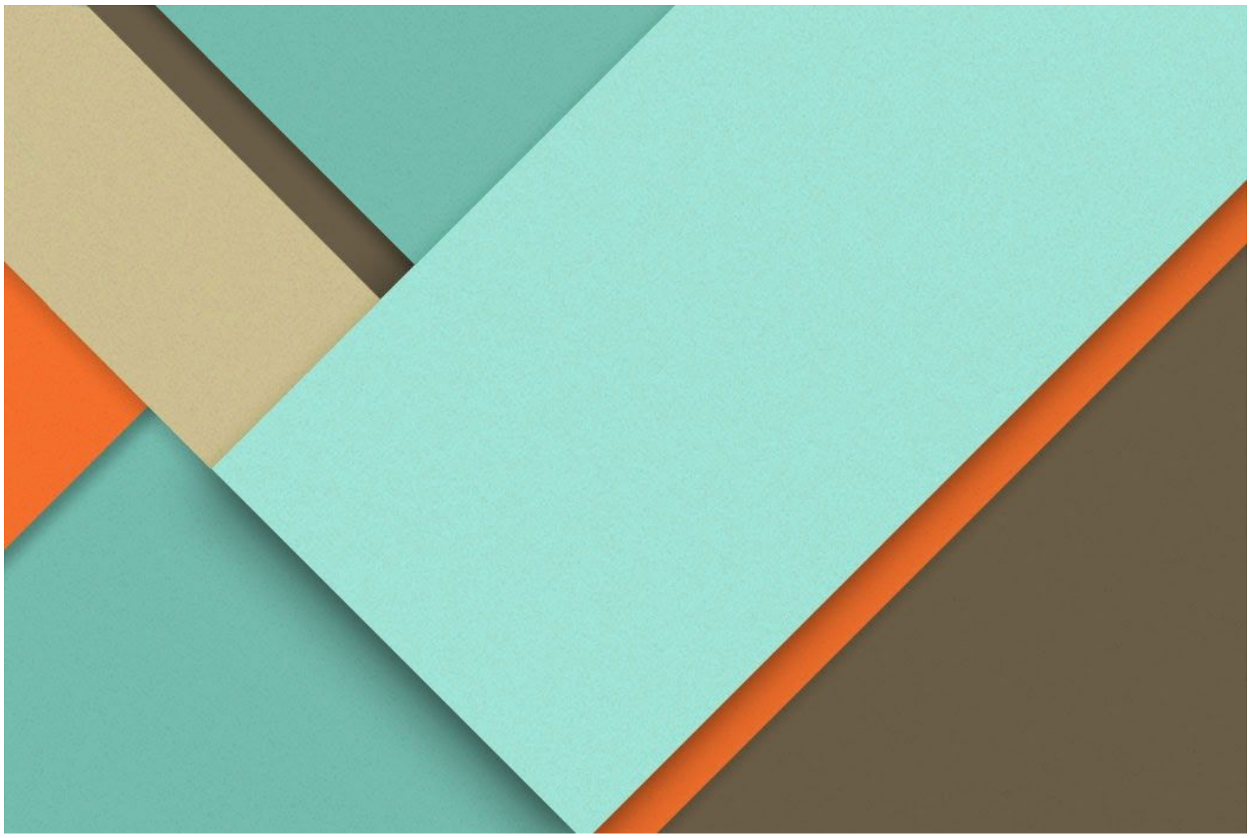


Simple Shell

Ubuntu operating system for development..

Afnan Mousa Mabrouk
ID : 15

Overview

Required to implement a Unix shell program. A shell is simply a program that conveniently allows you to run other programs.

Goals

- ➢ shell must support the following Commands:
  - The internal shell command "exit" which terminates the shell.
  - A command with no arguments.
  - A command with arguments.
  - A command, with or without arguments, executed in the background using &.

Overall organization :

- ➢ The follow of the code is to take the input command from the user in the **Store Input function** , split it by **Split function** , check if the command line contain char ‘ & ‘ to set the flag to indicate of make this process work in background,check if the command line equal ‘exit ‘ if not then call the **Execute Child function** to create the process.
- ➢ All these functions are called from the **Control function** ,which is called from the main function in the while loop which makes the user enable to insert more and more commands.
- ➢ After calling the **Control function** and entering the loop col the Signal to interrupt the termination of the child process addition to it, call the **logger function** to write in the log file .

Major functions:

The major functions :

- ➢ **Execute Child function :**
  - The mainly job of this function to create the child process by calling the fork function and after of that store the PID of this child then check if :
    1. Less than zero then the operation of the fork failed.
    2. Equal zero then execute this process.

3. Greater than zero , if this process creates to be worked in the background the parent of the process doesn't wait to terminate otherwise wait for his child terminate .

➢ **logger function :**
- This function creates a file and opens it to write in it if the process terminates and records the process PID ,Date ,Time .
- If the file is by default existing then the logger function writes in it.

