Pandit Deendayal Energy University ,Gandhinagar
School Of Technology

Department Of Computer Science And Engineering

OPERATING SYSTEM LAB

(20CP207P)

Name: Kunal Kumar Sahoo
Roll No: 21BCP100
Semester: IV
Division: 2-G3
Branch: Computer Science Engineering

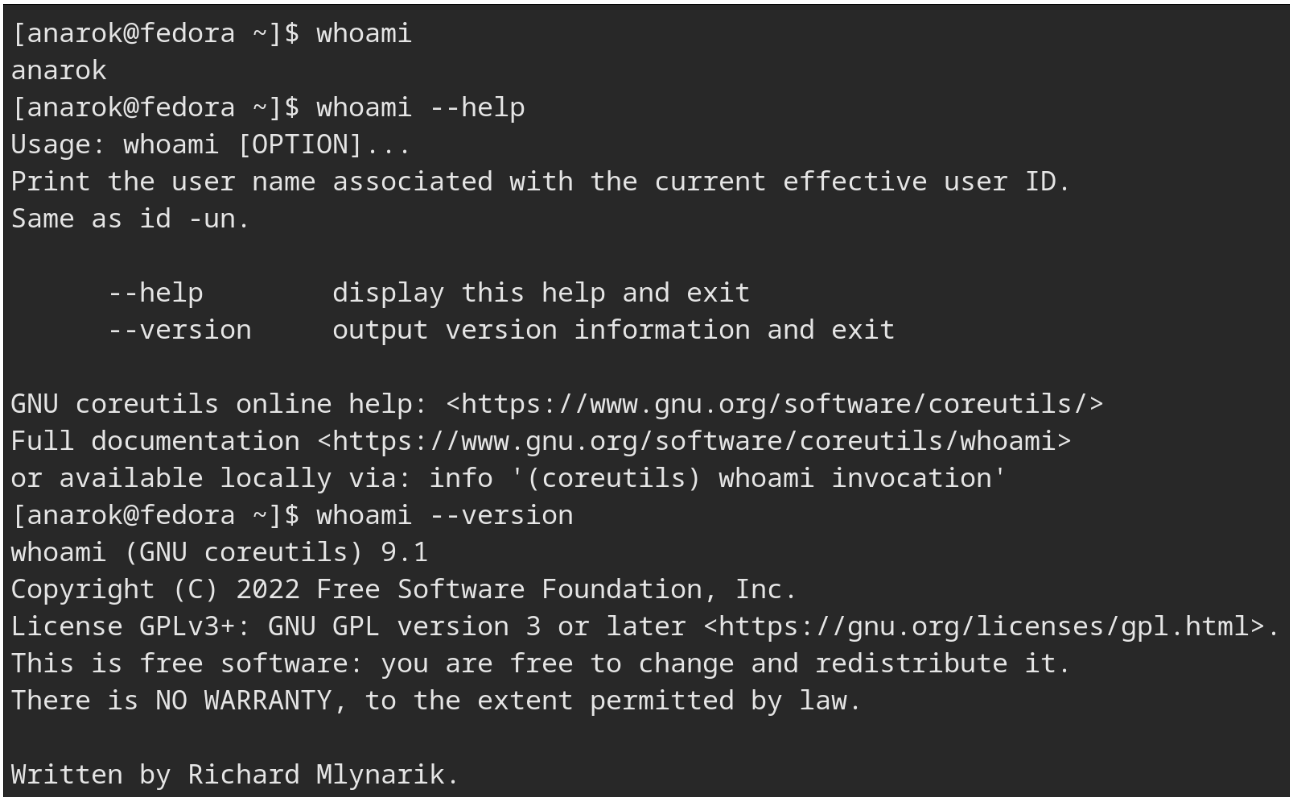
INDEX

Exp. No.	List of Experiments	Date	Page	Signature

Experiment-1: Basic Terminal Commands

1) **whoami command:**

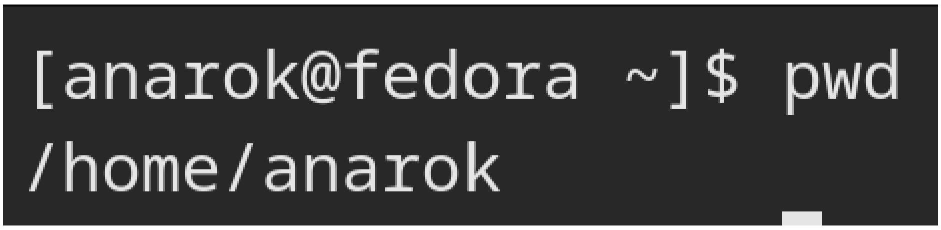
Displays user, group and privileges information for the user who is currently logged on to the local system. If used without parameters, whoami displays the current domain and user name.



```
[anarok@fedora ~]$ whoami
anarok
[anarok@fedora ~]$ whoami --help
Usage: whoami [OPTION]...
Print the user name associated with the current effective user ID.
Same as id -un.

      --help        display this help and exit
      --version     output version information and exit

GNU coreutils online help: <https://www.gnu.org/software/coreutils/>
Full documentation <https://www.gnu.org/software/coreutils/whoami>
or available locally via: info '(coreutils) whoami invocation'
[anarok@fedora ~]$ whoami --version
whoami (GNU coreutils) 9.1
Copyright (C) 2022 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <https://gnu.org/licenses/gpl.html>.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Written by Richard Mlynarik.
```

2) pwd command:

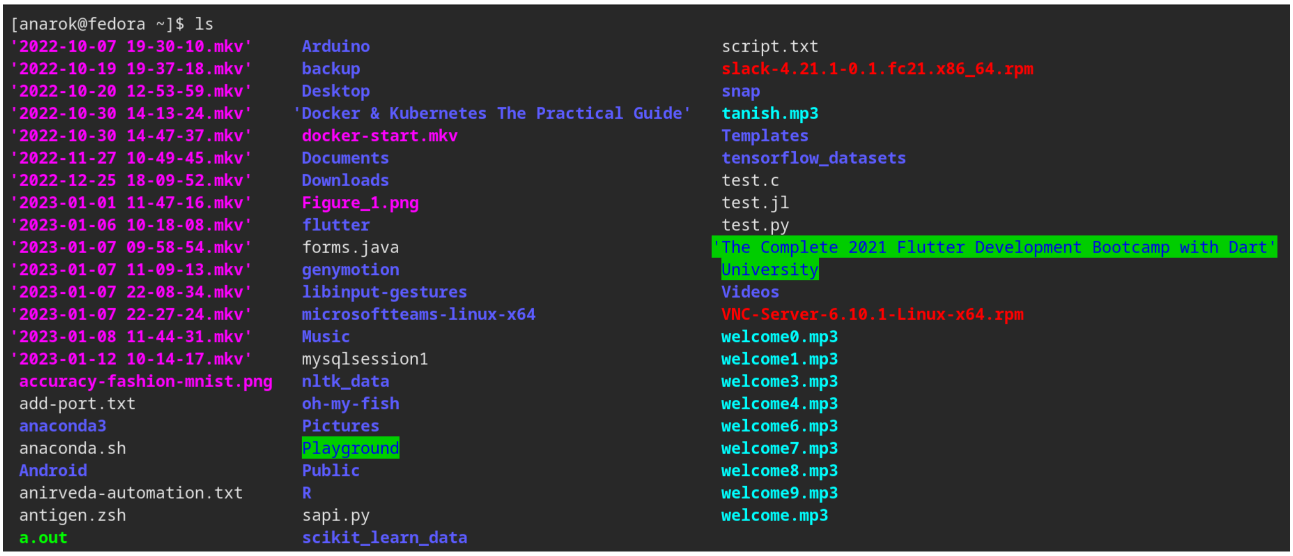
The pwd command writes to standard output the full path name of your current directory (from the root directory). All directories are separated by a / (slash).



```
[anarok@fedora ~]$ pwd
/home/anarok
```

3) ls command:

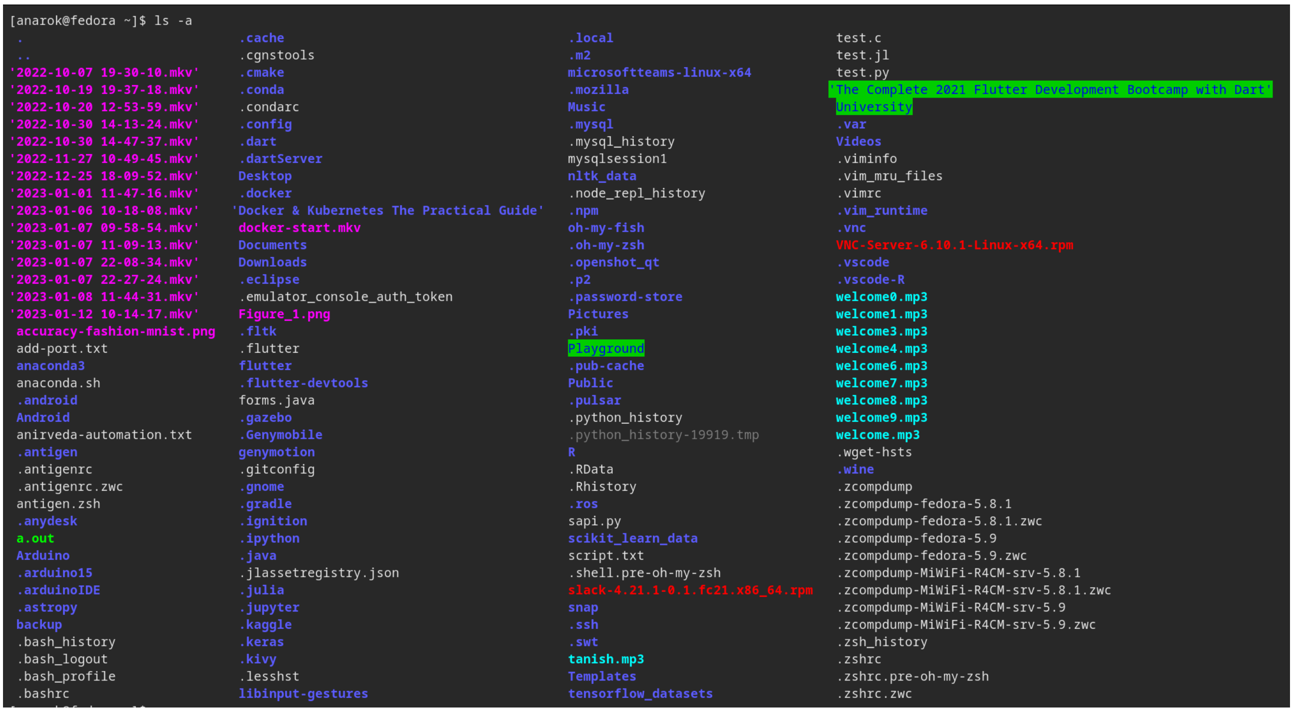
The ls command is **used to list files.** "ls" on its own lists all files in the current directory except for hidden files.



```
[anarok@fedora ~]$ ls
'2022-10-07 19-30-10.mkv'     Arduino                                    script.txt
'2022-10-19 19-37-18.mkv'     backup                                     slack-4.21.1-0.1.fc21.x86_64.rpm
'2022-10-20 12-53-59.mkv'     Desktop                                    snap
'2022-10-30 14-13-24.mkv'    'Docker & Kubernetes The Practical Guide'   tanish.mp3
'2022-10-30 14-47-37.mkv'     docker-start.mkv                           Templates
'2022-11-27 10-49-45.mkv'     Documents                                  tensorflow_datasets
'2022-12-25 18-09-52.mkv'     Downloads                                  test.c
'2023-01-01 11-47-16.mkv'     Figure_1.png                               test.jl
'2023-01-06 10-18-08.mkv'     flutter                                    test.py
'2023-01-07 09-58-54.mkv'     forms.java                                'The Complete 2021 Flutter Development Bootcamp with Dart'
'2023-01-07 11-09-13.mkv'     genymotion                                 University
'2023-01-07 22-08-34.mkv'     libinput-gestures                          Videos
'2023-01-07 22-27-24.mkv'     microsoftteams-linux-x64                   VNC-Server-6.10.1-Linux-x64.rpm
'2023-01-08 11-44-31.mkv'     Music                                      welcome0.mp3
'2023-01-12 10-14-17.mkv'     mysqlsession1                              welcome1.mp3
 accuracy-fashion-mnist.png   nltk_data                                  welcome3.mp3
 add-port.txt                 oh-my-fish                                 welcome4.mp3
 anaconda3                    Pictures                                   welcome6.mp3
 anaconda.sh                  Playground                                 welcome7.mp3
 Android                      Public                                     welcome8.mp3
 anirveda-automation.txt      R                                          welcome9.mp3
 antigen.zsh                  sapi.py                                    welcome.mp3
 a.out                        scikit_learn_data
```

3.1) ls -R command:

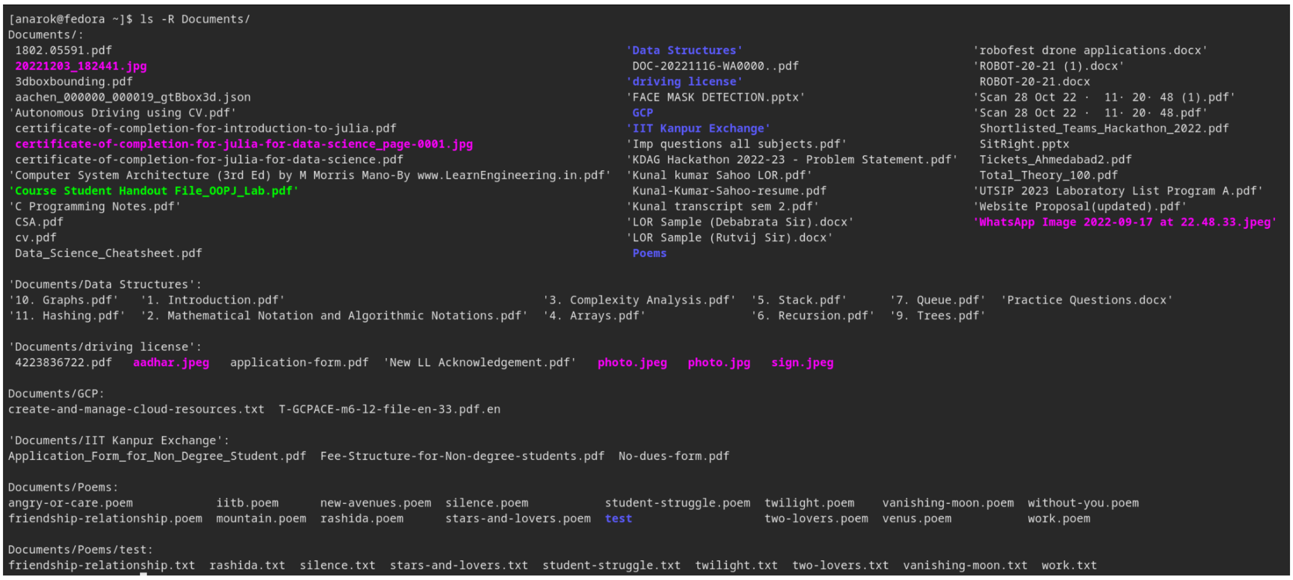
The ls command lists the contents of your current working directory and recurring files.



```
[anarok@fedora ~]$ ls -a
 .                                .cache                                         .local                                  test.c
 ..                               .cgnstools                                     .m2                                     test.jl
'2022-10-07 19-30-10.mkv'         .cmake                                         microsoftteams-linux-x64                test.py
'2022-10-19 19-37-18.mkv'         .conda                                         .mozilla                               'The Complete 2021 Flutter Development Bootcamp with Dart'
'2022-10-20 12-53-59.mkv'         .condarc                                       Music                                   University
'2022-10-30 14-13-24.mkv'         .config                                        .mysql                                  .var
'2022-10-30 14-47-37.mkv'         .dart                                          .mysql_history                          Videos
'2022-11-27 10-49-45.mkv'         .dartServer                                    mysqlsession1                           .viminfo
'2022-12-25 18-09-52.mkv'         Desktop                                        nltk_data                               .vim_mru_files
'2023-01-01 11-47-16.mkv'         .docker                                        .node_repl_history                      .vimrc
'2023-01-06 10-18-08.mkv'        'Docker & Kubernetes The Practical Guide'       .npm                                    .vim_runtime
'2023-01-07 09-58-54.mkv'         docker-start.mkv                               oh-my-fish                              .vnc
'2023-01-07 11-09-13.mkv'         Documents                                      .oh-my-zsh                              VNC-Server-6.10.1-Linux-x64.rpm
'2023-01-07 22-08-34.mkv'         Downloads                                      .openshot_qt                            .vscode
'2023-01-07 22-27-24.mkv'         .eclipse                                       .p2                                     .vscode-R
'2023-01-08 11-44-31.mkv'         .emulator_console_auth_token                   .password-store                         welcome0.mp3
'2023-01-12 10-14-17.mkv'         Figure_1.png                                   Pictures                                welcome1.mp3
 accuracy-fashion-mnist.png       .fltk                                          .pki                                    welcome3.mp3
 add-port.txt                     .flutter                                       Playground                              welcome4.mp3
 anaconda3                        flutter                                        .pub-cache                              welcome6.mp3
 anaconda.sh                      .flutter-devtools                              Public                                  welcome7.mp3
 .android                         forms.java                                     .pulsar                                 welcome8.mp3
 Android                          .gazebo                                        .python_history                         welcome9.mp3
 anirveda-automation.txt          .Genymobile                                    .python_history-19919.tmp               welcome.mp3
 .antigen                         genymotion                                     R                                       .wget-hsts
 .antigenrc                       .gitconfig                                     .RData                                  .wine
 .antigenrc.zwc                   .gnome                                         .Rhistory                               .zcompdump
 antigen.zsh                      .gradle                                        .ros                                    .zcompdump-fedora-5.8.1
 .anydesk                         .ignition                                      sapi.py                                 .zcompdump-fedora-5.8.1.zwc
 a.out                            .ipython                                       scikit_learn_data                       .zcompdump-fedora-5.9
 Arduino                          .java                                          script.txt                              .zcompdump-fedora-5.9.zwc
 .arduino15                       .jlassetregistry.json                          .shell.pre-oh-my-zsh                    .zcompdump-MiWiFi-R4CM-srv-5.8.1
 .arduinoIDE                      .julia                                         slack-4.21.1-0.1.fc21.x86_64.rpm        .zcompdump-MiWiFi-R4CM-srv-5.8.1.zwc
 .astropy                         .jupyter                                       snap                                    .zcompdump-MiWiFi-R4CM-srv-5.9
 backup                           .kaggle                                        .ssh                                    .zcompdump-MiWiFi-R4CM-srv-5.9.zwc
 .bash_history                    .keras                                         .swt                                    .zsh_history
 .bash_logout                     .kivy                                          tanish.mp3                              .zshrc
 .bash_profile                    .lesshst                                       Templates                               .zshrc.pre-oh-my-zsh
 .bashrc                          libinput-gestures                              tensorflow_datasets                     .zshrc.zwc
```

