmdir a/b/c a/b a'
sort	Gets input from stdin and output it on stdout after sorting **-b** Ignore leading blanks **-r** for reverse order **-t** for specifying delimiters (e.g.: -t";") **-kn** to sort by column n **-n** for numeric sort **-c** check for sorted input; do not sort **-d** Dictionary order **-f** fold lower case to upper case characters **-i** consider only printable characters **-g** compare according to general numerical value (general numeric sort)
length	a string operation to return the number of characters stored in a string
evince	To view PDF and other common document formats
time	Run programs and summarize their system resource usage (shows runtime in seconds). **real** Total execution time **user** Time spent in user space **sys** Time spent in kernel space
lsb_release	Shows basic OS Info. **-a** Shows all OS details (must be used)
uname	Prints OS name on stdout **-a** Shows detailed OS info
lscpu	shows detailed CPU specs
readelf	to reaf ELF files (.o and .out) **-a** shows all info **-s** shows symbol table (.symtab) **-S** shows section header **-h** shows ELF header **-l** shows program header
lpr	Line printer prints the contents of specified files to printer
bc	Command line calculator
script	Make typescript of terminal session
wc	print number of lines, word, char counts for each file (Ctrl+D to quit) **-l** for lines only **-w** for words only **-m** for character count only **-c** byte count
cut	Display selected fields (**-f** for fields, **-d**"delimiter"). Default delimiter is TAB **e.g:** cut -d":" -f1-3,5 passwd (column 1,2,3,5)
paste	horiontally concatenate files (Seperated by TAB)
grep	("General Regular expression Processor") Print lines matching or not matching a pattern. **-i** for case-insensitive search **-v** for negation

	-c print count of lines matching/not matching (for -v)
uniq	Report or omit consecutive repeated/duplicate lines. **-c** gives line count **-u** for showing only unique lines **-d** for showing only duplicated lines
mesg	Permit or deny messages ***mesg [-y/-n]***
split	Split a file into multiple files. Output pieces of FILE to PREFIXaa, PREFIXab, ...; default size is 1000 lines, and default PREFIX is 'x'. With no FILE, or when FILE is -, read standard input. **split [OPTION]... [FILE [PREFIX]]** **-b SIZE** put SIZE bytes per output file **-C lines** put at most SIZE bytes of records per output file
For Comparison and Searching	
comm	Compare 2 sorted files line by line 1st column unique to File1 2nd column unique to File2 3rd column COMMON in both **-1** suppress column 1 (lines unique to FILE1) **-2** suppress column 2 (lines unique to FILE2) **-3** suppress column 3 (lines that appear in both files) **--nocheck-order** do not check that the input is correctly sorted
cmp	Compare 2 files byte by byte and stops at first difference **-l** for not stopping on 1st difference (byte values are in octal) **Note:** All remaining bytes will be different after 1st byte in files
diff	Compare files line by line **e.g:** *diff f1 f2* #(I want to make f1 similar to f2) **c** change **a** append **d** delete **<** is for 1st file **>** is for 2nd file **=** is for common lines (in both files) ***diff -c file1 file2 >new.patch*** To save differences as a patch file to update 1st file to match 2nd file
locate	To find all location of files by specified name in DB (it don't search in directory hierarchy)
sudo updatedb	to update file DB used by "locate" (updated once per day implicitly)
find	search for files in directory hierarchy **-name** Finds by name **-size** Finds by file size (k=Kilobytes, M=Megabytes, G=Gigabytes) **-atime** access time **-ctime** status change time **-mtime** modification time **type** (**f** = normal files, **d** = directories, **s** = sockets, **p** = named pipes, **b**=block, **c**=character, l=soft-link) **EXAMPLES** find ~ -mtime 1 Finds files that are modified 1 day ago find ~ -mmin 10 Finds files that are modified 10 min ago find . type f | wc -l Find in the PWD, all the files whose type is regular file and give their count find / -perm /7000 2>/dev/null for viewing all files with special permissions
For Archiving	
tar cvf	Create tar file in PWD (1st pass name for archive file then directories and files to archive)
tar tvf	To view .tar files not extract them
tar xvf	To extract .tar files in present working directory
tar xzf	To unzip and extract .tar files in present working directory
gzip	To zip files. Original file is replaced by zip file. (extension = .gz)