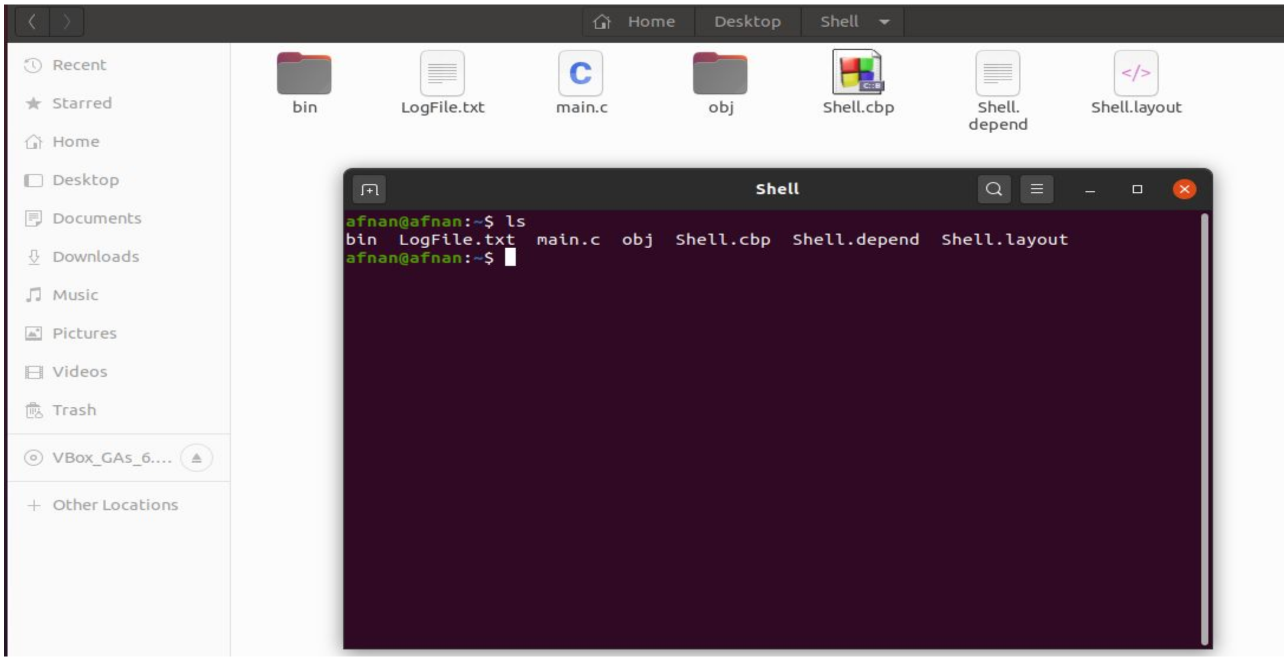
➢ **Control function :**
- Typically this function controls calling of all other functions .
- This function called in the main of the program in the while loop .

➢ **Split function :**
- Here split the input string from the user by one white space by strtok method.

Sample runs :

I. Sample run for command with no arguments

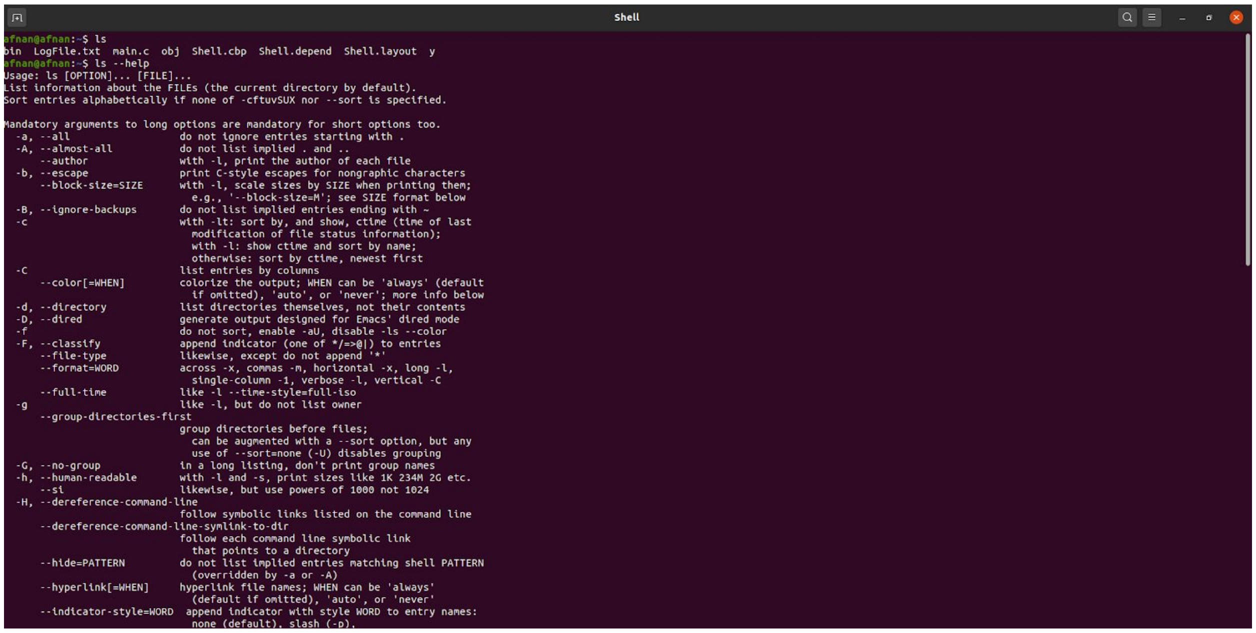
Use the " ls " command to show the folders and files in this directory .



```
afnan@afnan:~$ ls
bin  LogFile.txt  main.c  obj  Shell.cbp  Shell.depend  Shell.layout
afnan@afnan:~$ 
```