3.2) ls -a command:

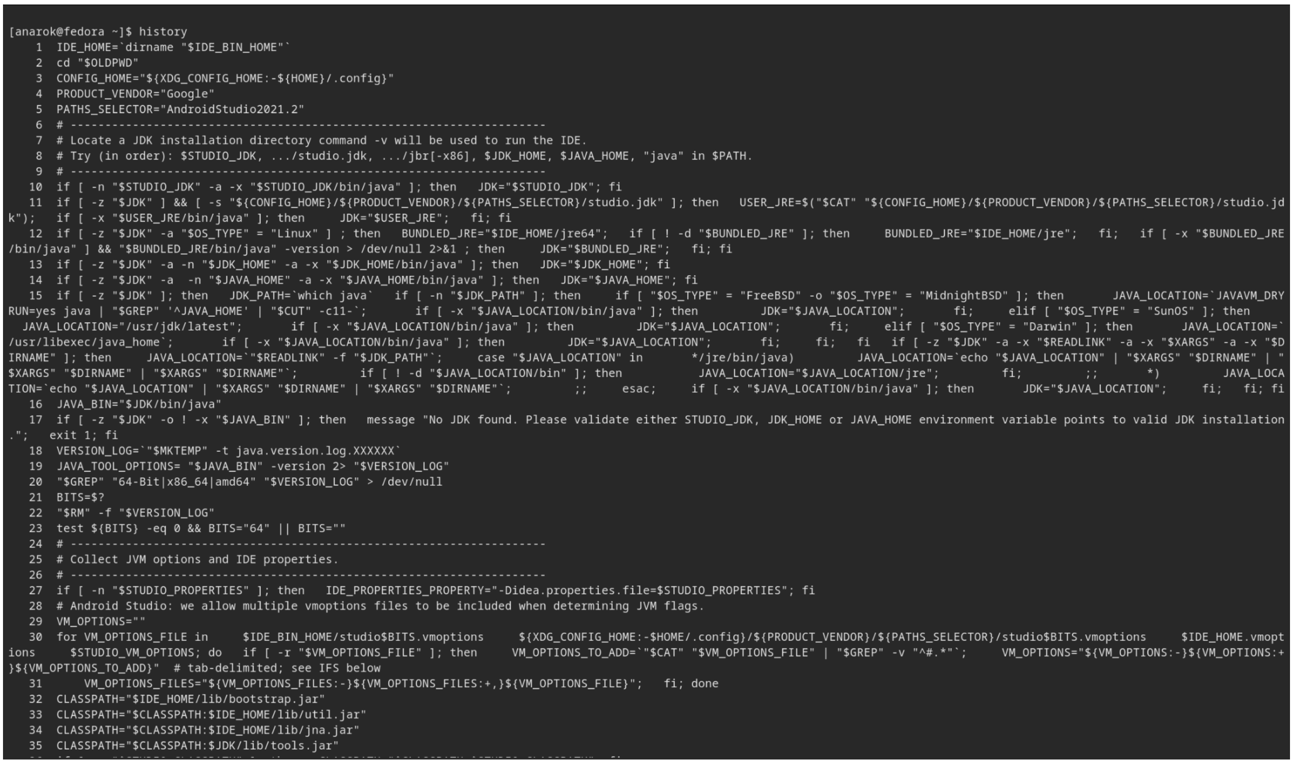
The ls command is used to list hidden directories and files.



```
[anarok@fedora ~]$ ls -R Documents/
Documents/:
 1802.05591.pdf                                                                        'Data Structures'                                    'robofest drone applications.docx'
 20221203_182441.jpg                                                                    DOC-20221116-WA0000..pdf                            'ROBOT-20-21 (1).docx'
 3dboxbounding.pdf                                                                     'driving license'                                     ROBOT-20-21.docx
 aachen_000000_000019_gtBbox3d.json                                                    'FACE MASK DETECTION.pptx'                           'Scan 28 Oct 22 ·  11· 20· 48 (1).pdf'
'Autonomous Driving using CV.pdf'                                                       GCP                                                 'Scan 28 Oct 22 ·  11· 20· 48.pdf'
 certificate-of-completion-for-introduction-to-julia.pdf                               'IIT Kanpur Exchange'                                 Shortlisted_Teams_Hackathon_2022.pdf
 certificate-of-completion-for-julia-for-data-science_page-0001.jpg                    'Imp questions all subjects.pdf'                      SitRight.pptx
 certificate-of-completion-for-julia-for-data-science.pdf                              'KDAG Hackathon 2022-23 - Problem Statement.pdf'      Tickets_Ahmedabad2.pdf
'Computer System Architecture (3rd Ed) by M Morris Mano-By www.LearnEngineering.in.pdf'  'Kunal kumar Sahoo LOR.pdf'                          Total_Theory_100.pdf
'Course Student Handout File_OOPJ_Lab.pdf'                                              Kunal-Kumar-Sahoo-resume.pdf                        'UTSIP 2023 Laboratory List Program A.pdf'
'C Programming Notes.pdf'                                                              'Kunal transcript sem 2.pdf'                         'Website Proposal(updated).pdf'
 CSA.pdf                                                                               'LOR Sample (Debabrata Sir).docx'                    'WhatsApp Image 2022-09-17 at 22.48.33.jpeg'
 cv.pdf                                                                                'LOR Sample (Rutvij Sir).docx'
 Data_Science_Cheatsheet.pdf                                                            Poems

'Documents/Data Structures':
'10. Graphs.pdf'   '1. Introduction.pdf'                                '3. Complexity Analysis.pdf'  '5. Stack.pdf'      '7. Queue.pdf'  'Practice Questions.docx'
'11. Hashing.pdf'  '2. Mathematical Notation and Algorithmic Notations.pdf'  '4. Arrays.pdf'               '6. Recursion.pdf'  '9. Trees.pdf'

'Documents/driving license':
 4223836722.pdf   aadhar.jpeg   application-form.pdf  'New LL Acknowledgement.pdf'   photo.jpeg   photo.jpg   sign.jpeg

Documents/GCP:
create-and-manage-cloud-resources.txt  T-GCPACE-m6-l2-file-en-33.pdf.en

'Documents/IIT Kanpur Exchange':
Application_Form_for_Non_Degree_Student.pdf  Fee-Structure-for-Non-degree-students.pdf  No-dues-form.pdf

Documents/Poems:
angry-or-care.poem            iitb.poem      new-avenues.poem  silence.poem           student-struggle.poem  twilight.poem    vanishing-moon.poem  without-you.poem
friendship-relationship.poem  mountain.poem  rashida.poem      stars-and-lovers.poem  test                   two-lovers.poem  venus.poem           work.poem

Documents/Poems/test:
friendship-relationship.txt  rashida.txt  silence.txt  stars-and-lovers.txt  student-struggle.txt  twilight.txt  two-lovers.txt  vanishing-moon.txt  work.txt
```

4) history command:

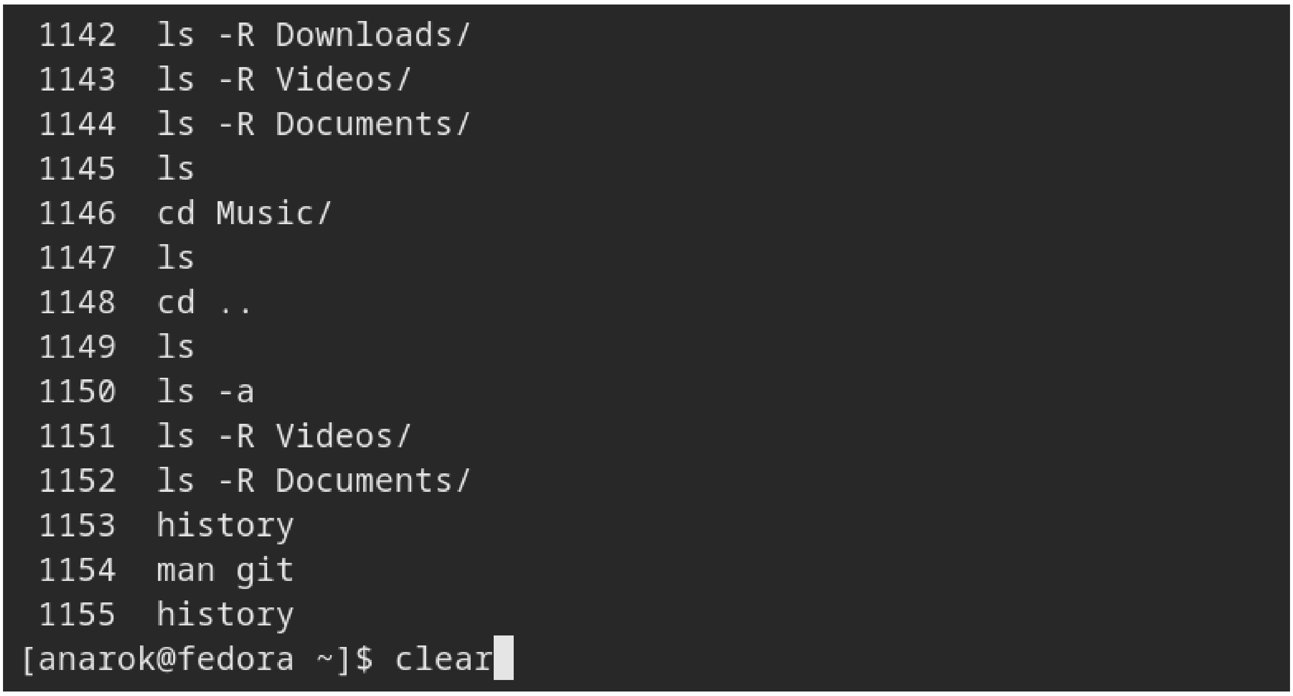
history command is used to view the previously executed command.



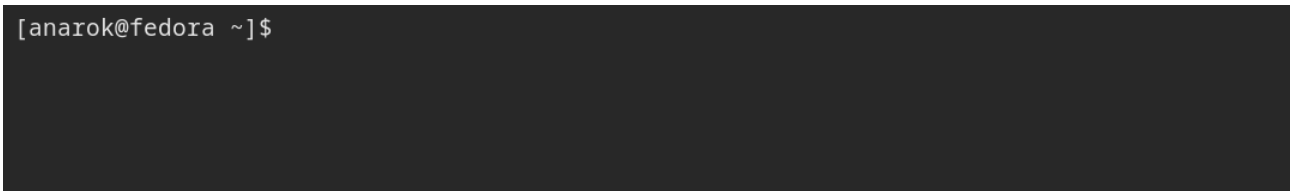
```
[anarok@fedora ~]$ history
    1  IDE_HOME=`dirname "$IDE_BIN_HOME"`
    2  cd "$OLDPWD"
    3  CONFIG_HOME="${XDG_CONFIG_HOME:-${HOME}/.config}"
    4  PRODUCT_VENDOR="Google"
    5  PATHS_SELECTOR="AndroidStudio2021.2"
    6  # ---------------------------------------------------------------------
    7  # Locate a JDK installation directory command -v will be used to run the IDE.
    8  # Try (in order): $STUDIO_JDK, .../studio.jdk, .../jbr[-x86], $JDK_HOME, $JAVA_HOME, "java" in $PATH.
    9  # ---------------------------------------------------------------------
   10  if [ -n "$STUDIO_JDK" -a -x "$STUDIO_JDK/bin/java" ]; then   JDK="$STUDIO_JDK"; fi
   11  if [ -z "$JDK" ] && [ -s "${CONFIG_HOME}/${PRODUCT_VENDOR}/${PATHS_SELECTOR}/studio.jdk" ]; then   USER_JRE=$("$CAT" "${CONFIG_HOME}/${PRODUCT_VENDOR}/${PATHS_SELECTOR}/studio.jd
k");   if [ -x "$USER_JRE/bin/java" ]; then     JDK="$USER_JRE";   fi; fi
   12  if [ -z "$JDK" -a "$OS_TYPE" = "Linux" ] ; then   BUNDLED_JRE="$IDE_HOME/jre64";   if [ ! -d "$BUNDLED_JRE" ]; then     BUNDLED_JRE="$IDE_HOME/jre";   fi;   if [ -x "$BUNDLED_JRE
/bin/java" ] && "$BUNDLED_JRE/bin/java" -version > /dev/null 2>&1 ; then     JDK="$BUNDLED_JRE";   fi; fi
   13  if [ -z "$JDK" -a -n "$JDK_HOME" -a -x "$JDK_HOME/bin/java" ]; then   JDK="$JDK_HOME"; fi
   14  if [ -z "$JDK" -a  -n "$JAVA_HOME" -a -x "$JAVA_HOME/bin/java" ]; then   JDK="$JAVA_HOME"; fi
   15  if [ -z "$JDK" ]; then   JDK_PATH=`which java`   if [ -n "$JDK_PATH" ]; then     if [ "$OS_TYPE" = "FreeBSD" -o "$OS_TYPE" = "MidnightBSD" ]; then       JAVA_LOCATION=`JAVAVM_DRY
RUN=yes java | "$GREP" '^JAVA_HOME' | "$CUT" -c11-`;       if [ -x "$JAVA_LOCATION/bin/java" ]; then         JDK="$JAVA_LOCATION";       fi;     elif [ "$OS_TYPE" = "SunOS" ]; then
  JAVA_LOCATION="/usr/jdk/latest";       if [ -x "$JAVA_LOCATION/bin/java" ]; then         JDK="$JAVA_LOCATION";       fi;     elif [ "$OS_TYPE" = "Darwin" ]; then       JAVA_LOCATION=`
/usr/libexec/java_home`;       if [ -x "$JAVA_LOCATION/bin/java" ]; then         JDK="$JAVA_LOCATION";       fi;     fi;   fi   if [ -z "$JDK" -a -x "$READLINK" -a -x "$XARGS" -a -x "$D
IRNAME" ]; then     JAVA_LOCATION=`"$READLINK" -f "$JDK_PATH"`;     case "$JAVA_LOCATION" in       */jre/bin/java)         JAVA_LOCATION=`echo "$JAVA_LOCATION" | "$XARGS" "$DIRNAME" | "
$XARGS" "$DIRNAME" | "$XARGS" "$DIRNAME"`;         if [ ! -d "$JAVA_LOCATION/bin" ]; then           JAVA_LOCATION="$JAVA_LOCATION/jre";         fi;         ;;       *)         JAVA_LOCA
TION=`echo "$JAVA_LOCATION" | "$XARGS" "$DIRNAME" | "$XARGS" "$DIRNAME"`;         ;;     esac;     if [ -x "$JAVA_LOCATION/bin/java" ]; then       JDK="$JAVA_LOCATION";     fi;   fi; fi
   16  JAVA_BIN="$JDK/bin/java"
   17  if [ -z "$JDK" -o ! -x "$JAVA_BIN" ]; then   message "No JDK found. Please validate either STUDIO_JDK, JDK_HOME or JAVA_HOME environment variable points to valid JDK installation
.";   exit 1; fi
   18  VERSION_LOG=`"$MKTEMP" -t java.version.log.XXXXXX`
   19  JAVA_TOOL_OPTIONS= "$JAVA_BIN" -version 2> "$VERSION_LOG"
   20  "$GREP" "64-Bit|x86_64|amd64" "$VERSION_LOG" > /dev/null
   21  BITS=$?
   22  "$RM" -f "$VERSION_LOG"
   23  test ${BITS} -eq 0 && BITS="64" || BITS=""
   24  # ---------------------------------------------------------------------
   25  # Collect JVM options and IDE properties.
   26  # ---------------------------------------------------------------------
   27  if [ -n "$STUDIO_PROPERTIES" ]; then   IDE_PROPERTIES_PROPERTY="-Didea.properties.file=$STUDIO_PROPERTIES"; fi
   28  # Android Studio: we allow multiple vmoptions files to be included when determining JVM flags.
   29  VM_OPTIONS=""
   30  for VM_OPTIONS_FILE in     $IDE_BIN_HOME/studio$BITS.vmoptions     ${XDG_CONFIG_HOME:-$HOME/.config}/${PRODUCT_VENDOR}/${PATHS_SELECTOR}/studio$BITS.vmoptions     $IDE_HOME.vmopt
ions     $STUDIO_VM_OPTIONS; do   if [ -r "$VM_OPTIONS_FILE" ]; then     VM_OPTIONS_TO_ADD=`"$CAT" "$VM_OPTIONS_FILE" | "$GREP" -v "^#.*"`;     VM_OPTIONS="${VM_OPTIONS:-}${VM_OPTIONS:+
}${VM_OPTIONS_TO_ADD}"  # tab-delimited; see IFS below
   31      VM_OPTIONS_FILES="${VM_OPTIONS_FILES:-}${VM_OPTIONS_FILES:+,}${VM_OPTIONS_FILE}";   fi; done
   32  CLASSPATH="$IDE_HOME/lib/bootstrap.jar"
   33  CLASSPATH="$CLASSPATH:$IDE_HOME/lib/util.jar"
   34  CLASSPATH="$CLASSPATH:$IDE_HOME/lib/jna.jar"
   35  CLASSPATH="$CLASSPATH:$JDK/lib/tools.jar"
```

