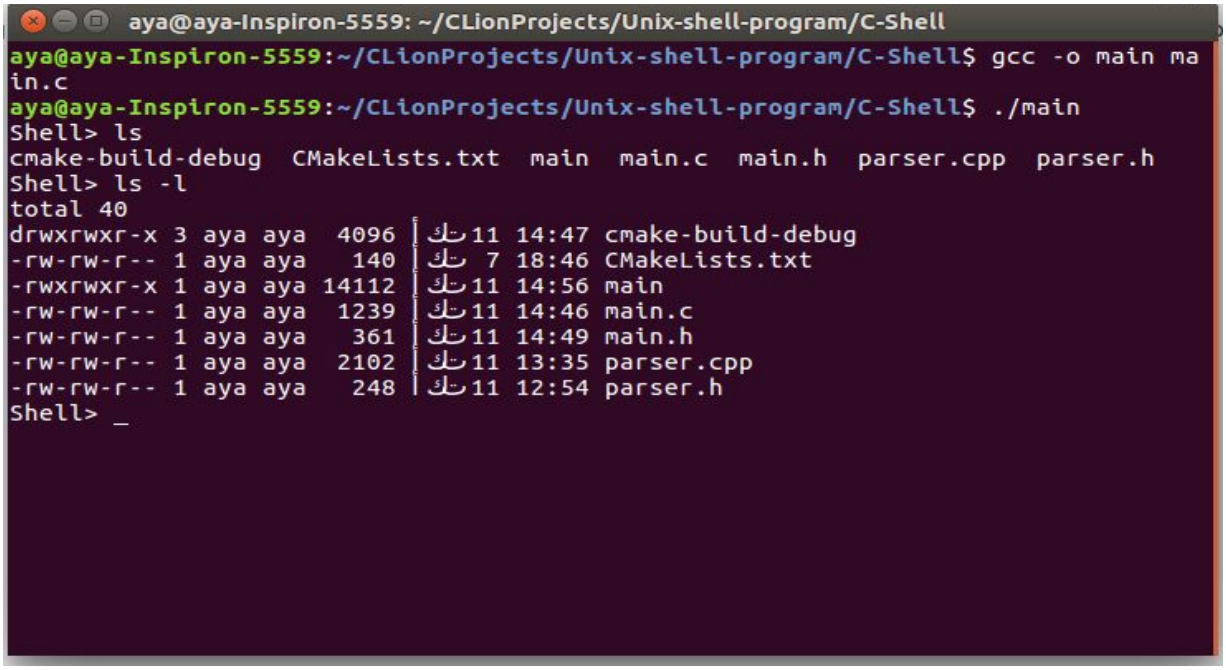
Operating system

C - Shell

Made by :- Aya Ashraf Saber Mohamed

ID:- 02



Introduction

It is the description of the implement a Unix shell program. A shell is simply a program that conveniently allows you to run other programs. Read up on your favorite shell to see what it does.

Implementation support the following:

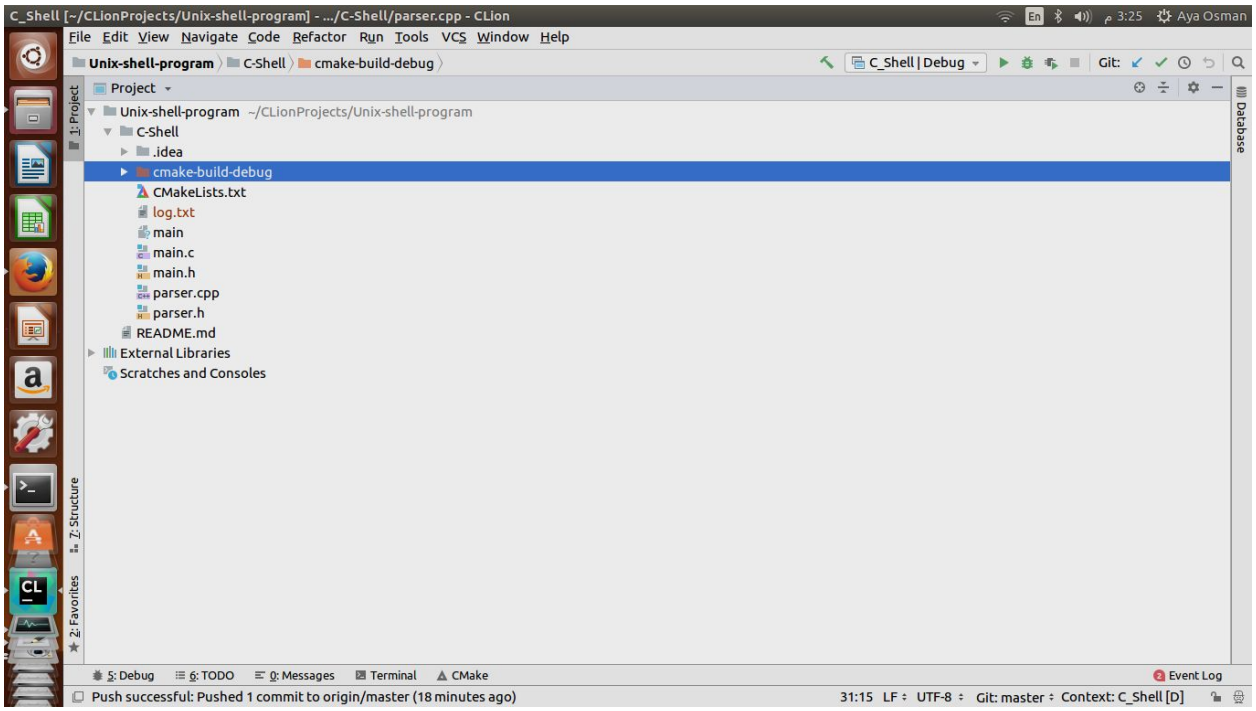
1. The internal shell command "exit" which terminates the shell.