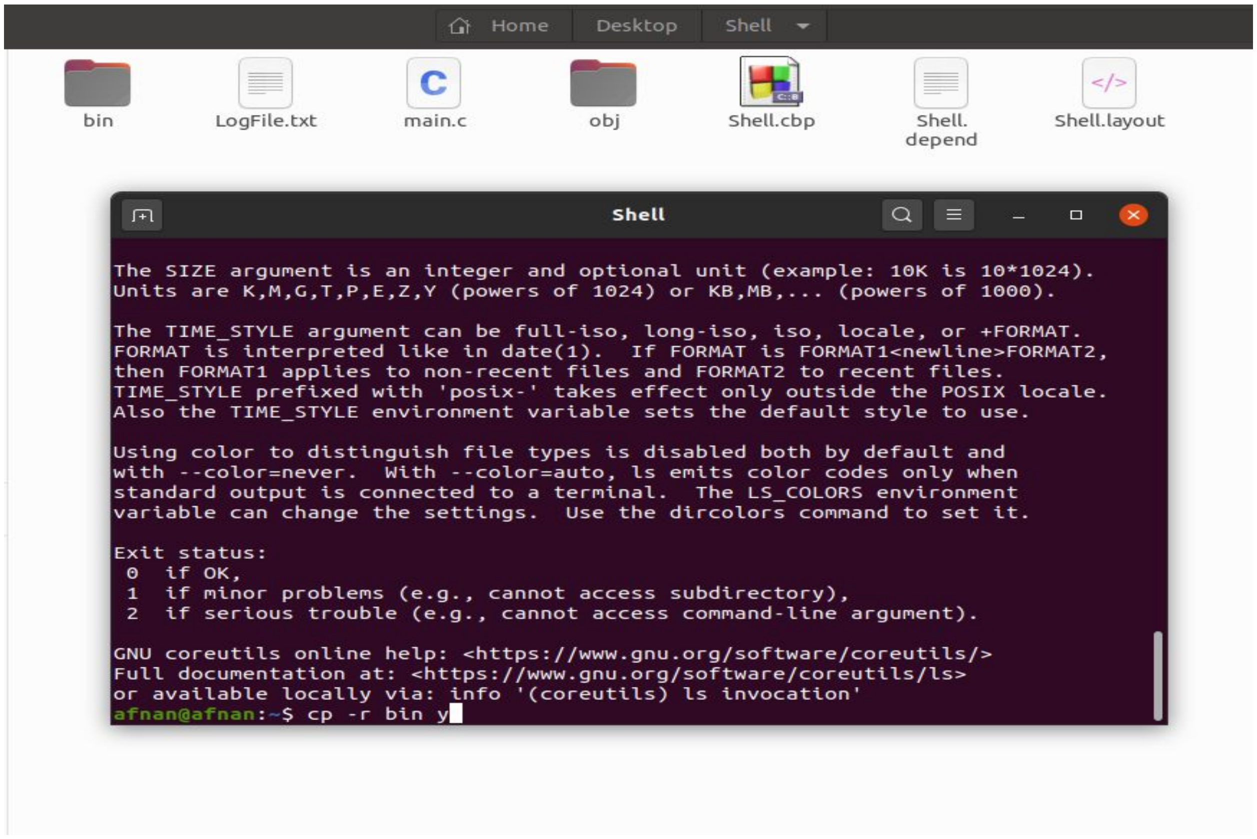
II. command with arguments

➢ Enter the ' ls --help " Command .



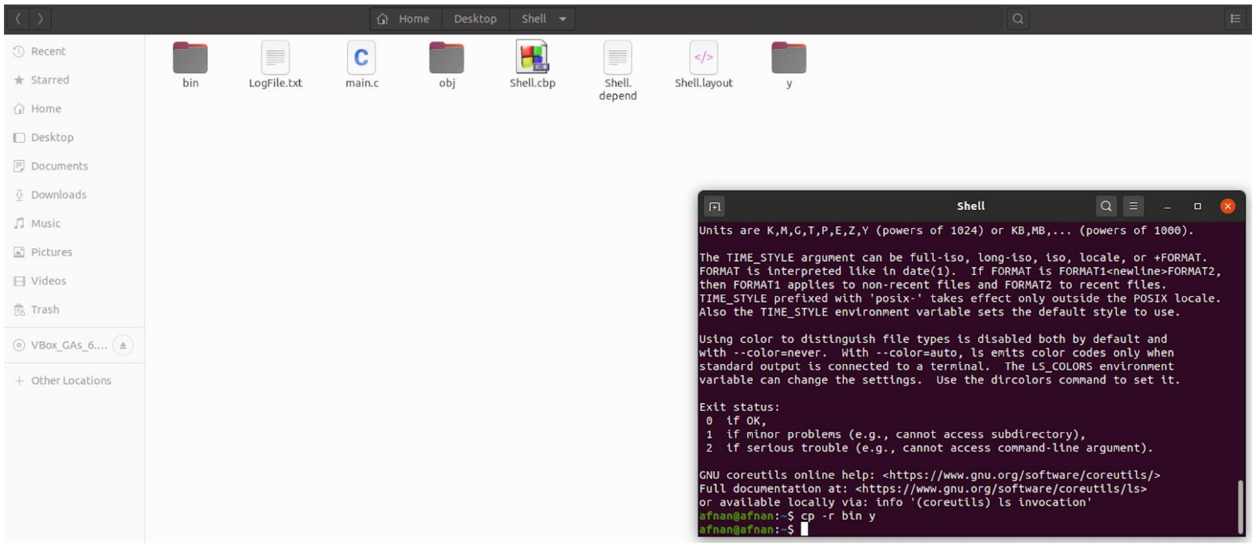
```
afnan@afnan:~$ ls
bin  LogFile.txt  main.c  obj  Shell.cbp  Shell.depend  Shell.layout  y
afnan@afnan:~$ ls --help
Usage: ls [OPTION]... [FILE]...
List information about the FILEs (the current directory by default).
Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.

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]         colorize the output; WHEN can be 'always' (default
                               if omitted), 'auto', or 'never'; more info below
  -d, --directory            list directories themselves, not their contents
  -D, --dired                generate output designed for Emacs' dired mode
  -f                         do not sort, enable -aU, disable -ls --color
  -F, --classify             append indicator (one of */=>@|) to entries
      --file-type            likewise, except do not append '*'
      --format=WORD          across -x, commas -m, horizontal -x, long -l,
                               single-column -1, verbose -l, vertical -C
      --full-time            like -l --time-style=full-iso
  -g                         like -l, but do not list owner
      --group-directories-first
                             group directories before files;
                               can be augmented with a --sort option, but any
                               use of --sort=none (-U) disables grouping
  -G, --no-group             in a long listing, don't print group names
  -h, --human-readable       with -l and -s, print sizes like 1K 234M 2G etc.
      --si                   likewise, but use powers of 1000 not 1024
  -H, --dereference-command-line
                             follow symbolic links listed on the command line
      --dereference-command-line-symlink-to-dir
                             follow each command line symbolic link
                               that points to a directory
      --hide=PATTERN         do not list implied entries matching shell PATTERN
                               (overridden by -a or -A)
      --hyperlink[=WHEN]     hyperlink file names; WHEN can be 'always'
                               (default if omitted), 'auto', or 'never'
      --indicator-style=WORD  append indicator with style WORD to entry names:
                               none (default), slash (-p),
```

➢ Insert the ' cp -r bin y " , this Command to copy all files from bin folder to the y folder after generating it,in the first picture the Shell folder before copying all files from bin and adding in the y folder .