5) clear command:

The clear command **clears your screen, if possible.** The clear command first checks the TERM environment variable for the terminal type



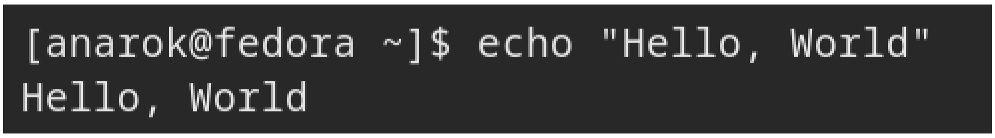
```
 1142  ls -R Downloads/
 1143  ls -R Videos/
 1144  ls -R Documents/
 1145  ls
 1146  cd Music/
 1147  ls
 1148  cd ..
 1149  ls
 1150  ls -a
 1151  ls -R Videos/
 1152  ls -R Documents/
 1153  history
 1154  man git
 1155  history
[anarok@fedora ~]$ clear
```



```
[anarok@fedora ~]$
```

6) echo command:

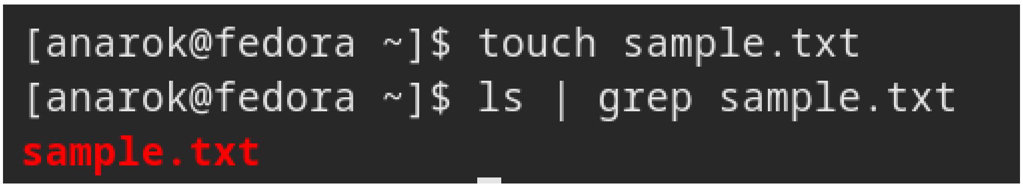
In computing, echo is a command that outputs the strings that are passed to it as arguments.



```
[anarok@fedora ~]$ echo "Hello, World"
Hello, World
```

7) touch command:

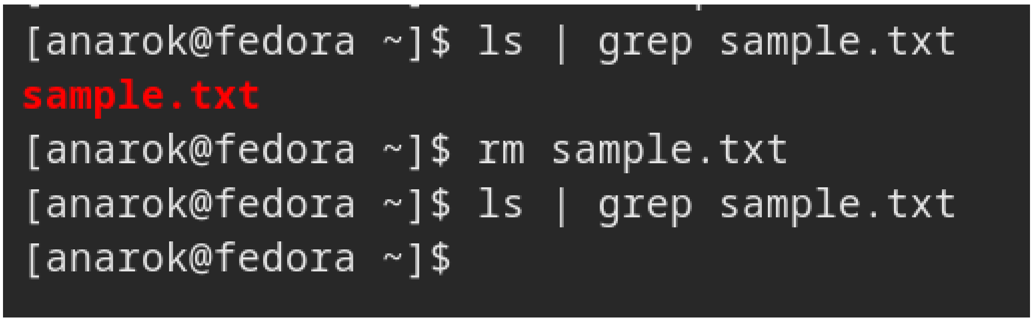
The touch command updates the access and modification times of each file specified by the File parameter of each directory specified by the Directory parameter.



```
[anarok@fedora ~]$ touch sample.txt
[anarok@fedora ~]$ ls | grep sample.txt
sample.txt
```

8) rm command:

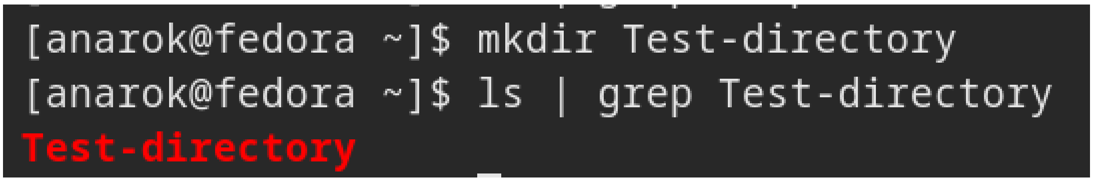
The rm command is **used to delete files.** rm -i will ask before deleting each file.



```
[anarok@fedora ~]$ ls | grep sample.txt
sample.txt
[anarok@fedora ~]$ rm sample.txt
[anarok@fedora ~]$ ls | grep sample.txt
[anarok@fedora ~]$
```

9) mkdir command:

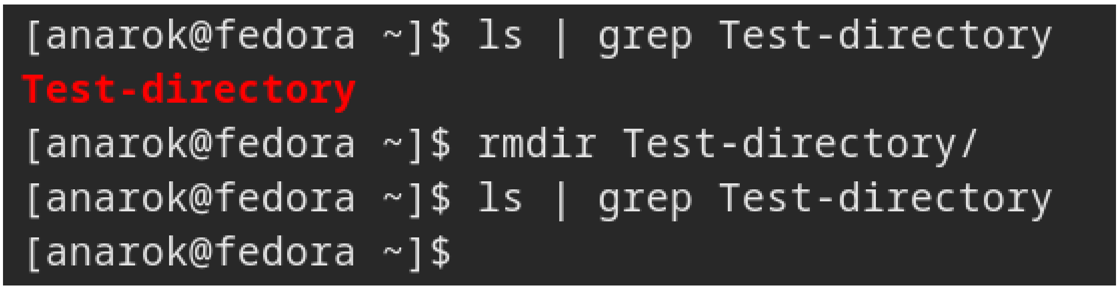
The mkdir command in Linux/Unix allows users to create or make new directories. mkdir stands for “make directory.”



```
[anarok@fedora ~]$ mkdir Test-directory
[anarok@fedora ~]$ ls | grep Test-directory
Test-directory
```

10) rmdir command:

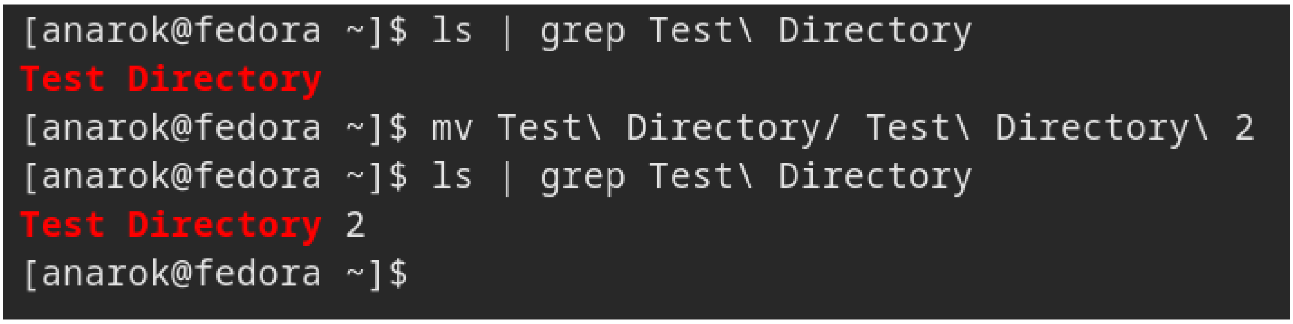
The rmdir command **removes the directory, specified by the Directory parameter, from the system.** The directory must be empty before you can remove it, and you must have write permission in its parent directory.



```
[anarok@fedora ~]$ ls | grep Test-directory
Test-directory
[anarok@fedora ~]$ rmdir Test-directory/
[anarok@fedora ~]$ ls | grep Test-directory
[anarok@fedora ~]$
```

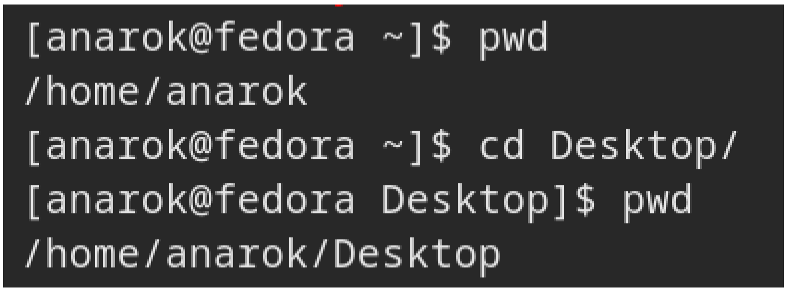
11) mv command:

The mv command moves files and directories from one directory to another or renames a file or directory. If you move a file or directory to a new directory, it retains the base file name.



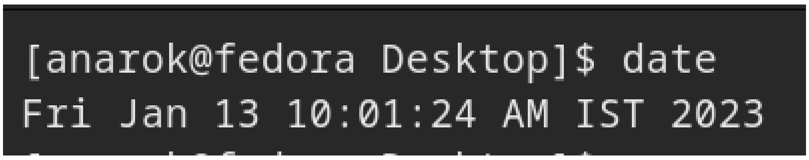
```
[anarok@fedora ~]$ ls | grep Test\ Directory
Test Directory
[anarok@fedora ~]$ mv Test\ Directory/ Test\ Directory\ 2
[anarok@fedora ~]$ ls | grep Test\ Directory
Test Directory 2
[anarok@fedora ~]$
```

12) cd command:
The cd command, also known as chdir (change directory), is a command-line shell command used to change the current working directory in various operating systems. It can be used in shell scripts and batch files.



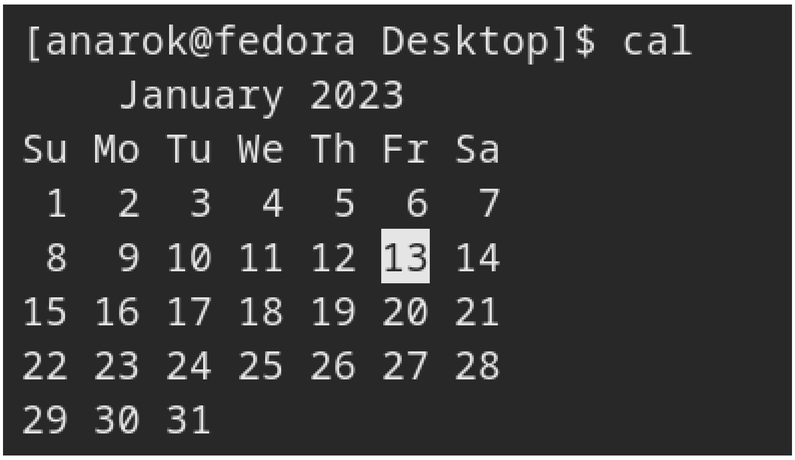
```
[anarok@fedora ~]$ pwd
/home/anarok
[anarok@fedora ~]$ cd Desktop/
[anarok@fedora Desktop]$ pwd
/home/anarok/Desktop
```

13) date command:
The date command is used to print the current day, date and time set in the machine.



```
[anarok@fedora Desktop]$ date
Fri Jan 13 10:01:24 AM IST 2023
```

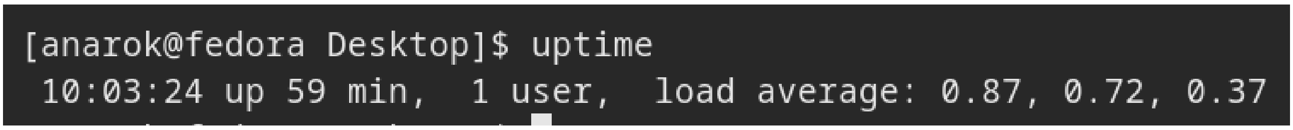
14) cal command:
The cal command is used to print the calendar of the current month set in the machine.



```
[anarok@fedora Desktop]$ cal
    January 2023
Su Mo Tu We Th Fr Sa
 1  2  3  4  5  6  7
 8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31
```

15) uptime command:

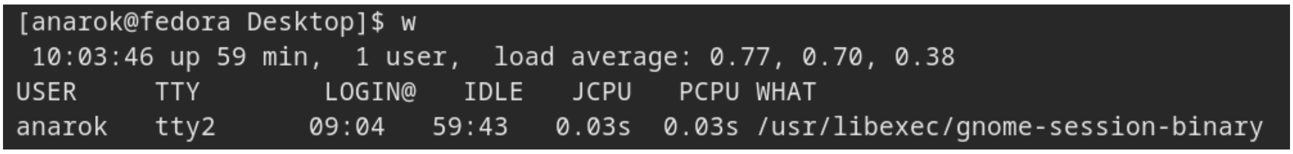
The uptime command displays how long the computer has been booted and is in operation.



```
[anarok@fedora Desktop]$ uptime
 10:03:24 up 59 min,  1 user,  load average: 0.87, 0.72, 0.37
```

16) w command:

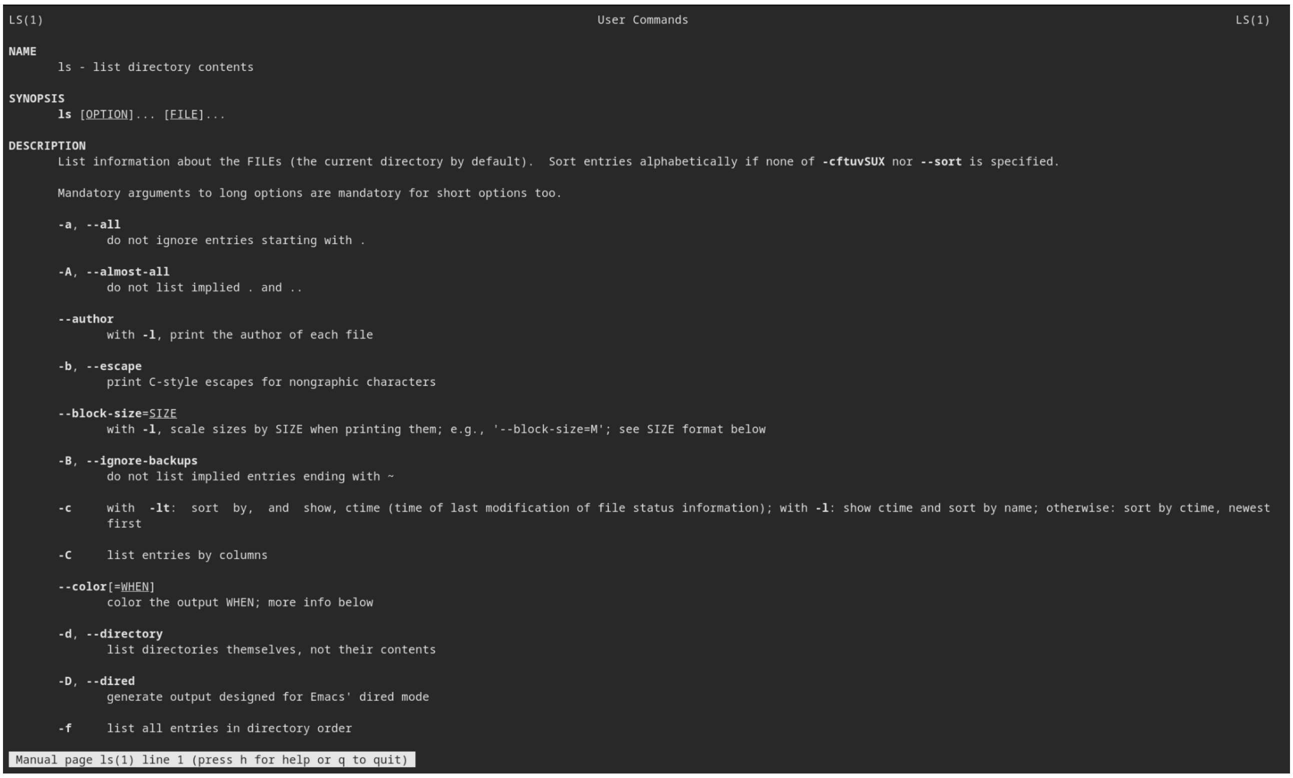
The w command displays how many users are online.



```
[anarok@fedora Desktop]$ w
 10:03:46 up 59 min,  1 user,  load average: 0.77, 0.70, 0.38
USER     TTY        LOGIN@   IDLE   JCPU   PCPU WHAT
anarok   tty2      09:04   59:43   0.03s  0.03s /usr/libexec/gnome-session-binary
```

17) man command:

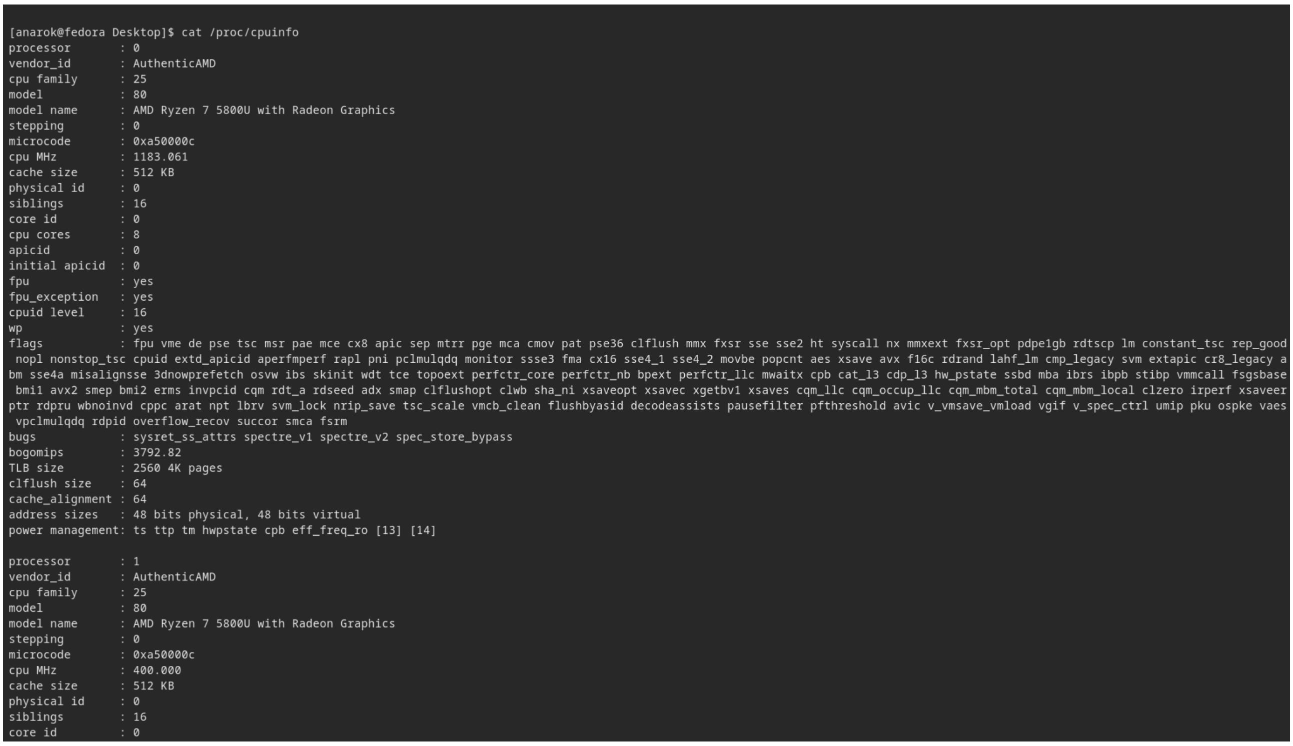
Displays documentation or manual of the command passed as the argument.



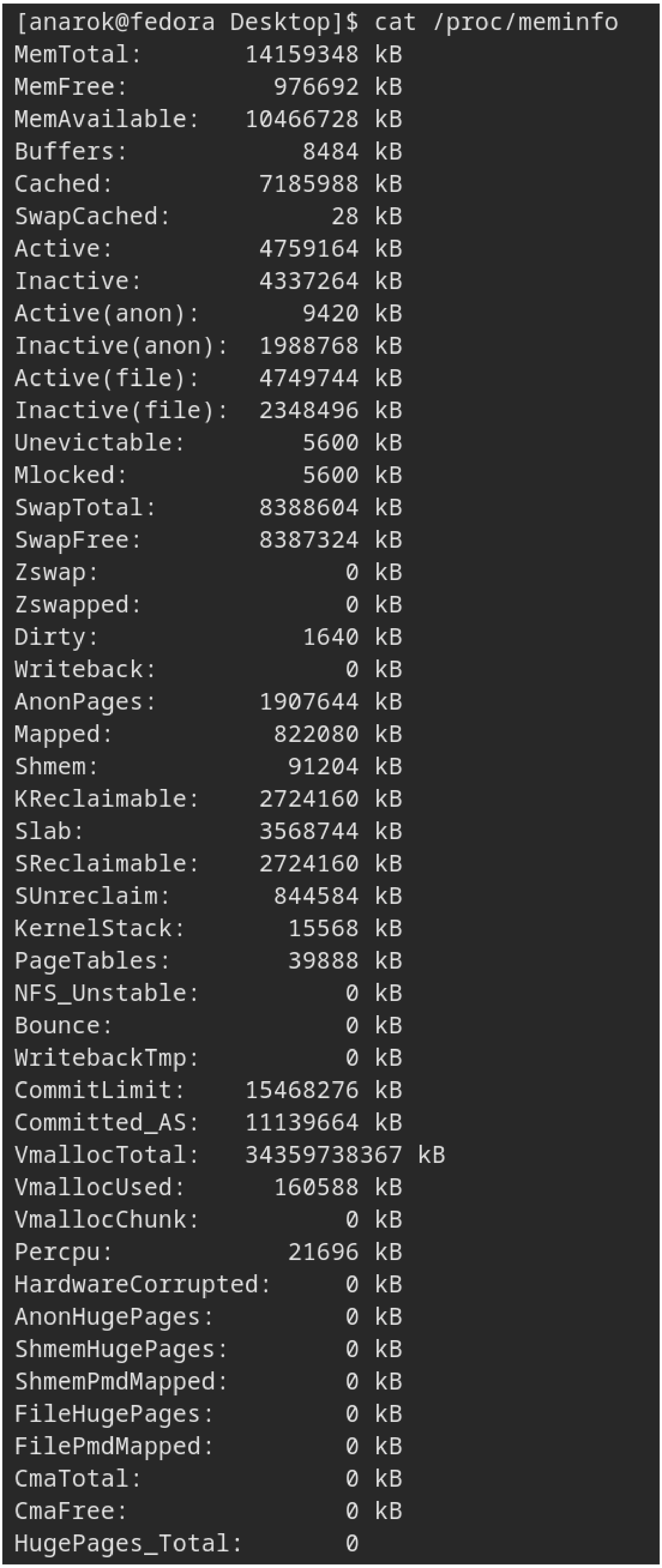
```
LS(1)                                   User Commands                                   LS(1)

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

       -b, --escape
              print C-style escapes for nongraphic characters

       --block-size=SIZE
              with -l, scale sizes by SIZE when printing them; e.g., '--block-size=M'; see SIZE format below

       -B, --ignore-backups
              do not list implied entries ending with ~

       -c     with  -lt:  sort  by,  and  show, ctime (time of last modification of file status information); with -l: show ctime and sort by name; otherwise: sort by ctime, newest
              first

       -C     list entries by columns

       --color[=WHEN]
              color the output WHEN; more info below

       -d, --directory
              list directories themselves, not their contents

       -D, --dired
              generate output designed for Emacs' dired mode

       -f     list all entries in directory order

 Manual page ls(1) line 1 (press h for help or q to quit)
```

18) cat command:

Displays information passed in the argument in the console in concatenated form.



```
[anarok@fedora Desktop]$ cat /proc/cpuinfo
processor       : 0
vendor_id       : AuthenticAMD
cpu family      : 25
model           : 80
model name      : AMD Ryzen 7 5800U with Radeon Graphics
stepping        : 0
microcode       : 0xa50000c
cpu MHz         : 1183.061
cache size      : 512 KB
physical id     : 0
siblings        : 16
core id         : 0
cpu cores       : 8
apicid          : 0
initial apicid  : 0
fpu             : yes
fpu_exception   : yes
cpuid level     : 16
wp              : yes
flags           : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 ht syscall nx mmxext fxsr_opt pdpe1gb rdtscp lm constant_tsc rep_good
 nopl nonstop_tsc cpuid extd_apicid aperfmperf rapl pni pclmulqdq monitor ssse3 fma cx16 sse4_1 sse4_2 movbe popcnt aes xsave avx f16c rdrand lahf_lm cmp_legacy svm extapic cr8_legacy a
bm sse4a misalignsse 3dnowprefetch osvw ibs skinit wdt tce topoext perfctr_core perfctr_nb bpext perfctr_llc mwaitx cpb cat_l3 cdp_l3 hw_pstate ssbd mba ibrs ibpb stibp vmmcall fsgsbase
 bmi1 avx2 smep bmi2 erms invpcid cqm rdt_a rdseed adx smap clflushopt clwb sha_ni xsaveopt xsavec xgetbv1 xsaves cqm_llc cqm_occup_llc cqm_mbm_total cqm_mbm_local clzero irperf xsaveer
ptr rdpru wbnoinvd cppc arat npt lbrv svm_lock nrip_save tsc_scale vmcb_clean flushbyasid decodeassists pausefilter pfthreshold avic v_vmsave_vmload vgif v_spec_ctrl umip pku ospke vaes
 vpclmulqdq rdpid overflow_recov succor smca fsrm
bugs            : sysret_ss_attrs spectre_v1 spectre_v2 spec_store_bypass
bogomips        : 3792.82
TLB size        : 2560 4K pages
clflush size    : 64
cache_alignment : 64
address sizes   : 48 bits physical, 48 bits virtual
power management: ts ttp tm hwpstate cpb eff_freq_ro [13] [14]

processor       : 1
vendor_id       : AuthenticAMD
cpu family      : 25
model           : 80
model name      : AMD Ryzen 7 5800U with Radeon Graphics
stepping        : 0
microcode       : 0xa50000c
cpu MHz         : 400.000
cache size      : 512 KB
physical id     : 0
siblings        : 16
core id         : 0
```

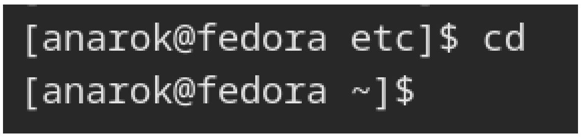


```
[anarok@fedora Desktop]$ cat /proc/meminfo
MemTotal:       14159348 kB
MemFree:          976692 kB
MemAvailable:   10466728 kB
Buffers:            8484 kB
Cached:          7185988 kB
SwapCached:           28 kB
Active:          4759164 kB
Inactive:        4337264 kB
Active(anon):       9420 kB
Inactive(anon):  1988768 kB
Active(file):    4749744 kB
Inactive(file):  2348496 kB
Unevictable:        5600 kB
Mlocked:            5600 kB
SwapTotal:       8388604 kB
SwapFree:        8387324 kB
Zswap:                 0 kB
Zswapped:              0 kB
Dirty:              1640 kB
Writeback:             0 kB
AnonPages:       1907644 kB
Mapped:           822080 kB
Shmem:             91204 kB
KReclaimable:    2724160 kB
Slab:            3568744 kB
SReclaimable:    2724160 kB
SUnreclaim:       844584 kB
KernelStack:       15568 kB
PageTables:        39888 kB
NFS_Unstable:          0 kB
Bounce:                0 kB
WritebackTmp:          0 kB
CommitLimit:    15468276 kB
Committed_AS:   11139664 kB
VmallocTotal:   34359738367 kB
VmallocUsed:      160588 kB
VmallocChunk:          0 kB
Percpu:            21696 kB
HardwareCorrupted:     0 kB
AnonHugePages:         0 kB
ShmemHugePages:        0 kB
ShmemPmdMapped:        0 kB
FileHugePages:         0 kB
FilePmdMapped:         0 kB
CmaTotal:              0 kB
CmaFree:               0 kB
HugePages_Total:       0
```

Experiment-2: Basic Terminal Commands-2

1) **cd**

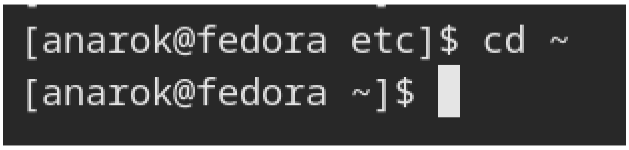
   cd stands for change directory



```
[anarok@fedora etc]$ cd
[anarok@fedora ~]$
```

2) **cd ~**

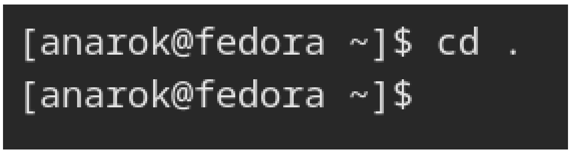
   ~ stands for 'home'. So, this command will change the current directory to home.



```
[anarok@fedora etc]$ cd ~
[anarok@fedora ~]$ █
```

3) **cd .**

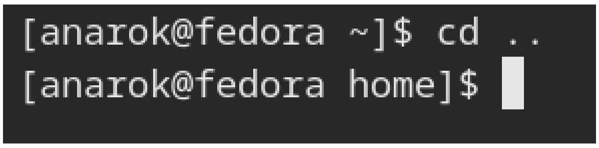
   . stands for current directory. So this command will redirect to the same directory itself.



```
[anarok@fedora ~]$ cd .
[anarok@fedora ~]$
```

4) **cd ..**

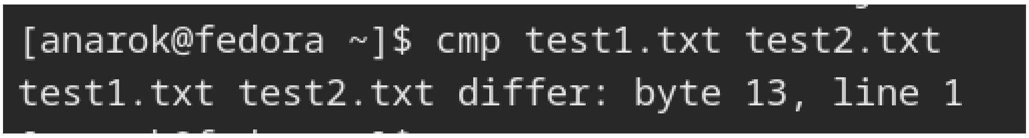
   .. stands for parent working directory. So this command will change the current working directory to the parent directory.



```
[anarok@fedora ~]$ cd ..
[anarok@fedora home]$ █
```

5) **cmp**

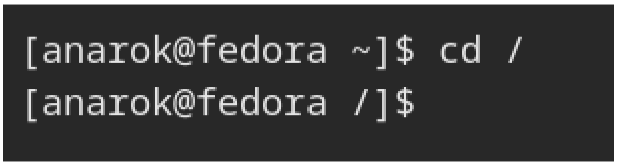
   cmp command is used to compare the two files byte by byte and checks whether two files are identical or not.



```
[anarok@fedora ~]$ cmp test1.txt test2.txt
test1.txt test2.txt differ: byte 13, line 1
```