	Note: We can zip tar files to obtain "tar balls"(.tar.gz), commonly used for software distribution
gunzip	To unzip files
IPC	
tee	it reads from stdin and writes to stdout and file(s) • It takes all arguments as output file • It doesn't take any input file without input redirection
mkfifo	it creates named pipes (only)
mknod	• it can create named pipes (p). • block special file (b) and • character special file (c) ***mknod -m 0666 file_name type maj min*** type=b,c,p (block,character,pipe)

Department of Data Science
University of the Punjab

Vim Editor

Command	Description
sudo apt-get install vim	To install vim editor
vimtutor	For detailed vim lessons
vim +	For opening file in append mode (cursor at last line)
vim +n	Cursor at start of line-n
vim +/string	Cursor on line with 1st occurence of "string"
ESC	Command mode
ESC+:	Last-line mode
q	To quit vim
i	Start typing before current character
I	Start typing from beginning of current line
a	Start typing after current character
A	Start typing from end of current line
o	Open new line below current line
O	Open new line above current line
h	To move cursor left
L	To move cursor right
K	To move cursor up
j	To move cursor down
gg	To goto 1st line
GG	To goto last line
End, $	Moves to end of current line
Home, 0	Moves to start of current line
Shift+G	To put prompt at the end of document
u	For undo
Ctrl+r	For redo
ESC+[/,?]	Then write string to search. **/** For forward search **?** For backward search **n** Find next **N** For finding in opposite direction
dw	For deleting a word
[n]dd	For deleting a line
[n]yy	For copying line
[n]p	For pasting n times below current line
[n]P	For pasting n times above current line
!	In last-line mode after command to override warning
:wq	in last-line mode to "save & quit"
:w!	to "save" and override warning
:w [filename]	To "save as"
:q!	To quit
:e!	To undo changes since last save
:[n]	To move to nth line
:$	To move to end of the file
:n1,n2[d,y]	To delete or copy a range of lines. (d= delete, y=copy) **:3,6d** delete lines 3-6 **:3,$y** copy from lines 3-end **:9,15y** copy lines 9-15
:1,$ s/search/replace	It will replace only one occurrence in each line of "search" with "replace"

:1,$ s/search/replace/g	It will replace all occurrences in each line of "search" with "replace"
:set number	To display line numbers
:set nonumber	To remove line numbers
:!command	To execute shell command in last-line mode inside vim editor (will execute only 1 command)
:sh	This will open new bash process to execute commands
For Multi-Filing	
:n	To move to next file
:N	To move to previous file
vim -o filenames	It will open files in multiple horizontal windows
vim -O filenames	It will open files in multiple vertical windows
Ctrl+ww	To move onto next file (if pressed in last file then moves to 1st)

File Management

Command	Description
lsattr	View extended file attributes
chattr	Change extended file attributes **chattr +/-[attr] file**
ln	For creating links. **-s** For soft-links
tty	Display the name of terminal you are using
xterm	to launch ptmx terminal
stty	To change and print terminal line settings. ***stty attribute value*** **-a** To view all attributes **stty sane** To reset all attributes to their defaults. **-isig** To off signals on terminal **-g** To save terminal settings **stty -echo** Turns off terminal echo **stty echo** Turns on terminal echo
File Permissions Management	
chown	Change user owner **chown :group file** (for changing grp owner using chown)
chgrp	Change group owner
chmod	For changing permissions If we write "chmod +r" r will be assigned to u/g/o
umask	To view/change umask **-S** To view complement of umask in symbolic way
getfacl	Foreach file, getfacl displays the file name, owner, the group, and the Access Control List (ACL). If a directory has a default ACL, getfacl also displays the default ACL.
setfacl	Sets file ACL **-m** add entry **-x** remove entry **-b** clear acl and delete all entries **-d** add default entry **-R** add recursive entry