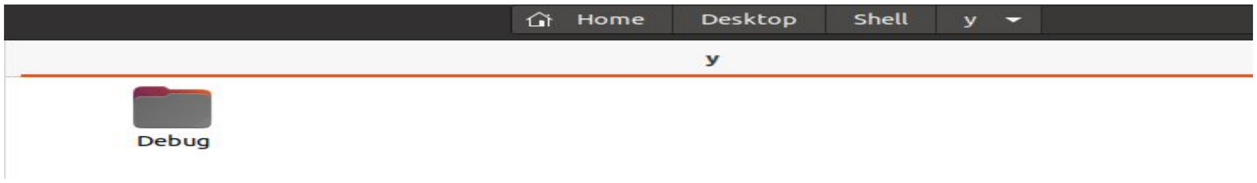


```
The SIZE argument is an integer and optional unit (example: 10K is 10*1024).
Units are K,M,G,T,P,E,Z,Y (powers of 1024) or KB,MB,... (powers of 1000).

The TIME_STYLE argument can be full-iso, long-iso, iso, locale, or +FORMAT.
FORMAT is interpreted like in date(1).  If FORMAT is FORMAT1<newline>FORMAT2,
then FORMAT1 applies to non-recent files and FORMAT2 to recent files.
TIME_STYLE prefixed with 'posix-' takes effect only outside the POSIX locale.
Also the TIME_STYLE environment variable sets the default style to use.

Using color to distinguish file types is disabled both by default and
with --color=never.  With --color=auto, ls emits color codes only when
standard output is connected to a terminal.  The LS_COLORS environment
variable can change the settings.  Use the dircolors command to set it.

Exit status:
 0  if OK,
 1  if minor problems (e.g., cannot access subdirectory),
 2  if serious trouble (e.g., cannot access command-line argument).

GNU coreutils online help: <https://www.gnu.org/software/coreutils/>
Full documentation at: <https://www.gnu.org/software/coreutils/ls>
or available locally via: info '(coreutils) ls invocation'
afnan@afnan:~$ cp -r bin y
```

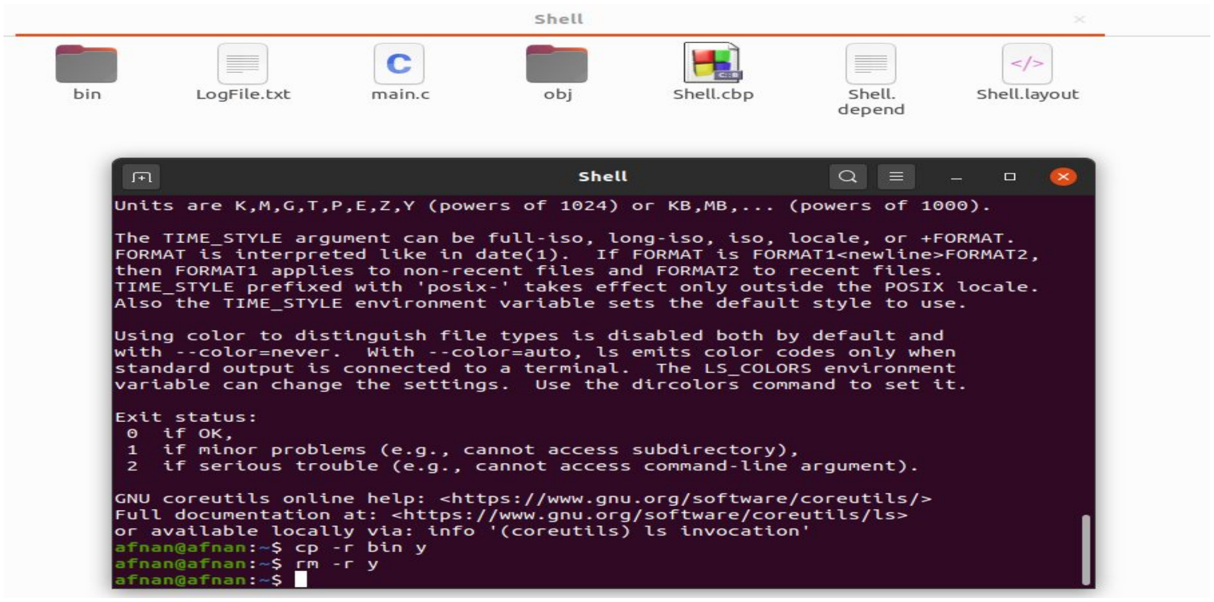
➢ After write the command line and press enter the y folder will appear in screen and contain all files.



```
Units are K,M,G,T,P,E,Z,Y (powers of 1024) or KB,MB,... (powers of 1000).

The TIME_STYLE argument can be full-iso, long-iso, iso, locale, or +FORMAT.
FORMAT is interpreted like in date(1).  If FORMAT is FORMAT1<newline>FORMAT2,
then FORMAT1 applies to non-recent files and FORMAT2 to recent files.
TIME_STYLE prefixed with 'posix-' takes effect only outside the POSIX locale.
Also the TIME_STYLE environment variable sets the default style to use.

Using color to distinguish file types is disabled both by default and
with --color=never.  With --color=auto, ls emits color codes only when
standard output is connected to a terminal.  The LS_COLORS environment
variable can change the settings.  Use the dircolors command to set it.

Exit status:
 0  if OK,
 1  if minor problems (e.g., cannot access subdirectory),
 2  if serious trouble (e.g., cannot access command-line argument).

GNU coreutils online help: <https://www.gnu.org/software/coreutils/>
Full documentation at: <https://www.gnu.org/software/coreutils/ls>
or available locally via: info '(coreutils) ls invocation'
afnan@afnan:~$ cp -r bin y
afnan@afnan:~$ 
```