2. A command with no arguments.

3. A command with arguments.

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

Code Hirechy:-

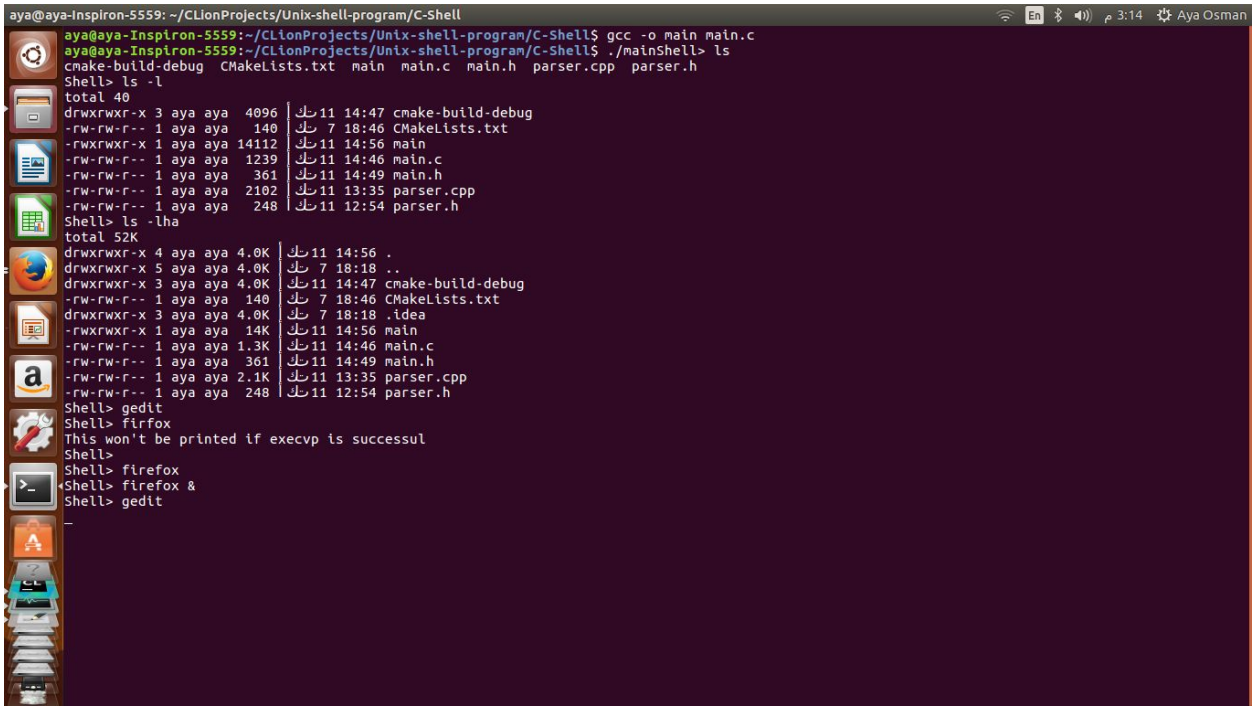


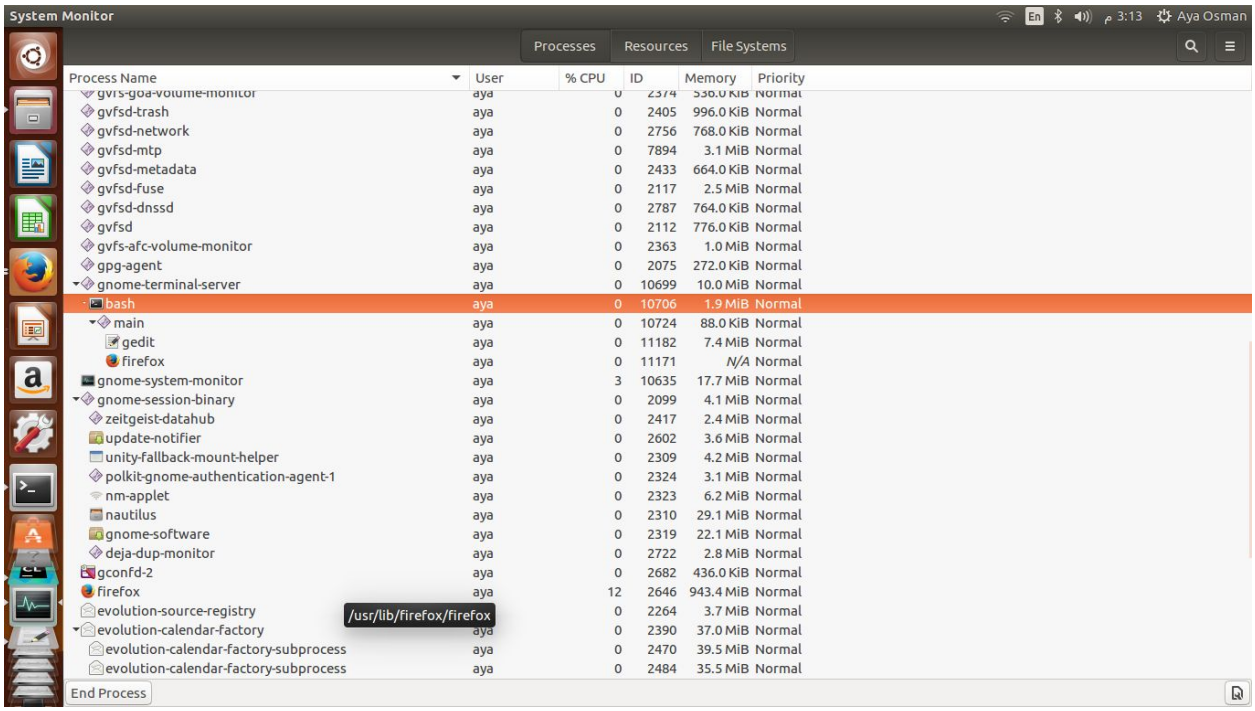
Function Description :-

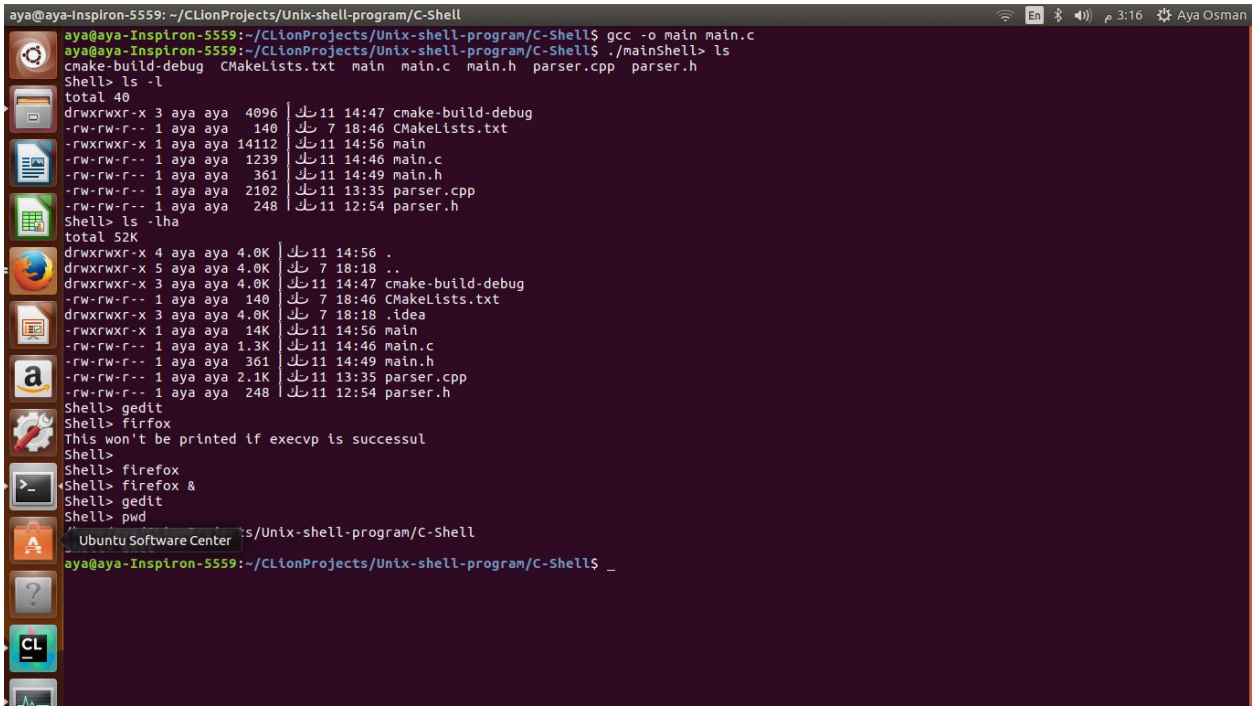
char* readInstruction();	This read line from the user.
char** get_arguments(**char*** line, **int*** backgroud);	This function take the inserted line and begin to split this line to words After trimming all tabs and new line. And predict the type of the command if it will run on the background or not
void signalHandler(**int** signum);	When signal is interrupted it print in file that the child process has finished.
void execute();	Execute function is responseble for the running of the system function it
int main()	The essential function which running the shell function.