Department of Data Science
University of the Punjab

Process Management

Command	Description
jobs	Lists currently running jobs and their status **-l** lists process IDs in addition to the normal information **-p** lists process IDs only
fg [pid] **fg %Jid**	To move a background process to foreground
bg [pid] **bg %Jid**	List background jobs or move a process to bg
kill	Send a signal to a job. (default SIGTERM) **kill [-s sigspec | -n signum | -sigspec] pid | jobspec** **kill -l [sigspec]** **-s sig** SIG is a signal name **-n sig** SIG is a signal number **-l** list the signal names; if arguments follow `-l' they are assumed to be signal numbers for which names should be listed
ps	Report a snapshot of current process (4 columns) **-A or -ax** to show all running process **-u [username]** List processes by user (displays 11 columns) **-l** displays 14 columns (long listing) **-a** Select all processes except both session and processes not associated with a terminal.
top	Shows detail process real-time info of top-20 processes, like task manager. Interactive, continuously (refreshes after every **3sec).** **Press:** **h** for help **n** to display only [n] processes (0=unlimited) **u** to display processes of particular user **s** to change refresh time **k** to send signal (it 1st asks for PID then signal number/name)
free	Displays amount of free and used memory in the system (6 columns) **-k** in KB (default) **-m** in MB **-b** in bytes **-g** in GB **--tera** in tera
vmstat	displays info about virtual memory (6 groups, 17 columns)
uptime	It shows system time, uptime, number of logged in users, load average for last 1,5 and 15 minutes respectively.
watch	Executes a program periodically, showing output in full screen (refreshes every 2sec)
halt	To halt the system.
nice	To run a command with specific NICE(-20 -> 19) value. Only root can use negative nice values. **-n** add integer N to the niceness (default 10)
renice	Alter priority of running processes. **renice [-n] priority [-g|-p|-u] identifier** **-n** Specify the scheduling priority to be used for the process, process group, or user. When used, it must be the first argument. **-g** Interpret the succeeding arguments as process group IDs. **-p** Interpret the succeeding arguments as process IDs (the default).

	-u Interpret the succeeding arguments as usernames or UIDs. The following command would change the priority of the processes with PIDs 987 and 32, plus all processes owned by the users daemon and root: **renice +1 987 -u daemon root -p 32**
schedtool	Query and set per-process CPU Scheduling parameters ***sudo apt-get install schedtool*** **-r** lists scheduling policies • N: SCHED_NORMAL (prio_min 0, prio_max 0) • F: SCHED_FIFO (prio_min 1, prio_max 99) • R: SCHED_RR (prio_min 1, prio_max 99) • B: SCHED_BATCH (prio_min 0, prio_max 0) • I: SCHED_ISO (policy not implemented) • D: SCHED_IDLEPRIO (prio_min 0, prio_max 0) -n For changing nice value -p for changing Static Priority -a for changing affinity value -e to execute command with different scheduling parameters • schedtool PID • schedtool -[POLICY Letter] PID • schedtool -a [affinity in HEX] PID e.g: ***schedtool -a 0x1 3199*** (0x1,0x2,0x4,0x8,...) • schedtool -n [NICE Val] PID • schedtool -p [Static PRI] PID #for -R -F • For -R -F Static Priority should be mentioned with Policy and in sudo mode