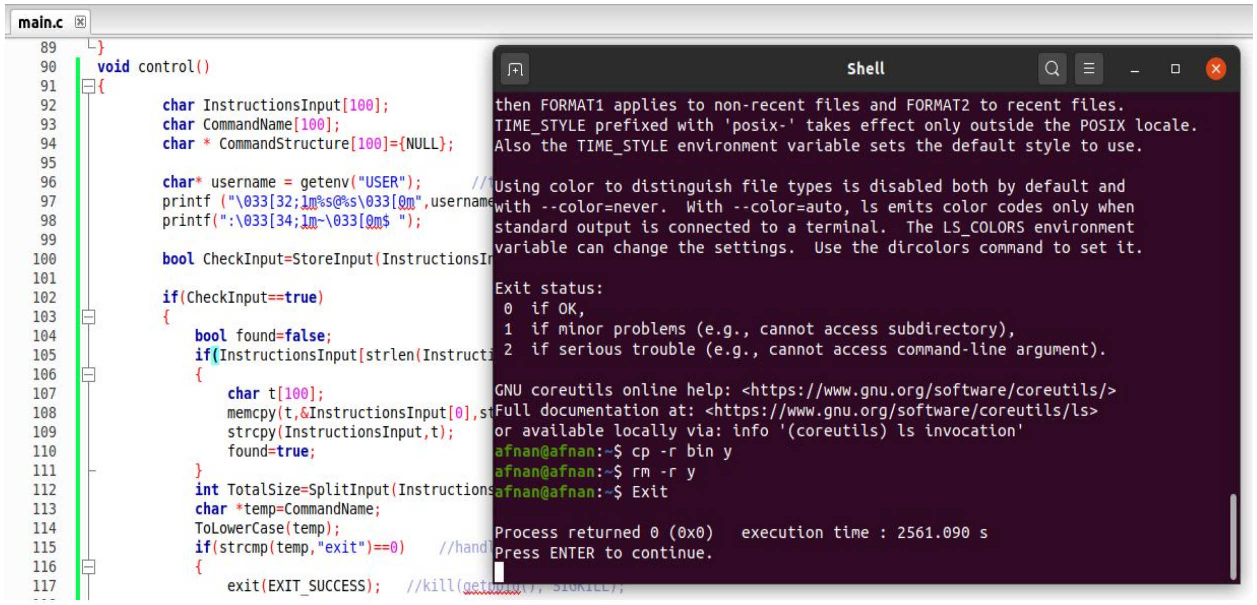
➢ To remove the folder y enter the command “rm -r y”



```
Units are K,M,G,T,P,E,Z,Y (powers of 1024) or KB,MB,... (powers of 1000).

The TIME_STYLE argument can be full-iso, long-iso, iso, locale, or +FORMAT.
FORMAT is interpreted like in date(1).  If FORMAT is FORMAT1<newline>FORMAT2,
then FORMAT1 applies to non-recent files and FORMAT2 to recent files.
TIME_STYLE prefixed with 'posix-' takes effect only outside the POSIX locale.
Also the TIME_STYLE environment variable sets the default style to use.

Using color to distinguish file types is disabled both by default and
with --color=never.  With --color=auto, ls emits color codes only when
standard output is connected to a terminal.  The LS_COLORS environment
variable can change the settings.  Use the dircolors command to set it.

Exit status:
 0  if OK,
 1  if minor problems (e.g., cannot access subdirectory),
 2  if serious trouble (e.g., cannot access command-line argument).

GNU coreutils online help: <https://www.gnu.org/software/coreutils/>
Full documentation at: <https://www.gnu.org/software/coreutils/ls>
or available locally via: info '(coreutils) ls invocation'
afnan@afnan:~$ cp -r bin y
afnan@afnan:~$ rm -r y
afnan@afnan:~$ 
```

III. The internal shell command "exit" which terminates the shell :

- ➢ The user enters the “exit” command in upper or lower case.

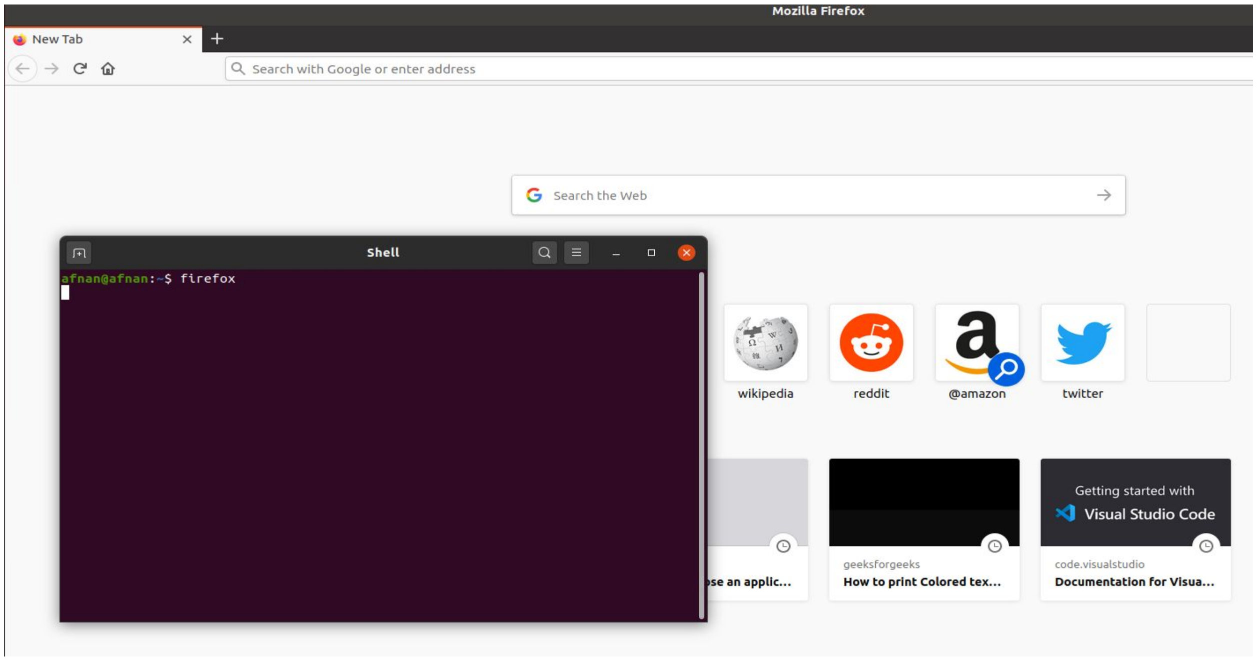


```
}
void control()
{
        char InstructionsInput[100];
        char CommandName[100];
        char * CommandStructure[100]={NULL};

        char* username = getenv("USER");      //
        printf ("\033[32;1m%s@%s\033[0m",username
        printf(":\033[34;1m~\033[0m$ ");

        bool CheckInput=StoreInput(InstructionsIn

        if(CheckInput==true)
        {
            bool found=false;
            if(InstructionsInput[strlen(Instructi
            {
                char t[100];
                memcpy(t,&InstructionsInput[0],st
                strcpy(InstructionsInput,t);
                found=true;
            }
            int TotalSize=SplitInput(Instructions
            char *temp=CommandName;
            ToLowerCase(temp);
            if(strcmp(temp,"exit")==0)    //hand
            {
                exit(EXIT_SUCCESS);    //kill(getpid(), SIGKILL);
```

```
then FORMAT1 applies to non-recent files and FORMAT2 to recent files.
TIME_STYLE prefixed with 'posix-' takes effect only outside the POSIX locale.
Also the TIME_STYLE environment variable sets the default style to use.

Using color to distinguish file types is disabled both by default and
with --color=never.  With --color=auto, ls emits color codes only when
standard output is connected to a terminal.  The LS_COLORS environment
variable can change the settings.  Use the dircolors command to set it.

Exit status:
 0  if OK,
 1  if minor problems (e.g., cannot access subdirectory),
 2  if serious trouble (e.g., cannot access command-line argument).

GNU coreutils online help: <https://www.gnu.org/software/coreutils/>
Full documentation at: <https://www.gnu.org/software/coreutils/ls>
or available locally via: info '(coreutils) ls invocation'
afnan@afnan:~$ cp -r bin y
afnan@afnan:~$ rm -r y
afnan@afnan:~$ Exit

Process returned 0 (0x0)   execution time : 2561.090 s
Press ENTER to continue.
```