6) **cd /**

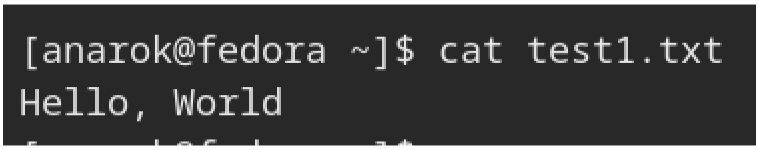
/ redirects to the root directory of the operating system.



```
[anarok@fedora ~]$ cd /
[anarok@fedora /]$
```

7) **cat**

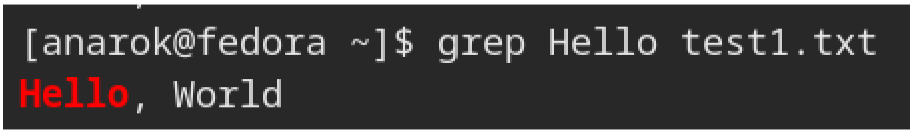
cat command reads the contents of the file. It helps to create, view and concatenate files.



```
[anarok@fedora ~]$ cat test1.txt
Hello, World
```

8) **grep**

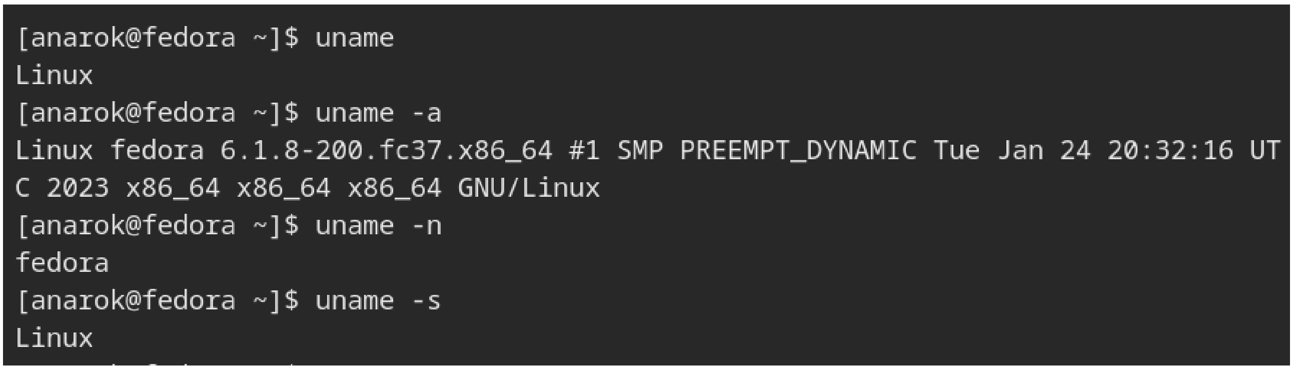
grep command used to search for a string of characters in a specified file or text. The text pattern search is regex. When it finds a match, it prints the line with the result.



```
[anarok@fedora ~]$ grep Hello test1.txt
Hello, World
```

9) **uname, uname -a, uname -n, uname -s**

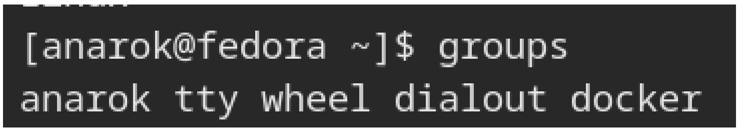
uname -a prints all the system information in the following order: Kernel name, network node hostname, kernel release date, kernel version, machine hardware name, hardware platform, operating system. Uname -s prints the kernel name. uname -n prints the hostname of the network node(current computer).



```
[anarok@fedora ~]$ uname
Linux
[anarok@fedora ~]$ uname -a
Linux fedora 6.1.8-200.fc37.x86_64 #1 SMP PREEMPT_DYNAMIC Tue Jan 24 20:32:16 UT
C 2023 x86_64 x86_64 x86_64 GNU/Linux
[anarok@fedora ~]$ uname -n
fedora
[anarok@fedora ~]$ uname -s
Linux
```

10) groups

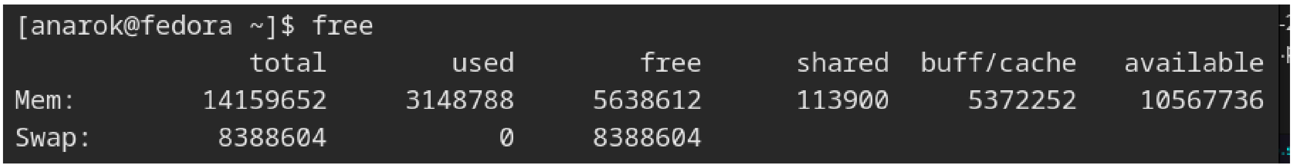
Groups command displays all the names of the group a user is part of.



```
[anarok@fedora ~]$ groups
anarok tty wheel dialout docker
```

11) free

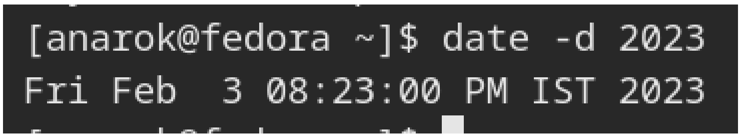
free command outputs a summary of RAM usage, including total, used, free, shared, and available memory and swap space.



```
[anarok@fedora ~]$ free
               total        used        free      shared  buff/cache   available
Mem:        14159652     3148788     5638612      113900     5372252    10567736
Swap:        8388604           0     8388604
```

12) date -d

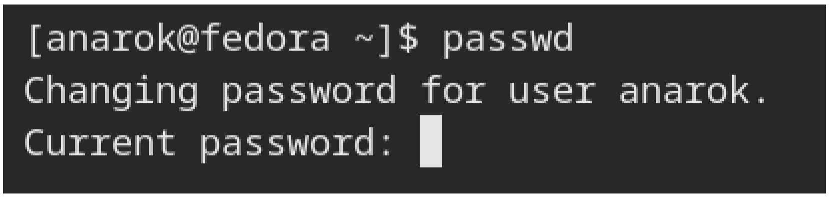
This option allows users to operate on a specific date.



```
[anarok@fedora ~]$ date -d 2023
Fri Feb  3 08:23:00 PM IST 2023
```

13) passwd

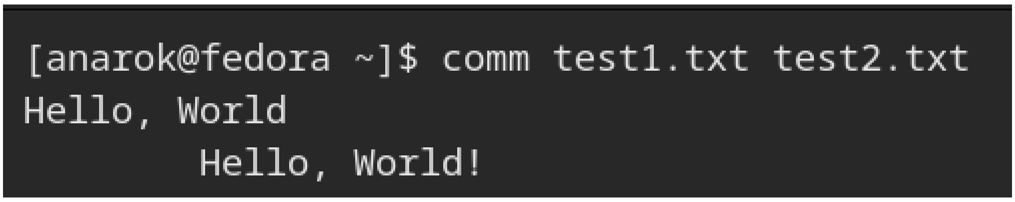
passwd command is used to change the user account passwords.



```
[anarok@fedora ~]$ passwd
Changing password for user anarok.
Current password: 
```

14) comm

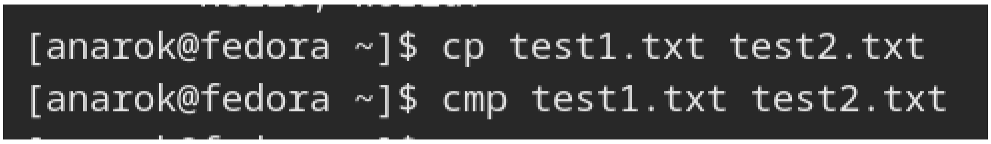
The comm command compares two files or streams.



```
[anarok@fedora ~]$ comm test1.txt test2.txt
Hello, World
        Hello, World!
```

15) cp

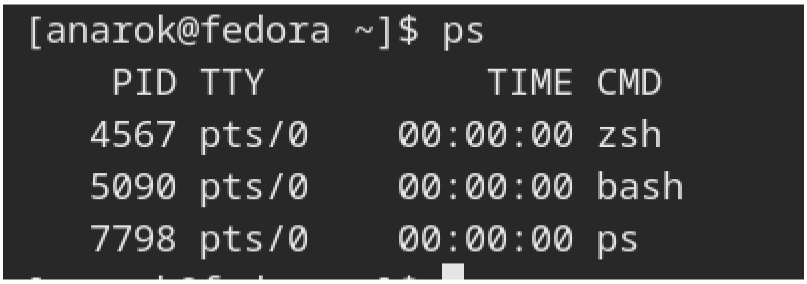
cp command copies one file’s content to another file.



```
[anarok@fedora ~]$ cp test1.txt test2.txt
[anarok@fedora ~]$ cmp test1.txt test2.txt
```

16) ps

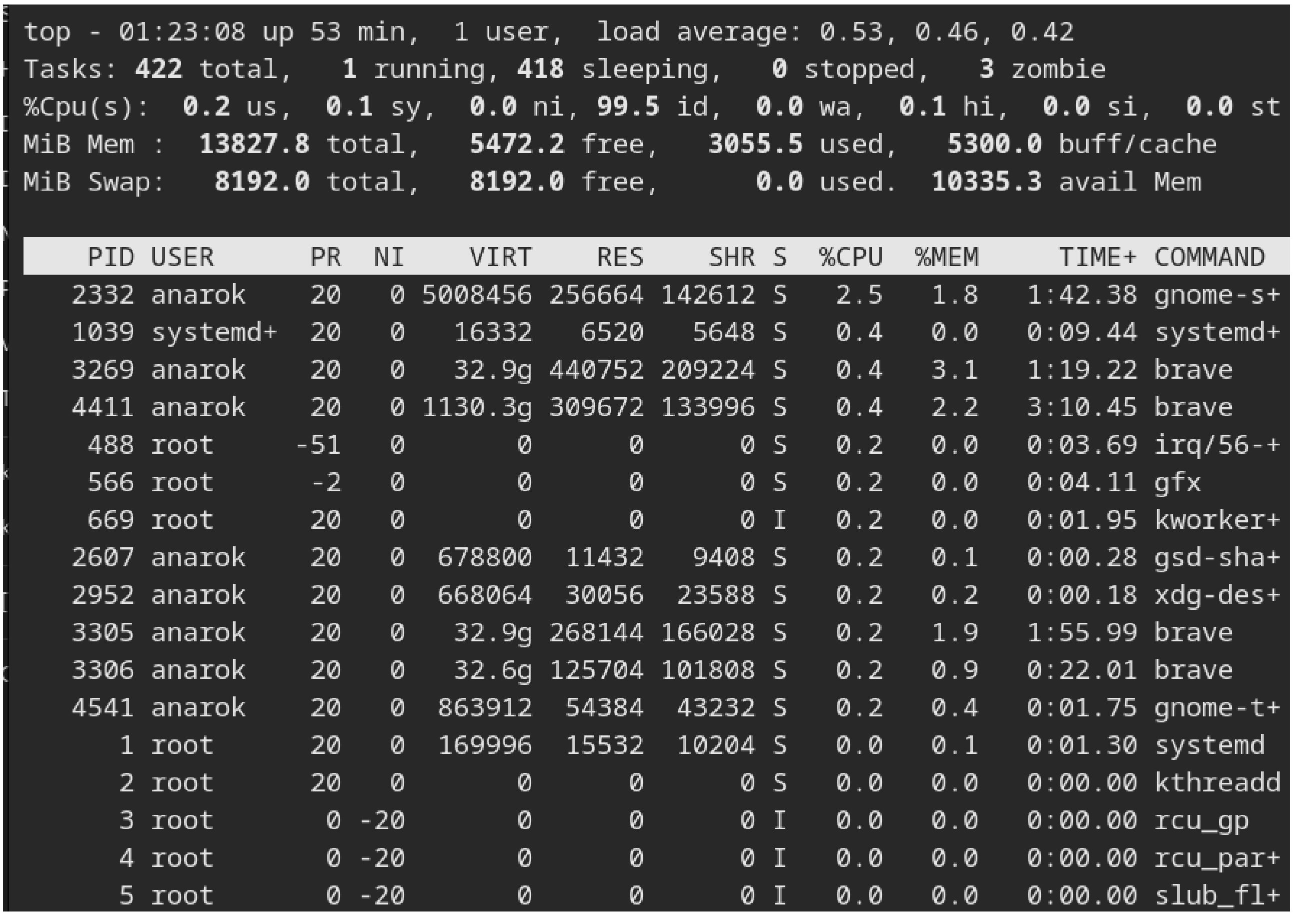
Ps command is used to list the currently running processes and their PIDs along with some other information depending on different options.



```
[anarok@fedora ~]$ ps
    PID TTY          TIME CMD
   4567 pts/0    00:00:00 zsh
   5090 pts/0    00:00:00 bash
   7798 pts/0    00:00:00 ps
```

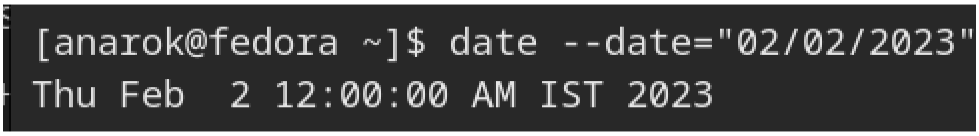
17) top

Top command is used to show the Linux processes. It provides a dynamic real-time view of the running system.



```
top - 01:23:08 up 53 min,  1 user,  load average: 0.53, 0.46, 0.42
Tasks: 422 total,   1 running, 418 sleeping,   0 stopped,   3 zombie
%Cpu(s):  0.2 us,  0.1 sy,  0.0 ni, 99.5 id,  0.0 wa,  0.1 hi,  0.0 si,  0.0 st
MiB Mem :  13827.8 total,   5472.2 free,   3055.5 used,   5300.0 buff/cache
MiB Swap:   8192.0 total,   8192.0 free,      0.0 used.  10335.3 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
   2332 anarok    20   0 5008456 256664 142612 S   2.5   1.8   1:42.38 gnome-s+
   1039 systemd+  20   0   16332   6520   5648 S   0.4   0.0   0:09.44 systemd+
   3269 anarok    20   0   32.9g 440752 209224 S   0.4   3.1   1:19.22 brave
   4411 anarok    20   0 1130.3g 309672 133996 S   0.4   2.2   3:10.45 brave
    488 root     -51   0       0      0      0 S   0.2   0.0   0:03.69 irq/56-+
    566 root      -2   0       0      0      0 S   0.2   0.0   0:04.11 gfx
    669 root      20   0       0      0      0 I   0.2   0.0   0:01.95 kworker+
   2607 anarok    20   0  678800  11432   9408 S   0.2   0.1   0:00.28 gsd-sha+
   2952 anarok    20   0  668064  30056  23588 S   0.2   0.2   0:00.18 xdg-des+
   3305 anarok    20   0   32.9g 268144 166028 S   0.2   1.9   1:55.99 brave
   3306 anarok    20   0   32.6g 125704 101808 S   0.2   0.9   0:22.01 brave
   4541 anarok    20   0  863912  54384  43232 S   0.2   0.4   0:01.75 gnome-t+
      1 root      20   0  169996  15532  10204 S   0.0   0.1   0:01.30 systemd
      2 root      20   0       0      0      0 S   0.0   0.0   0:00.00 kthreadd
      3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
      4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_par+
      5 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 slub_fl+
```

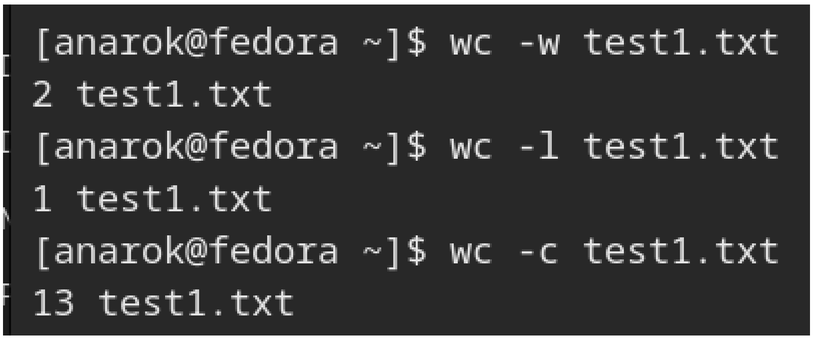
18) date –date



```
[anarok@fedora ~]$ date --date="02/02/2023"
Thu Feb  2 12:00:00 AM IST 2023
```

19) wc -w, wc -l, wc -c

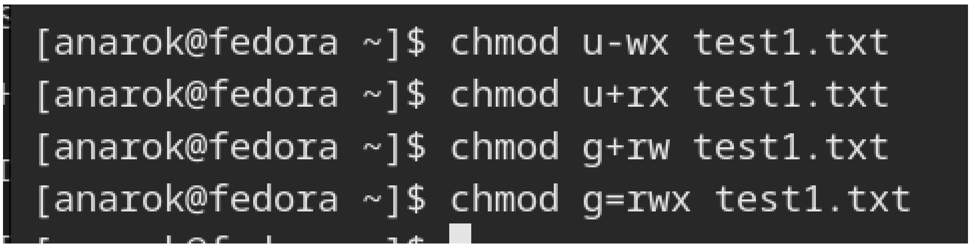
Gives the count of words, lines and characters in the given file.



```
[anarok@fedora ~]$ wc -w test1.txt
2 test1.txt
[anarok@fedora ~]$ wc -l test1.txt
1 test1.txt
[anarok@fedora ~]$ wc -c test1.txt
13 test1.txt
```

20) chmod

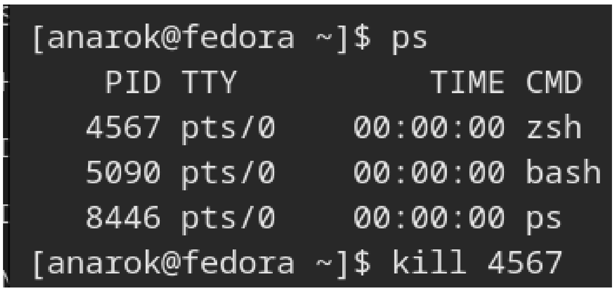
Use to change permissions of a file or directory. (chmod -change mode) Syntax: ch mod category operation permission file. Assigns write and execute permissions for users (- used to reduce power).