User Management

Command	Description
visudo	used to edit /etc/sudoers file
adduser	More interactive and recommended (***sudo adduser user1***)
useradd	It is low-level command to add user. And we need to give some extra info as well. Its minimum requirements are: **-m** to make directory of that user **-d** To specify directory path (/home/username) **useradd -m -d /home/user2 user2**
deluser	User we want to delete should be logged out. It don't deletes user HOME Dir
userdel	Low level also deletes HOME directory and files. **-r** to delete home dir and associated files as well of this user. ***sudo userdel -r user1***
usermod	To modify user info. **e.g.: *usermod -a -G gp2 user1*** (makes user1 member of gp2) • If we don't use -a then it will not append new user but overwrite it (that is all previous group members will be removed) -c to change personal info column value **sudo usermod -c "Personel Info" user2** -s to change default user shell **sudo usermod -s /bin/sh user2** -l to change username **sudo usermod -l user007 user2** (new name 1st) -d to change Home Directory -L to lock user (this user can't log in) -U to unlock locked user -g to change primary group -G to change secondary group -a, --append Add the user to the supplementary group(s). Use only with the -G
groupadd	To add new group. (***sudo groupadd gp1***)
groupmod	To modify group. **-n** is used for changing group name.
groupdel	To delete group. (***sudo groupdel name***)
chage	Used to change password expiry info of a user (***sudo chage user2***). **-l** to view just password setting of particular user
chsh	Used to change default user shell
chfn	Used to change user personal info
finger	shows user info in detail (may have to install it manually)
id	it displays ID (UID) and primary GIDs and groups you belong to gid=primary group, groups = Secondary group
su	(switch user) We can use it to login using any username if we know its password (e.g: su -root) • Using '-' will also give you the target user environment. You will find yourself in the target user HOME Directory and his default login shell

Disk/Memory Management

Command	Description
	Disk Formatting
man fs	Man page for all commonly used file-systems and their characteristics.
mkfs	to build filesystem (format partitions) • its use is deprecated now there are commands for each FS like *mkfs.ext*, *mkfs.ext2* etc. ***mkfs.<FS_Name>*** • There should not be any data on partition we want to format if there is take its backup
mount	To mount a partition. **mount -t type device /dir** ***mount /dev/sda3 /opt*** (mounts /dev/sda3 to /opt)
unmount	To unmount a partition. (***umount /dev/sda3***)
lsblk	Lists info about all available block devices (sda, sr0) • It shows 7-columns by default • To view only selected columns, use -o then enter names of columns (, seperated) **lsblk /dev/sda** (List info about /dev/sda only.) **lsblk -o name,type,fstype,parttype,size,mode /dev/sda**
	FS Architecture
e2label	Changes label on ext2/ext3/ext4 file systems. **e2label /dev/sda3 "anas3"** To assign label **e2label /dev/sda3 ""** To unassign label
tune2fs	Shows super block info in detail: **tune2fs -l /dev/sda1**
stat	Lists all i-node block info of a file or device e.g.: inode, permissions, times, size, owners etc. ***stat filename/device*** *stat /etc/passwd* *stat -f /dev/sda1* (**-f** shows info about device e.g: /dev/sda1)
df	(Disk free) Displays amount of diskspace available on partition/FS ***df -i devices*** If no devices are mentioned then list info for all active partitions. **-i** shows info about inodes
du	(Disk usage) Displays how much space a particular file or directory has occupied. **-h** shows size in human readable form K, M, G **du ~** Recursively shows sizes of all files, dir, sub-dirs inside ~
lsof	(List of opened files) System Wide File Table ***lsof -p PID*** (list files opened by PID only)
fuser	• Identify process using files or sockets. • Used to list PIDs and usernames of processes using a specific file **-u** To show username as well ***fuser -u /etc/passwd***
	Disk Partitioning