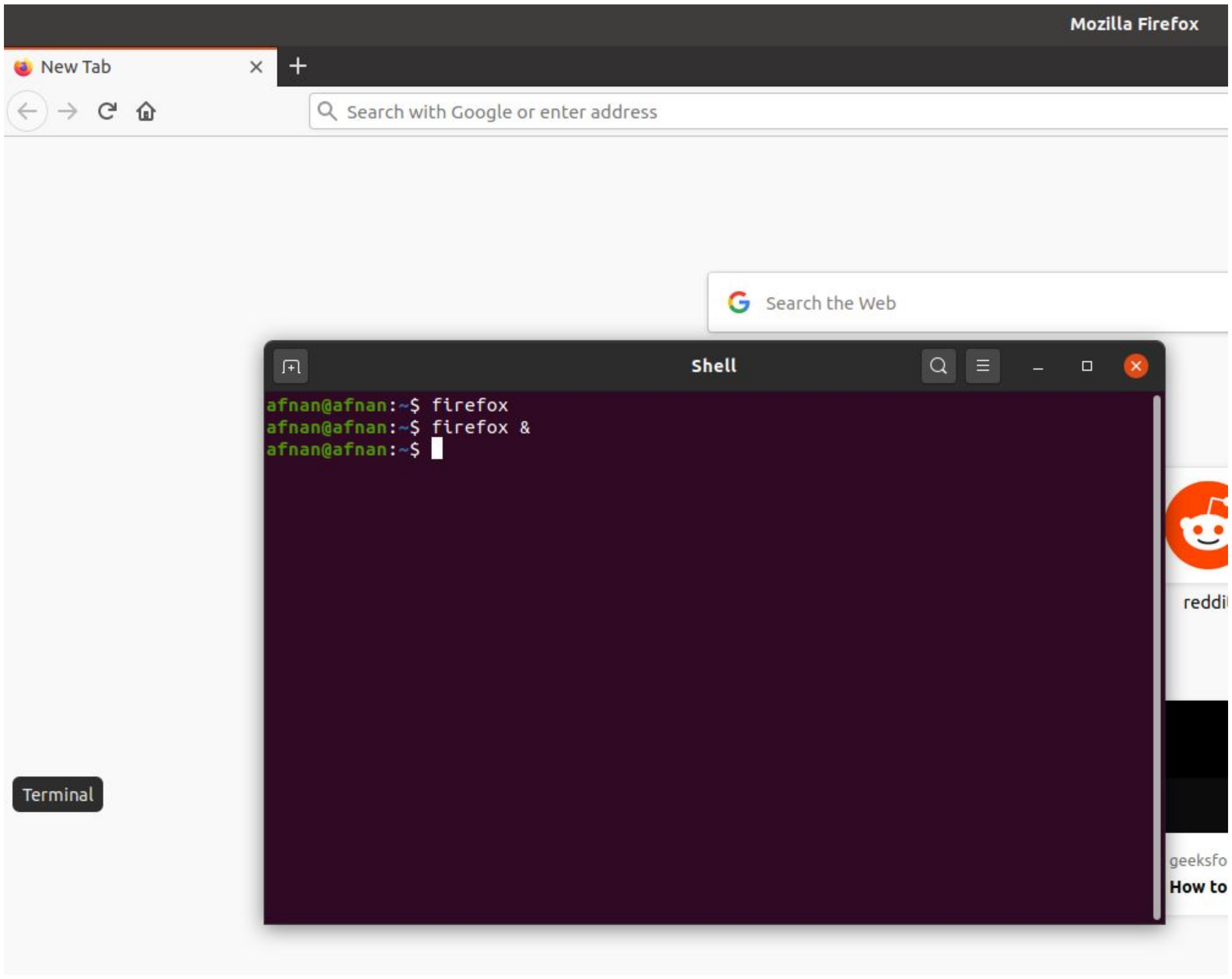
IV. A command, with or without arguments, executed in the background using &:

- ➢ In case that the command dont have the ‘&’ character that maining the parent waits for the child to terminate and doesn't take any command until the child terminates.



```
afnan@afnan:~$ firefox
```