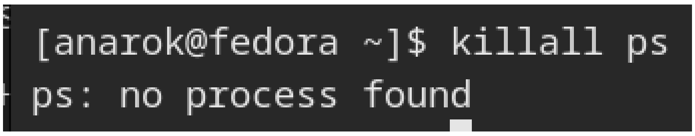
```
[anarok@fedora ~]$ chmod u-wx test1.txt
[anarok@fedora ~]$ chmod u+rx test1.txt
[anarok@fedora ~]$ chmod g+rw test1.txt
[anarok@fedora ~]$ chmod g=rwx test1.txt
```

21) kill pid, killall proc

kill pid kills the given process id whereas killall proc kills all the process running



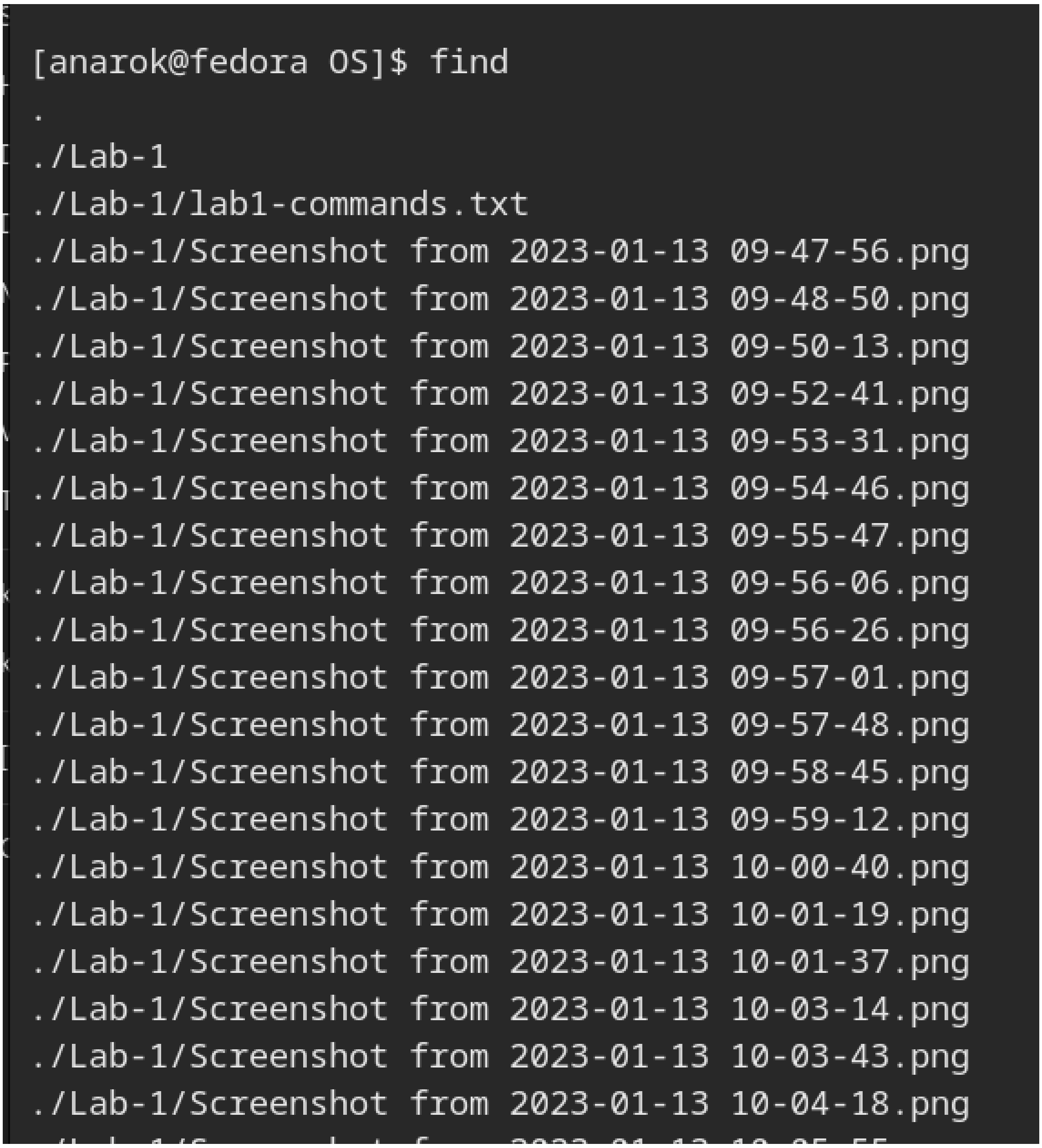
```
[anarok@fedora ~]$ ps
    PID TTY          TIME CMD
   4567 pts/0    00:00:00 zsh
   5090 pts/0    00:00:00 bash
   8446 pts/0    00:00:00 ps
[anarok@fedora ~]$ kill 4567
```



```
[anarok@fedora ~]$ killall ps
ps: no process found
```

22) find

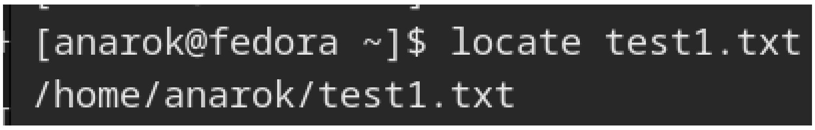
The find command in UNIX is a command line utility for walking a file hierarchy. It can be used to find files and directories and perform subsequent operations on them.



```
[anarok@fedora OS]$ find
.
./Lab-1
./Lab-1/lab1-commands.txt
./Lab-1/Screenshot from 2023-01-13 09-47-56.png
./Lab-1/Screenshot from 2023-01-13 09-48-50.png
./Lab-1/Screenshot from 2023-01-13 09-50-13.png
./Lab-1/Screenshot from 2023-01-13 09-52-41.png
./Lab-1/Screenshot from 2023-01-13 09-53-31.png
./Lab-1/Screenshot from 2023-01-13 09-54-46.png
./Lab-1/Screenshot from 2023-01-13 09-55-47.png
./Lab-1/Screenshot from 2023-01-13 09-56-06.png
./Lab-1/Screenshot from 2023-01-13 09-56-26.png
./Lab-1/Screenshot from 2023-01-13 09-57-01.png
./Lab-1/Screenshot from 2023-01-13 09-57-48.png
./Lab-1/Screenshot from 2023-01-13 09-58-45.png
./Lab-1/Screenshot from 2023-01-13 09-59-12.png
./Lab-1/Screenshot from 2023-01-13 10-00-40.png
./Lab-1/Screenshot from 2023-01-13 10-01-19.png
./Lab-1/Screenshot from 2023-01-13 10-01-37.png
./Lab-1/Screenshot from 2023-01-13 10-03-14.png
./Lab-1/Screenshot from 2023-01-13 10-03-43.png
./Lab-1/Screenshot from 2023-01-13 10-04-18.png
```

23) locate

Locate command in Linux is used to find the files by name.

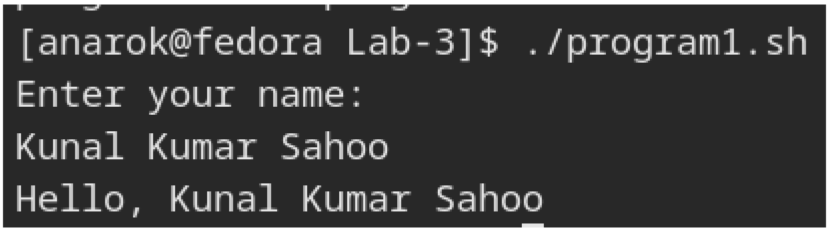


```
[anarok@fedora ~]$ locate test1.txt
/home/anarok/test1.txt
```

Experiment-3: Shell Scripting

1) **Write a shell script to print your name**

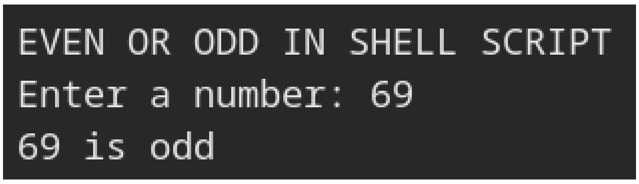
```
#!/bin/sh
echo "Enter your name: "
read NAME
echo "Hello, $NAME"
```



```
[anarok@fedora Lab-3]$ ./program1.sh
Enter your name:
Kunal Kumar Sahoo
Hello, Kunal Kumar Sahoo
```

2) **Write a shell script to find whether a number is even or odd**

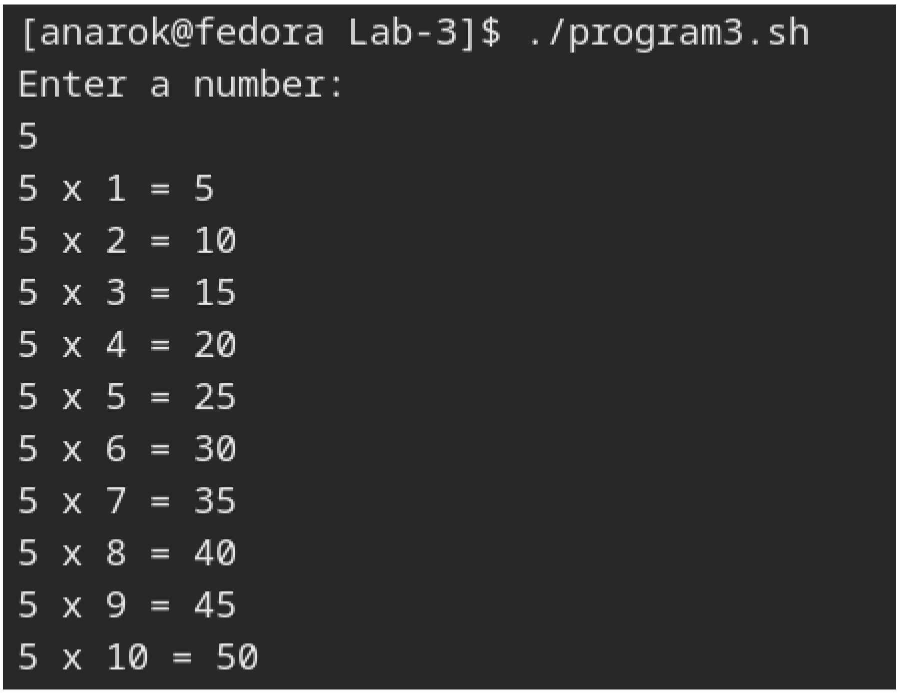
```
#!/bin/sh
clear
echo "EVEN OR ODD IN SHELL SCRIPT"
echo -n "Enter a number: "
read n
if [ `expr $n % 2` == 0 ]
then
   echo "$n is even"
else
   echo "$n is odd"
fi
```



```
EVEN OR ODD IN SHELL SCRIPT
Enter a number: 69
69 is odd
```

3) Write a shell script to print table of a given number

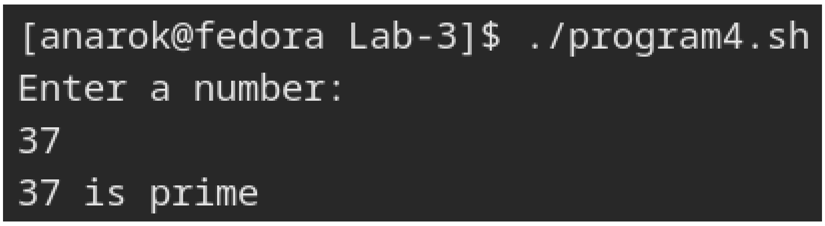
```
#!/bin/sh
echo "Enter a number: "
read n
i=1
while [ $i -le 10 ]
do
   echo "$n x $i = $(( n * i ))"
   i=$(( i + 1 ))
done
```



```
[anarok@fedora Lab-3]$ ./program3.sh
Enter a number:
5
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50
```

4) Write a shell script to check whether a given number is prime or not.

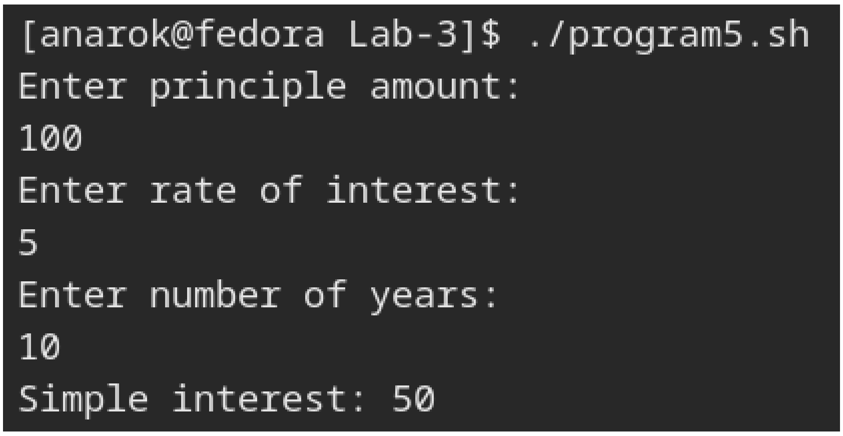
```
#!/bin/sh
echo "Enter a number: "
read number
for (( i=2;i<=$number/2;i++ ))
do
   if [ $(( number%i )) -eq 0 ]
   then
      echo "$number is composite"
      exit
   fi
done
echo "$number is prime"
```



```
[anarok@fedora Lab-3]$ ./program4.sh
Enter a number:
37
37 is prime
```

5) **Write a shell script to find the simple interest**

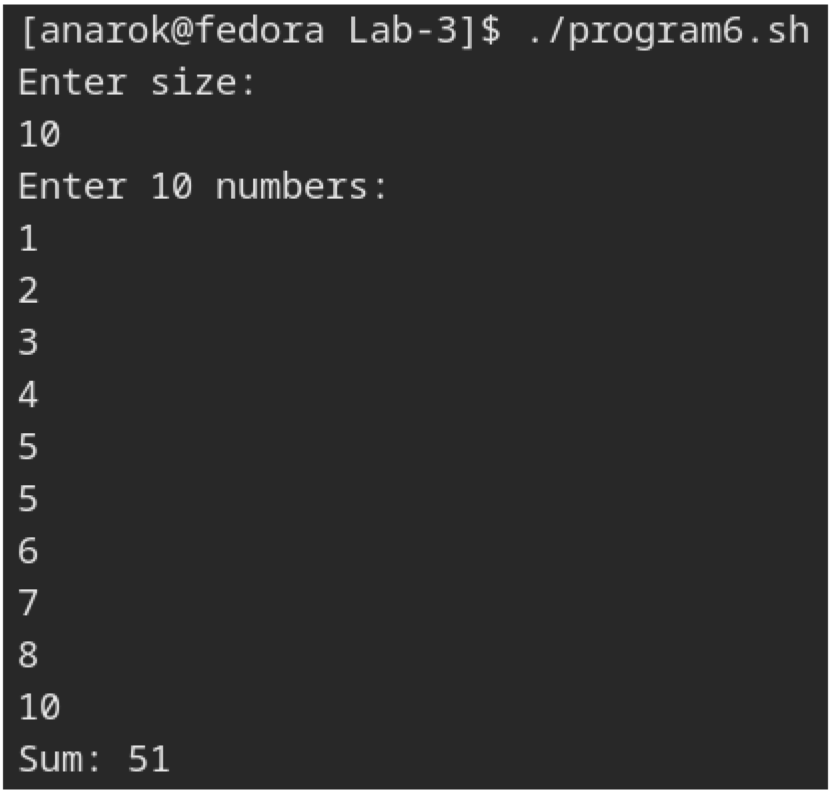
```
#!/bin/sh

echo "Enter principle amount: "
read p

echo "Enter rate of interest: "
read r

echo "Enter number of years: "
read t

s=$(( p*r*t/100 ))
echo "Simple interest: $s"
```



```
[anarok@fedora Lab-3]$ ./program5.sh
Enter principle amount:
100
Enter rate of interest:
5
Enter number of years:
10
Simple interest: 50
```