partx	Tell the kernel about the presence and numbering of on-disk partitions ***partx --show /dev/sda*** (List all partitions on Disk)
dd	Copy a file, converting and formatting according to the operands. ***dd if=/dev/sda bs=512 count=1*** (Shows contents of zero sector but not human readable) • It will read file /dev/sda and (if=/dev/sda) • reads just 512 bytes and (bs=512) • read once and show them (count=1)
hexdump	• The hexdump utility is a filter which displays the specified files, or the standard input, if no files are specified, in a user specified format. • Makes content of dd readable. **-C** Display the input offset in hexadecimal, followed by sixteen space-separated, two column, hexadecimal bytes, followed by the same sixteen bytes in %_p format enclosed in ``|'' characters. ***dd if=/dev/sda bs=512 count=1 | hexdump -C*** (Shows zero sector in readable HEX format)
fdisk	Manipulate disk partition table (interactive program) **-l** shows info about all the block devices and their partitions • ***fdisk -l /dev/sda*** (shows info about sda only) • ***fdisk –version*** (to check fdisk version) • To run fdisk use ***"fdisk /dev/sda"*** as root/sudo **m** For help. **p** Displays partition table **d** To delete partition **n** To create new partition p=primary,e=extended then write partiton digit (1-4) 1st sector (use default) last sector or size (we should use size in human form by proceding with +) e.g: +2G (default for last partition is all remaining disk space) **q** quit without saving changes **w** write table to disk and make changes permanent (use it carefully)

Department of Data Science
University of the Punjab

System Programming Commands

Command	Description
make	The make utility will determine automatically which pieces of a large program need to be recompiled, and issue the commands to recompile them. **make [OPTION]... [TARGET]...** **-f** To specify name of makefile to search for **-n** To tell make to print out what it would have done w/o actually doing it **-k** Tells make to keep going when an error is found, rather than stopping as soon as the first problem is detected.
ar	The GNU ar program creates, modifies, and extracts from archives. An archive is a single file holding a collection of other files. **ar -rcs libmymath.a myadd.o mysub.o mydiv.o mymul.o** **-r** Create a new archive **ar -r libfirst.a file1.o file2.o** **-q** Append an object file to an existing archive. **ar -q libfirst.a file3.o** **-d** delete object modules from an existing archive **ar -d libfirst.a file2.o** **-x** extract object modules in your PWD **ar -x /usr/lib/libm.a** **-t** display table of contents of an archive **ar -t /usr/lib/libm.a** **-c** Without it if an archive is not already existing then a warning will be displayed. **-s** To maintain files in particular order w.r.t to functions to avoid errors
ranlib	ranlib utility generates an index to the contents of an archive and stores it in the archive.
ldd	List dynamic dependencies displays the shared libraries that an executable (or a shared library) requires to run.
ldconfig	Conf igure dynamic linker run time bindings. Creates necessary links to the most recent shared library verions
objdump	This command can be used to obtain various information, including disassembled binary machine code from an executable file, compiled object or shared library. **-d** To disassemble
nm	This command lists the set of symbols def ined within an object library or executable program
objcopy	Copy and translate object files.
addr2line	Convert addresses into file names and line numbers
gcc	GNU Compiler: **gcc [options] file-list** **-o** Specify the name of executable file (default a.out) **-save-temps** To save all intermediate files: (*.i, *.s, *.o, a.out) **-E** Perform preprocessing only and generate file with .i extension **-S** Generate Assembly code with .s extension for the specific processor **-c** Suppress linking phase and keep object files with .o extension **-static** To force static linking **-lxxx** All libraries except std I/O, need to be explicitly linked with -l option.

	-Lpath By default, linker looks for libraries in /usr/lib/x86_64/ and /lib/ directories. If you want to link libraries located somewhere else, use -L option **-Ipath** By default, preprocessor first searches for include files in directory containing the source file, then in the directory named with -I option to gcc, and finally in /usr/include/ or /usr/include/c++/4.1.1
gdb	GNU Compiler we can also specify exe file with it to load it at startup, then we won't have to use *file* command **-tui** to open gdb in ncurses-interface mode (default 2 panels {code,command})
ulimit -c unlimited	To generate core file in case of abnormal termination. ***gdb -q ex2 core***