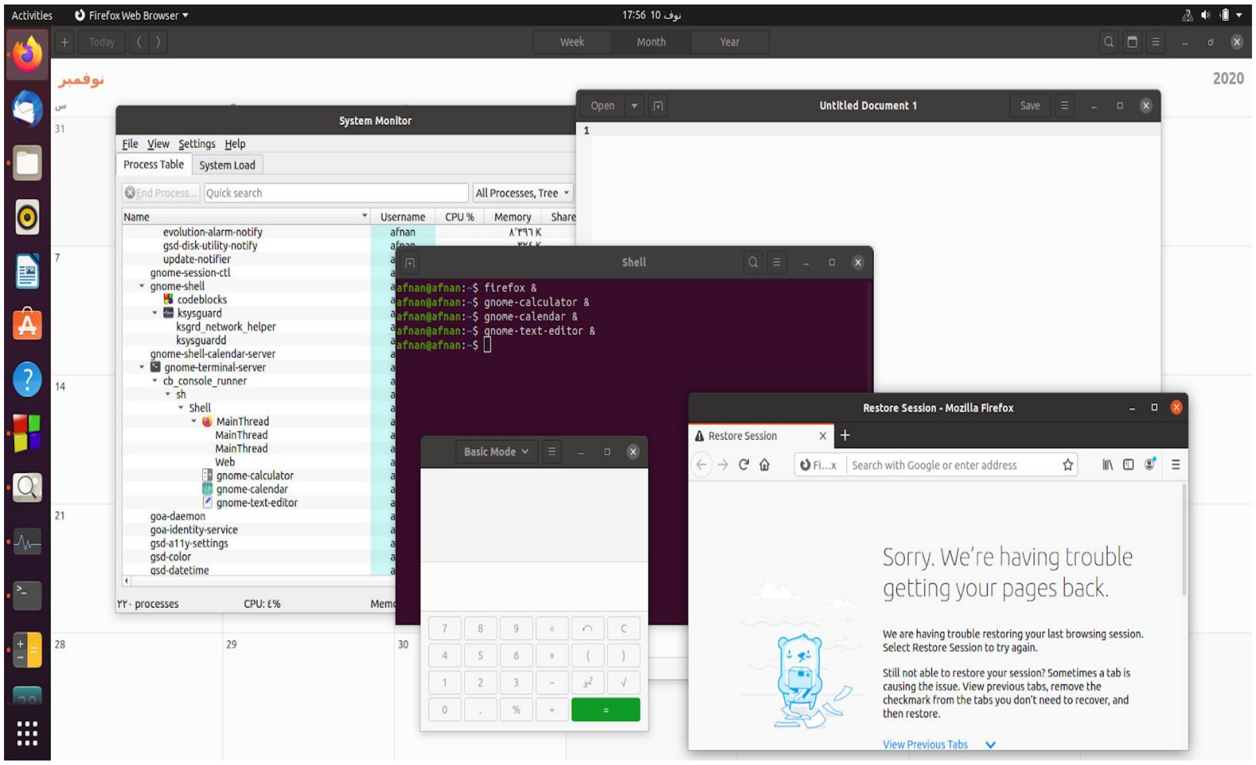
- ➢ In the above case the shell doesn't take any input until the firefox exit .
- ➢ In case that the process in the background the terminal can take any input from the user .



```
afnan@afnan:~$ firefox
afnan@afnan:~$ firefox &
afnan@afnan:~$ 
```

processes hierarchy in KSysguard :

➢ The KSysguard when open the firefox ,calculator,calendar,text editor in the background.

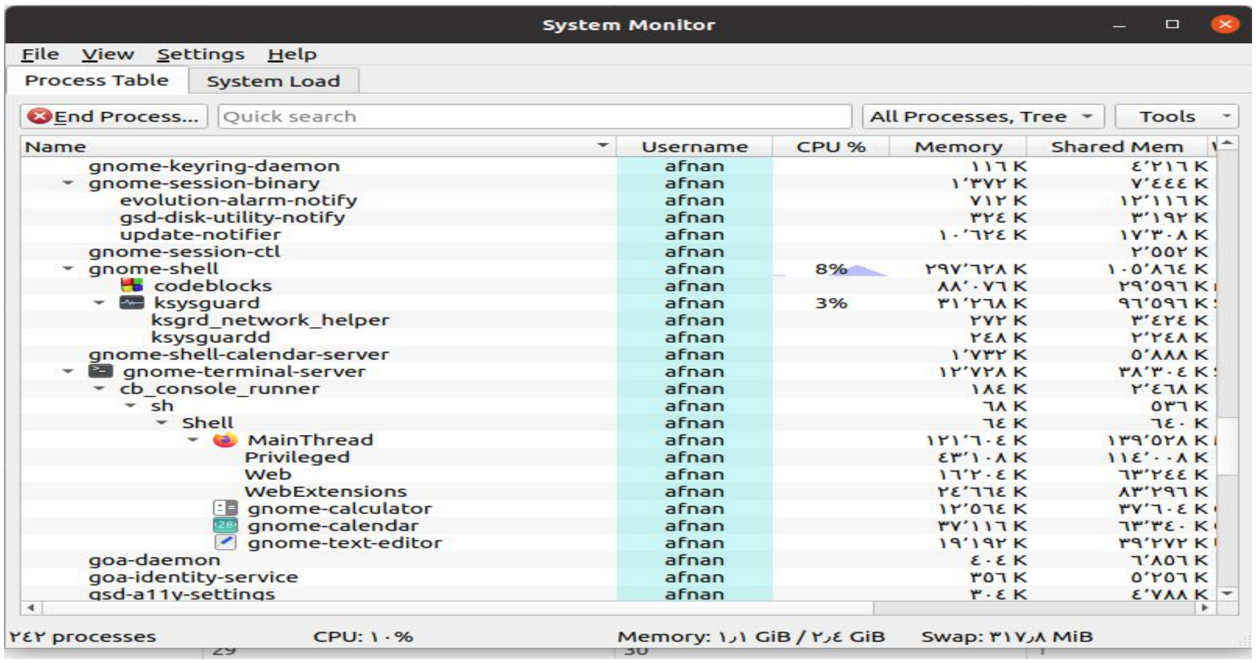


```
afnan@afnan:~$ firefox &
afnan@afnan:~$ gnome-calculator &
afnan@afnan:~$ gnome-calendar &
afnan@afnan:~$ gnome-text-editor &
afnan@afnan:~$
```

Sorry. We're having trouble getting your pages back.