6) **Write a shell script to find the sum of n numbers**

```
#!/bin/sh
echo "Enter size: "
read N
sum=0
echo "Enter $N numbers:"
for (( i=0;i<N;i++ ))
do
   read n
   sum=$(( sum+n  ))
done
echo "Sum: $sum"
```

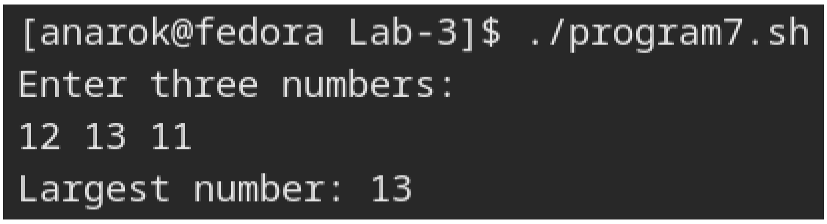


```
[anarok@fedora Lab-3]$ ./program6.sh
Enter size:
10
Enter 10 numbers:
1
2
3
4
5
5
6
7
8
10
Sum: 51
```

7) **Write a shell script to find the largest out of three numbers.**

```
#!/bin/sh

echo "Enter three numbers: "
read n1 n2 n3

if [ $n1 -gt $n2 ] && [ $n1 -gt $n3 ]
then
   max=$n1
elif [ $n2 -gt $n1 ] && [ $n2 -gt $n3 ]
```

```
then
   max=$n2
else
   max=$n2
fi

echo "Largest number: $max"
```

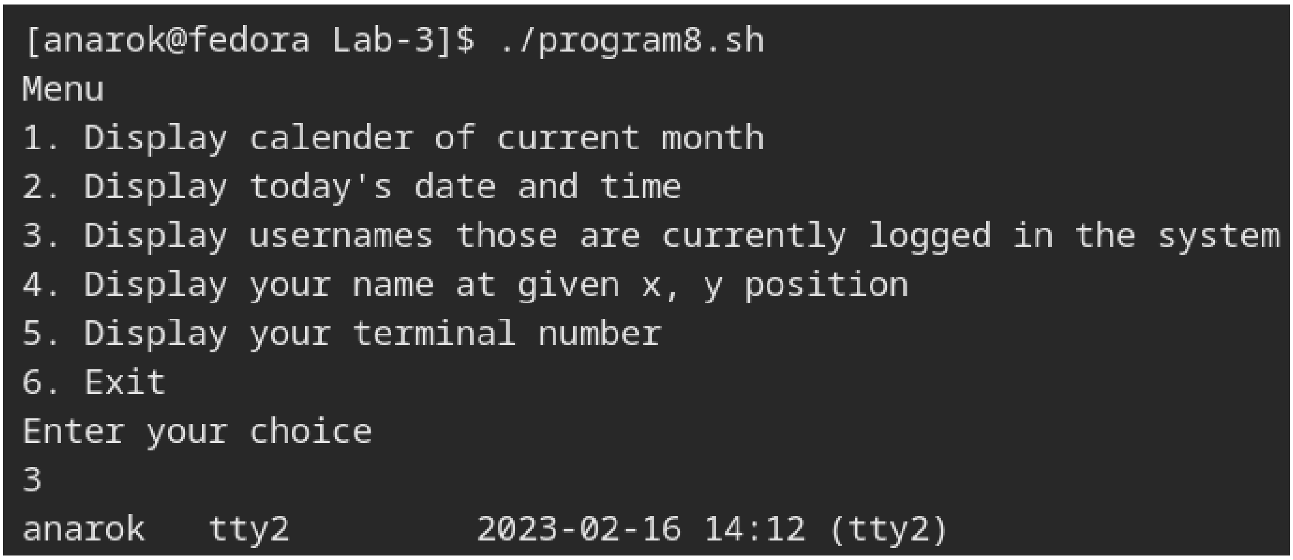


```
[anarok@fedora Lab-3]$ ./program7.sh
Enter three numbers:
12 13 11
Largest number: 13
```

8) Write a menu driven shell script will point the following menu and execute the given task.
a. Display calendar of current month
b. Display today’s date and time
c. Display username those are currently logged in the system
d. Display your name at the given x,y position.
e. Display your terminal number.

```
#!/bin/sh

echo "Menu"
echo "1. Display calender of current month"
echo "2. Display today's date and time"
echo "3. Display usernames those are currently logged in the system"
echo "4. Display your name at given x, y position"
echo "5. Display your terminal number"
echo "6. Exit"
echo "Enter your choice"

read c

case $c in
   1) cal;;
   2) date;;
   3) who;;
   4) clear
      echo "Enter x, y position: "
      read x y
```

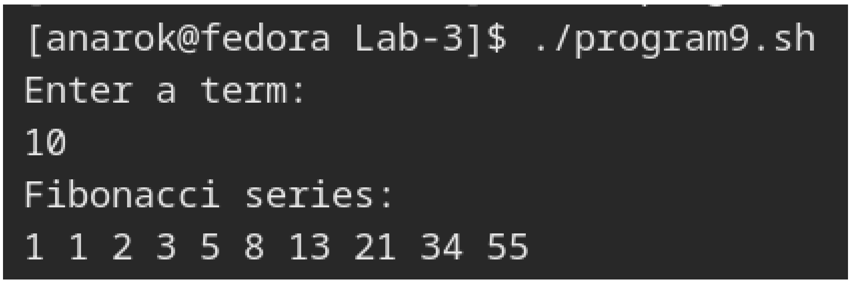
```
        tput cup $x $y
        whoami;;
    5) tty;;
    6) exit;;
esac
```



```
[anarok@fedora Lab-3]$ ./program8.sh
Menu
1. Display calender of current month
2. Display today's date and time
3. Display usernames those are currently logged in the system
4. Display your name at given x, y position
5. Display your terminal number
6. Exit
Enter your choice
3
anarok   tty2         2023-02-16 14:12 (tty2)
```

9) Write a shell script to generate the first n fibonacci numbers

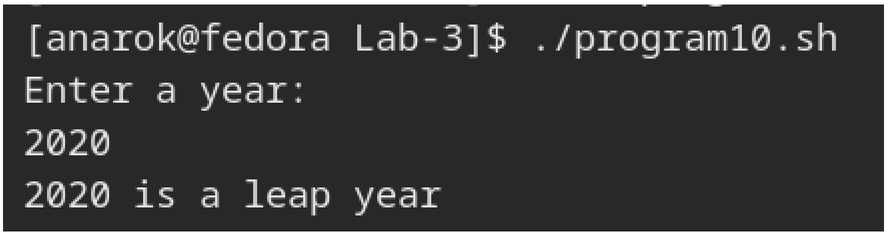
```
#!/bin/sh
echo "Enter a term: "
read n
echo "Fibonacci series: "
a=0
b=1
echo -n "$b "
for ((i=1;i<$n;i++))
do
    c=$(( a+b ))
    a=$b
    b=$c
    echo -n "$c "
done
echo
```



```
[anarok@fedora Lab-3]$ ./program9.sh
Enter a term:
10
Fibonacci series:
1 1 2 3 5 8 13 21 34 55
```

10) Write a shell script to find whether a given year is a leap year or not.

```
#!/bin/sh

echo "Enter a year: "
read year

if [ `expr $year % 400` -eq 0 ]
then
   echo "$year is a leap year"
elif [ `expr $year % 100` -eq 0 ]
then
   echo "$year is not a leap year"
elif [ `expr $year % 4` -eq 0 ]
then
   echo "$year is a leap year"
else
   echo "$year is not a leap year"
fi
```

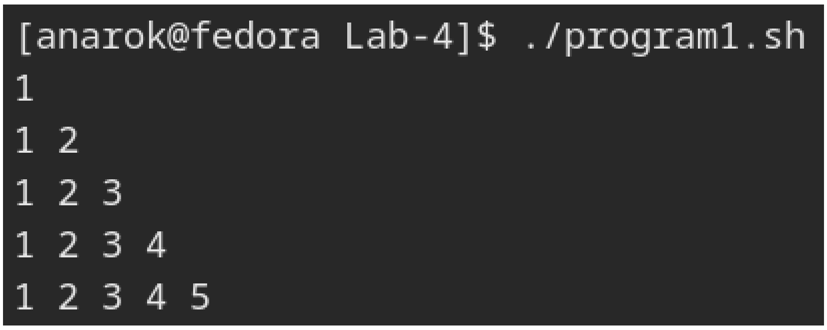


```
[anarok@fedora Lab-3]$ ./program10.sh
Enter a year:
2020
2020 is a leap year
```

Experiment-4: Shell Scripting

1) **Write a shell script to print half pyramid using numbers**

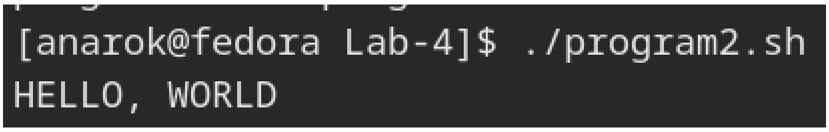
```
#!/bin/sh
number=1
rows=5
for((i=1; i<=rows; i++))
do
    for((j=1; j<=i; j++))
    do
        echo -n "$number "
        number=$((number+1))
    done
    number=1
    echo ""
done
```



```
[anarok@fedora Lab-4]$ ./program1.sh
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

2) **Write a shell script that changes text to uppercase texts**

```
#!/bin/sh
echo "Hello, World" | tr 'a-z' 'A-Z'
```

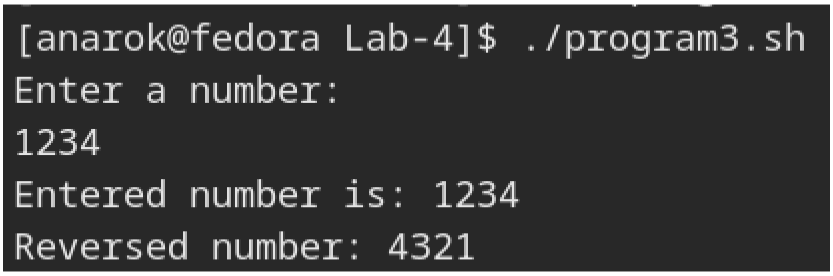


```
[anarok@fedora Lab-4]$ ./program2.sh
HELLO, WORLD
```

3) **Write a shell script to find the reverse of a given number**

```
#!/bin/sh
echo "Enter a number: "
read n
echo "Entered number is: $n"
reverse=0
while [ $n -gt 0 ]
do
```

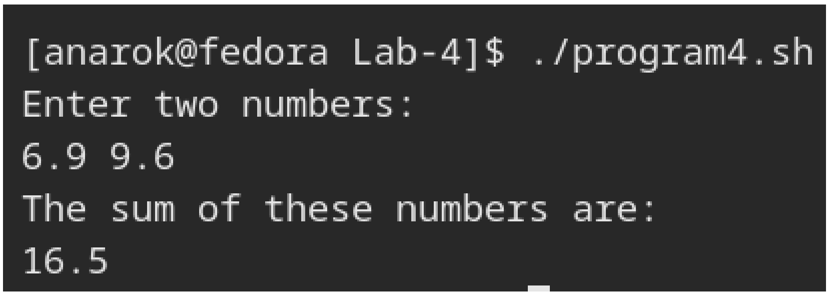
```
    r=$(( n%10 ))
    reverse=$(( reverse*10+$r ))
    n=$(( n/10 ))
done
echo "Reversed number: $reverse"
```



```
[anarok@fedora Lab-4]$ ./program3.sh
Enter a number:
1234
Entered number is: 1234
Reversed number: 4321
```

4) Write a shell script to find sum of floating point numbers

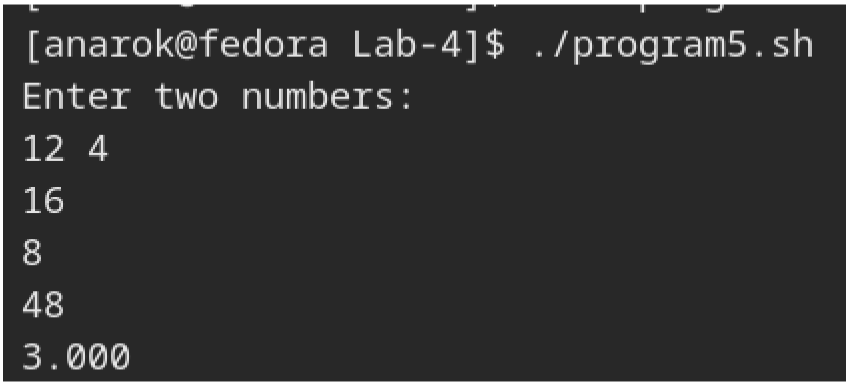
```
#!/bin/sh
echo "Enter two numbers: "
read num1 num2
echo "The sum of these numbers are: "
echo $num1 + $num2 | bc
```



```
[anarok@fedora Lab-4]$ ./program4.sh
Enter two numbers:
6.9 9.6
The sum of these numbers are:
16.5
```

5) Write a shell script to make the following operations menu based:
a) Addition
b) Subtraction
c) Multiplication
d) Division

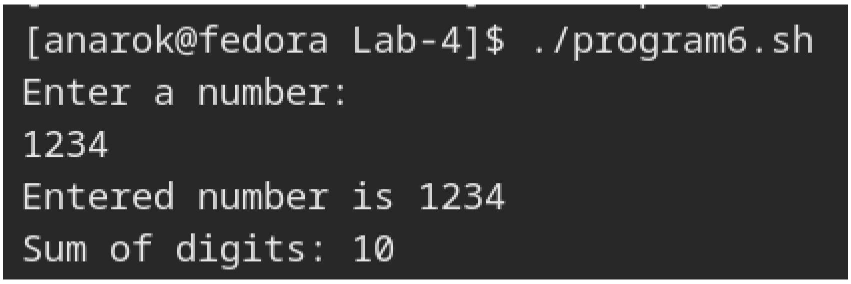
```
#!/bin/sh
echo "Enter two numbers: "
read num1 num2
echo $num1 + $num2 | bc
echo $num1 - $num2 | bc
echo $num1 \* $num2 | bc
echo "scale=3; $num1 / $num2" | bc
```



```
[anarok@fedora Lab-4]$ ./program5.sh
Enter two numbers:
12 4
16
8
48
3.000
```

6) **Write a shell script to find the sum of all digits for a given number**

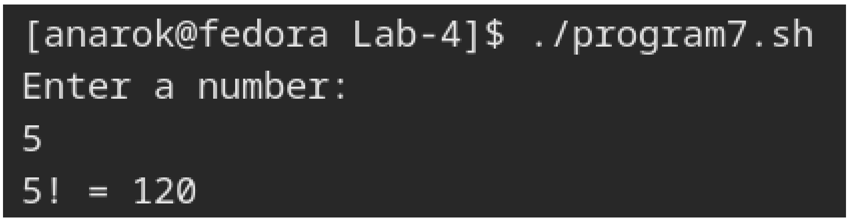
```
#!/bin/bash
echo "Enter a number: "
read n
echo "Entered number is $n"
sum=0
while [ $n -gt 0 ]
do
   x=$(( n%10 ))
   sum=$(( sum + $x ))
   n=$(( n/10 ))
done
echo "Sum of digits: $sum"
```