GDB Debugger Commands

Command	Description
file	To load program in GDB
attach	to load already running program in gdb using PID
run	to execute loaded program
info registers	to view contents of memory registers
info all-registers	to view contents of all memory registers
info inferiors	to view all current gdb session inferiors (loaded programs) • **inferior** is used by GDB to manage all loaded programs. Each inferior has a number assigned to it.
add-inferior	to add new inferior (load another program) ***add-inferior -exec a.out***
info break	to view all breakpoints in focused inferior
list	to view source code inside GDB (it also displays line number)

help	to get help inside gdb
	o It shows 12 classes of commands
	`arif@arif:~/gdb$ gdb -q`
	`(gdb) help`
	`List of classes of commands:`
	`aliases -- Aliases of other commands`
	`breakpoints -- Making program stop at certain points`
	`data -- Examining data`
	`files -- Specifying and examining files`
	`internals -- Maintenance commands`
	`obscure -- Obscure features`
	`running -- Running the program`
	`stack -- Examining the stack`
	`status -- Status inquiries`
	`support -- Support facilities`
	`tracepoints -- Tracing of program execution without stopping the program`
	`user-defined -- User-defined commands`
	o ***help all:*** to view all commands in gdb
	o ***help command:*** to view info about a gdb command
	o ***help class:*** to view commands inside a gdb commands class

disassemble	used to dump assembly of specified function in AT&T format
backtrace	used to get info about function stack frames (FSF
finish	It completes execution of current function, returns value to parent function and stop there after copying address of next instruction from FSF to rip.
layout-split	to view an addition assembly code panel in -tui interface mode of gdb
Breakpoints	
break	break command is used to set breakpoint **break 10** sets breakpoint at line-10 **Break main** sets breakpoint at 1st line of function main
next/n /ENTER	executes only next instruction (of HLL code) and if that instruction contain function call it will also execute that function code implicitly
continue/c	executes program till end or next breakpoint
ni/si	moves to next instruction of assembly code
disable	to disable a breakpoint temporarily by specifying its number ***disable 2*** #disables breakpoint with number-2

print	to view contents of a variable during execution (at breakpoint) **print /x i** #displays value in HEX **print /o i** #displays value in Octal **print /t i** #displays value in Binary **print i** #displays value in datatype format
whatis	to view datatype of variable
set	to change variable value at breakpoint. It has 2 syntax: o **set (i=10)** o **set variable i=10** o **set $rax=9** **#**to change register values precede name with **$**

Department of Data Science
University of the Punjab

Commonly used Git Commands

Command	Description
git init	Initializes local git repository in PWD
git clone <Link>	For cloning remote repository for the 1st time
git pull <Link>	Used after clone
git remote rm name	Removes remote repository named "name"
git remote add name <Link>	Adds remote repository named "name"
git push origin master	Pushes master branch to remote repository origin
git branch <name>	Creates new branch named "name"
git branch -l	List all branches
git checkout <branch>	Switches to specified branch
git checkout -b <name>	Creates new bracnch and switches to it
git branch -m <old> <new>	Rename a branch
git branch -d <branch>	Deletes a branch
git branch -D <branch>	Delete a branch with unmerged changes
git push origin <Branch_Name>	Pushes specified branch to remote repository origin
For COMPARISON	• git diff <file> # with staging index • git diff HEAD <file> # with local repo • git diff --satged <file> # cmp file in staging index with local repo • git diff <b1> <b2> # cmp t2o branches

PATCHING	• ***diff -c file1 file2 >new.patch***
	  • the differences are identified such that the first file could be modified to make it match the second file
	• ***patch -i new.patch***
	  • in dir of file1 to change this to file2

Config Files	• System: /etc/gitconfig • User: ~/.gitconfig • Project: ProjectDir/.git/config
HEAD	• ./.git/HEAD (tell branch) • ./.git/refs/heads/master (master=branch) • .git/refs/remotes/origin/master (remote master HEAD)
Excluding files	• Project Level: .git/info/exclude • Directory Level: ./.gitignore