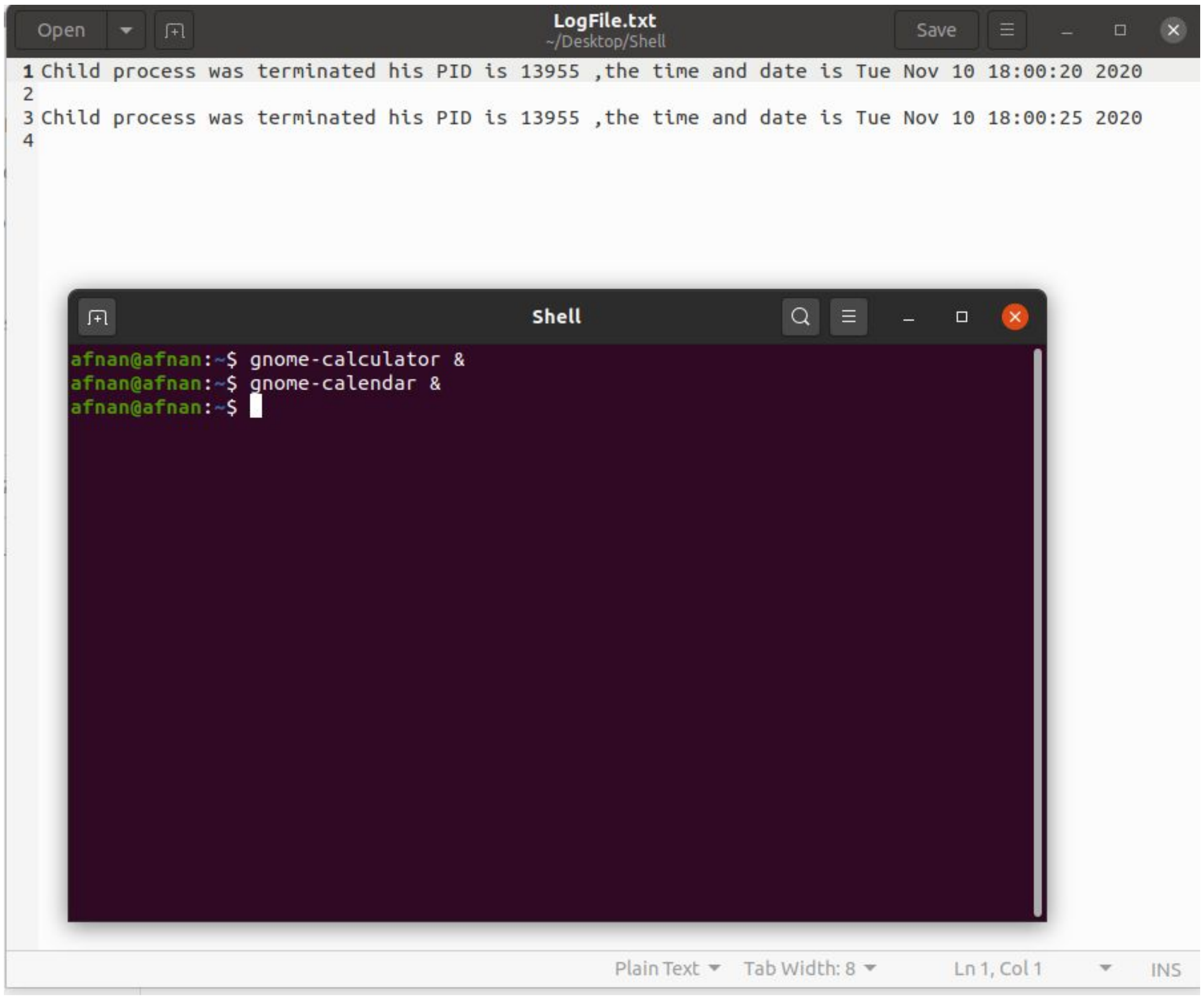
We are having trouble restoring your last browsing session. Select Restore Session to try again.

Still not able to restore your session? Sometimes a tab is causing the issue. View previous tabs, remove the checkmark from the tabs you don't need to recover, and then restore.



```
Name                              Username  CPU %  Memory      Shared Mem
    gnome-keyring-daemon          afnan            ١١٦ K       ٤’٢١٦ K
  ▾ gnome-session-binary          afnan            ١’٣٧٢ K     ٧’٤٤٤ K
        evolution-alarm-notify    afnan            ٧١٢ K       ١٢’١١٦ K
        gsd-disk-utility-notify   afnan            ٣٢٤ K       ٣’١٩٢ K
        update-notifier           afnan            ١٠’٦٢٤ K    ١٧’٣٠٨ K
    gnome-session-ctl             afnan                        ٢’٥٥٢ K
  ▾ gnome-shell                   afnan     8%     ٢٩٧’٦٢٨ K   ١٠٥’٨٦٤ K
        codeblocks                afnan            ٨٨’٠٧٦ K    ٢٩’٥٩٦ K
      ▾ ksysguard                 afnan     3%     ٣١’٢٦٨ K    ٩٦’٥٩٦ K
          ksgrd_network_helper    afnan            ٢٧٢ K       ٣’٤٢٤ K
          ksysguardd              afnan            ٢٤٨ K       ٢’٢٤٨ K
    gnome-shell-calendar-server   afnan            ١’٧٣٢ K     ٥’٨٨٨ K
  ▾ gnome-terminal-server         afnan            ١٢’٧٢٨ K    ٣٨’٣٠٤ K
      ▾ cb_console_runner         afnan            ١٨٤ K       ٢’٤٦٨ K
          ▾ sh                    afnan            ٦٨ K        ٥٣٦ K
              ▾ Shell             afnan            ٦٤ K        ٦٤٠ K
                  ▾ MainThread    afnan            ١٢١’٦٠٤ K   ١٣٩’٥٢٨ K
                      Privileged  afnan            ٤٣’١٠٨ K    ١١٤’٠٠٨ K
                      Web         afnan            ١٦’٢٠٤ K    ٦٣’٢٤٤ K
                      WebExtensions afnan          ٢٤’٦٦٤ K    ٨٣’٢٩٦ K
                    gnome-calculator afnan         ١٢’٥٦٤ K    ٣٧’٦٠٤ K
                    gnome-calendar afnan           ٣٧’١١٦ K    ٦٣’٣٤٠ K
                    gnome-text-editor afnan        ١٩’١٩٢ K    ٣٩’٢٧٢ K
    goa-daemon                    afnan            ٤٠٤ K       ٦’٨٥٦ K
    goa-identity-service          afnan            ٣٥٦ K       ٥’٢٥٦ K
    gsd-a11y-settings             afnan            ٣٠٤ K       ٤’٧٨٨ K

٢٤٢ processes        CPU: ١٠%        Memory: ١٫١ GiB / ٢٫٤ GiB        Swap: ٣١٧٫٨ MiB
```

Log file :



```
Child process was terminated his PID is 13955 ,the time and date is Tue Nov 10 18:00:20 2020

Child process was terminated his PID is 13955 ,the time and date is Tue Nov 10 18:00:25 2020

```

```
afnan@afnan:~$ gnome-calculator &
afnan@afnan:~$ gnome-calendar &
afnan@afnan:~$
```