```
[anarok@fedora Lab-4]$ ./program6.sh
Enter a number:
1234
Entered number is 1234
Sum of digits: 10
```

7) **Write a shell script to find the factorial of a given number**

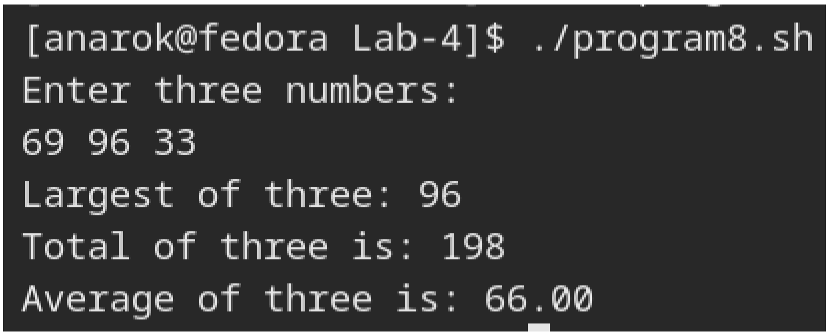
```
#!/bin/bash
echo "Enter a number: "
read n
i=1
fact=1
while [ $i -le $n ]
do
   fact=$(( fact*i ))
   i=$(( i+1 ))
done
echo "$n! = $fact"
```



```
[anarok@fedora Lab-4]$ ./program7.sh
Enter a number:
5
5! = 120
```

8) Write a shell script to find the largest of three numbers and also find the total average.

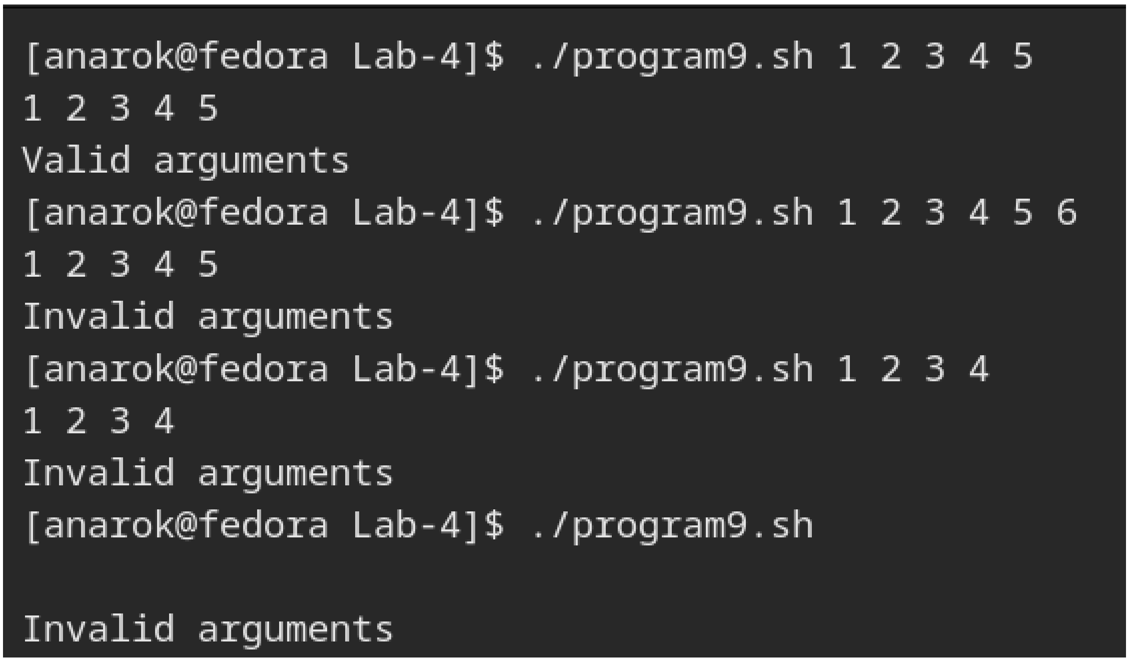
```
#!/bin/bash
echo "Enter three numbers: "
read n1 n2 n3
largest=$n1
if [ $n2 -gt $n1 ]
then
    largest=$n2
    if [ $n3 -gt $n2 ]
    then
        largest=$n3
    fi
fi
total=$(( n1 + n2 +n3 ))
avg=$(( total / 3 ))
echo "Largest of three: $largest"
echo "Total of three is: $total"
echo -n "Average of three is: "
echo "scale=2;$total/3" | bc
```



```
[anarok@fedora Lab-4]$ ./program8.sh
Enter three numbers:
69 96 33
Largest of three: 96
Total of three is: 198
Average of three is: 66.00
```

9) **Write a shell script which prints “invalid no. of arguments” if more than 5 command line arguments otherwise print “valid no. of arguments”.**

```
#!/bin/bash
echo $1 $2 $3 $4 $5
if [ $# -eq 5 ]
then
   echo "Valid arguments"
else
   echo "Invalid arguments"
fi
```

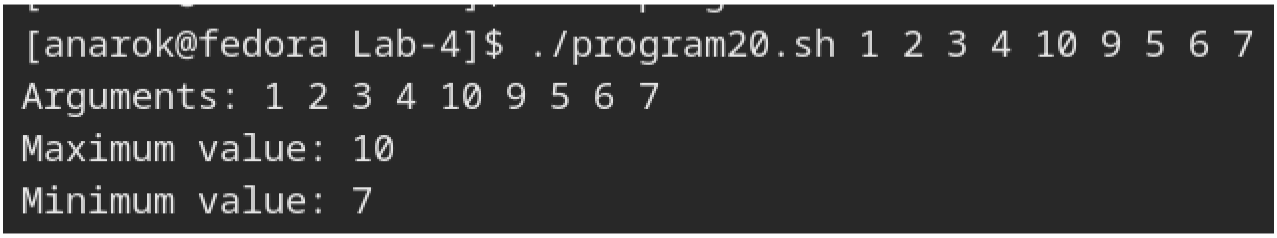


```
[anarok@fedora Lab-4]$ ./program9.sh 1 2 3 4 5
1 2 3 4 5
Valid arguments
[anarok@fedora Lab-4]$ ./program9.sh 1 2 3 4 5 6
1 2 3 4 5
Invalid arguments
[anarok@fedora Lab-4]$ ./program9.sh 1 2 3 4
1 2 3 4
Invalid arguments
[anarok@fedora Lab-4]$ ./program9.sh

Invalid arguments
```

10) **Write a shell script to find the max. and min. number from the given data set passed by command line argument.**

```
#!/bin/bash
echo "Arguments: $*"
max=$1
args=("$@")
for(( i=0;i<$#;i++ ))
do
   if [ ${args[i]} -gt $max ]
   then
      max=${args[i]}
   fi
done
echo "Maximum value: $max"
```

```
min=$1
for(( i=0;i<$#;i++ ))
do
    if [ ${args[i]} -lt $max ]
    then
        min=${args[i]}
    fi
done
echo "Minimum value: $min"
```

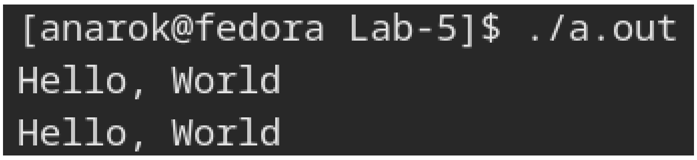


```
[anarok@fedora Lab-4]$ ./program20.sh 1 2 3 4 10 9 5 6 7
Arguments: 1 2 3 4 10 9 5 6 7
Maximum value: 10
Minimum value: 7
```

Experiment-5: Processes

1) **Write a program to create a simple child process**

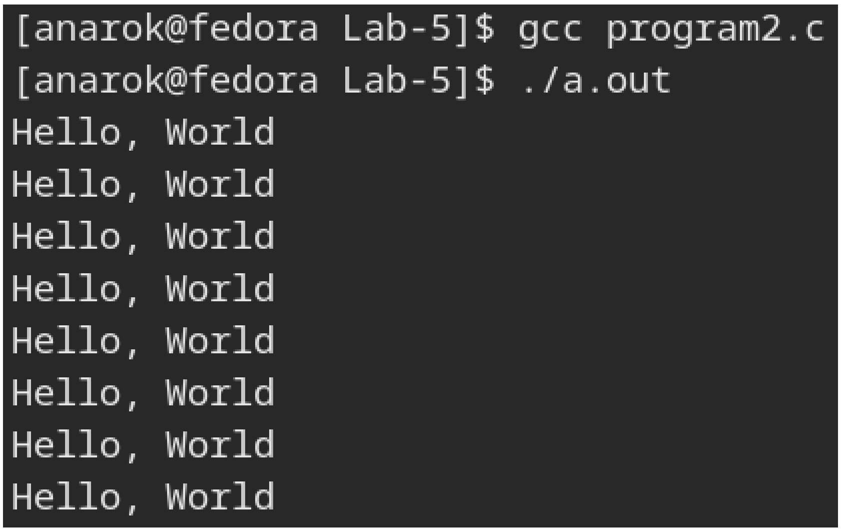
```
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>
int main() {
    fork();
    printf("Hello, World\n");
    return 0;
}
```



```
[anarok@fedora Lab-5]$ ./a.out
Hello, World
Hello, World
```

2) **Write a program to create multiple child processes.**

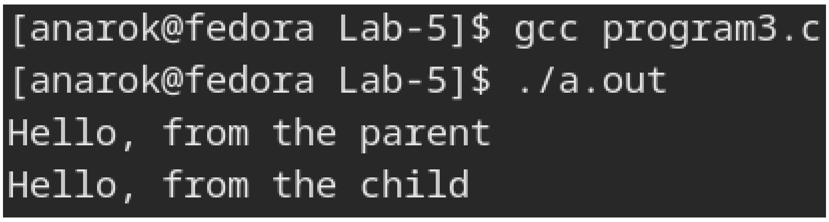
```
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>
int main() {
    fork();
    fork();
    fork();
    printf("Hello, World\n");
}
```



```
[anarok@fedora Lab-5]$ gcc program2.c
[anarok@fedora Lab-5]$ ./a.out
Hello, World
Hello, World
Hello, World
Hello, World
Hello, World
Hello, World
Hello, World
Hello, World
```

3) Write a program to check the return status of parent and child processes.

```
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>

void forkexample() {
   if(fork() == 0)
      printf("Hello, from the child\n");
   else
      printf("Hello, from the parent\n");
}

int main() {
   forkexample();
   return 0;
}
```

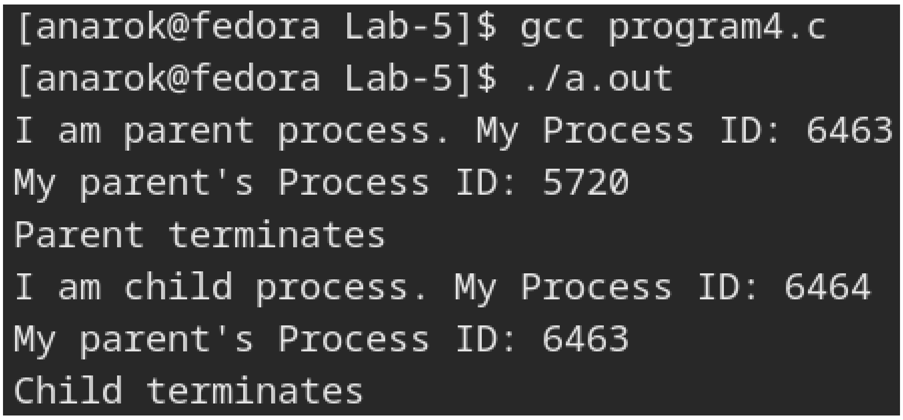


```
[anarok@fedora Lab-5]$ gcc program3.c
[anarok@fedora Lab-5]$ ./a.out
Hello, from the parent
Hello, from the child
```

4) Write a program to get the process IDs of a child process and it's parent process.

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <unistd.h>
int main() {
   int pid = fork();
   if(pid == 0) {
      printf("I am child process. My Process ID: %d\n", getpid());
      printf("My parent's Process ID: %d\n", getppid());
      printf("Child terminates\n");
      exit(0);
   } else {
      printf("I am parent process. My Process ID: %d\n", getpid());
      printf("My parent's Process ID: %d\n", getppid());
      printf("Parent terminates\n");
```

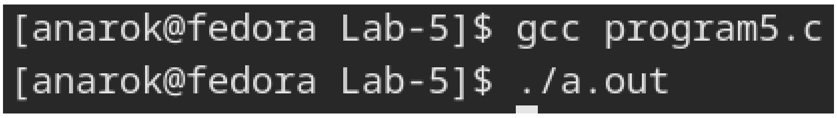
```
    }
    return 0;
}
```



```
[anarok@fedora Lab-5]$ gcc program4.c
[anarok@fedora Lab-5]$ ./a.out
I am parent process. My Process ID: 6463
My parent's Process ID: 5720
Parent terminates
I am child process. My Process ID: 6464
My parent's Process ID: 6463
Child terminates
```

5) Write a program to create a zombie process.

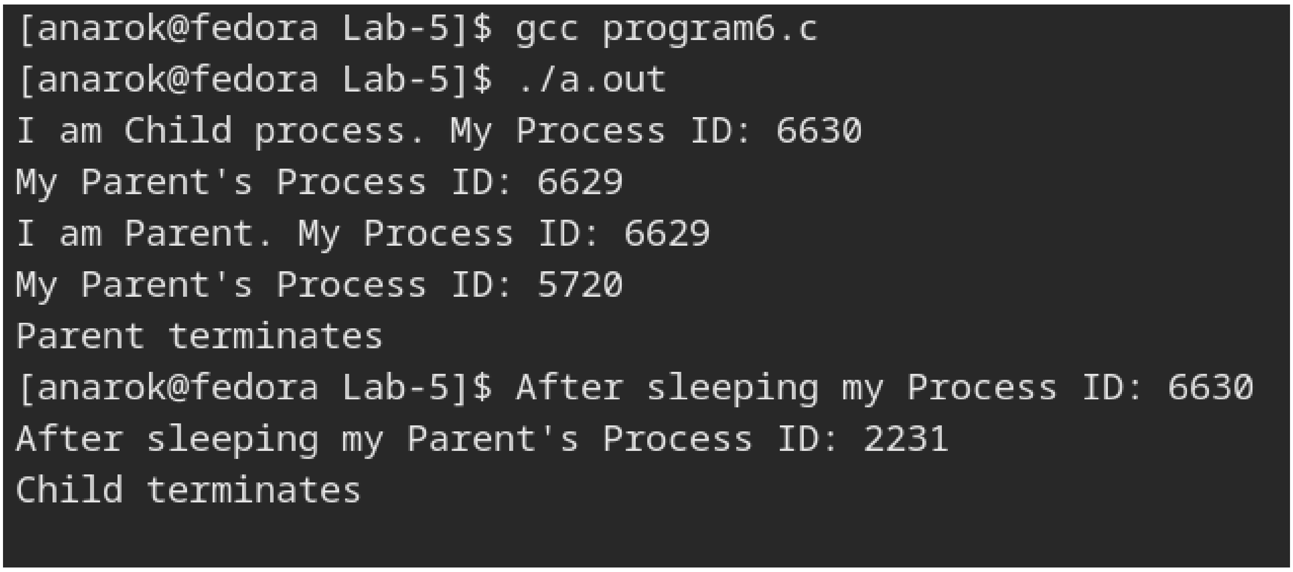
```
#include <stdlib.h>
#include <sys/types.h>
#include <unistd.h>
int main() {
    int child_pid = fork();
    if(child_pid > 0)
        sleep(60);
    else exit(0);
    return 0;
}
```



```
[anarok@fedora Lab-5]$ gcc program5.c
[anarok@fedora Lab-5]$ ./a.out
```

6) Write a program to create an orphan process.

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <unistd.h>

int main() {
    int pid = fork();
    if(pid == 0) {
        printf("I am Child process. My Process ID: %d\n", getpid());
        printf("My Parent's Process ID: %d\n", getppid());
        sleep(30);
        printf("After sleeping my Process ID: %d\n", getpid());
        printf("After sleeping my Parent's Process ID: %d\n", getppid());
        printf("Child terminates\n");
        exit(0);
    }
    else {
        sleep(20);
        printf("I am Parent. My Process ID: %d\n", getpid());
        printf("My Parent's Process ID: %d\n", getppid());
        printf("Parent terminates\n");
    }
    return 0;
}
```



```
[anarok@fedora Lab-5]$ gcc program6.c
[anarok@fedora Lab-5]$ ./a.out
I am Child process. My Process ID: 6630
My Parent's Process ID: 6629
I am Parent. My Process ID: 6629
My Parent's Process ID: 5720
Parent terminates
[anarok@fedora Lab-5]$ After sleeping my Process ID: 6630
After sleeping my Parent's Process ID: 2231
Child terminates
```