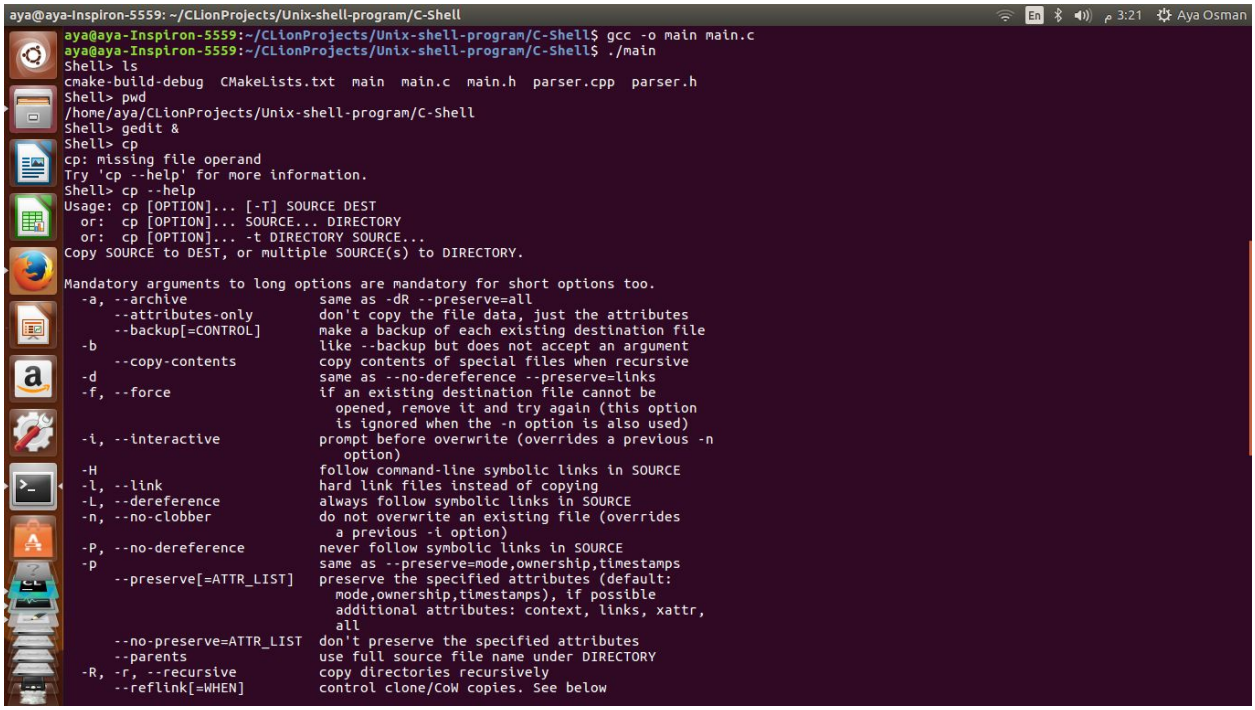
ScreenShot:-

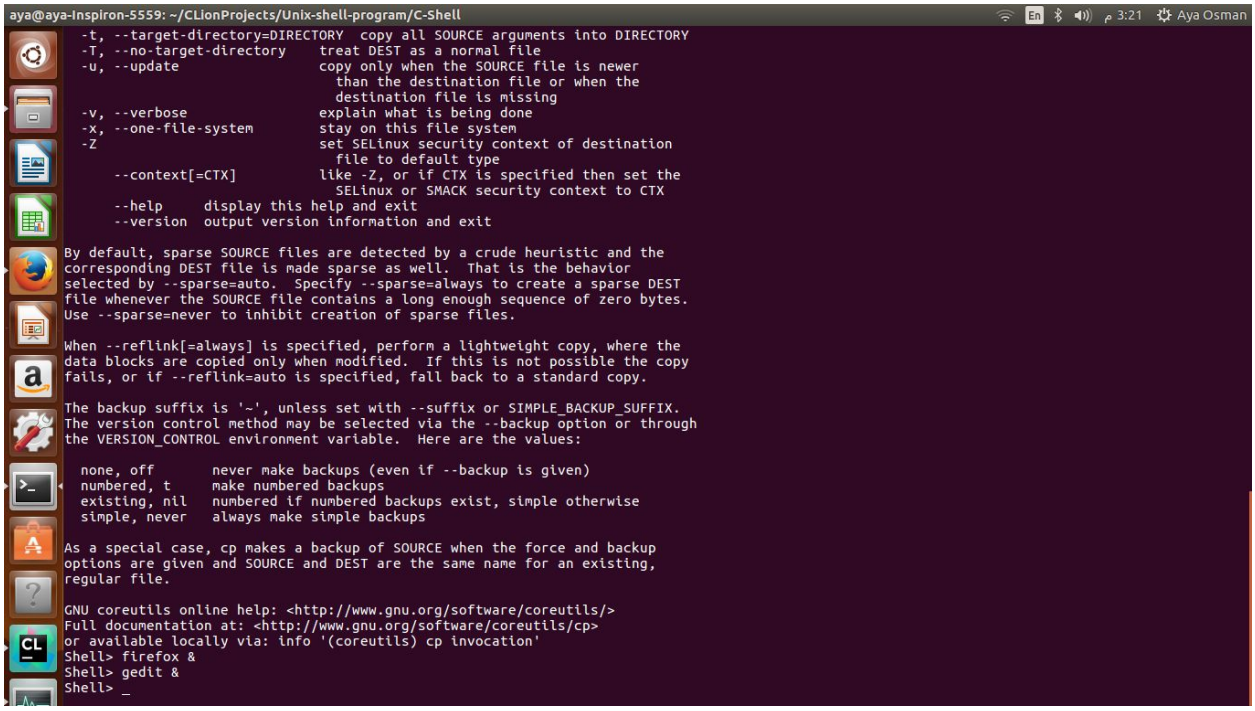
Running process :-



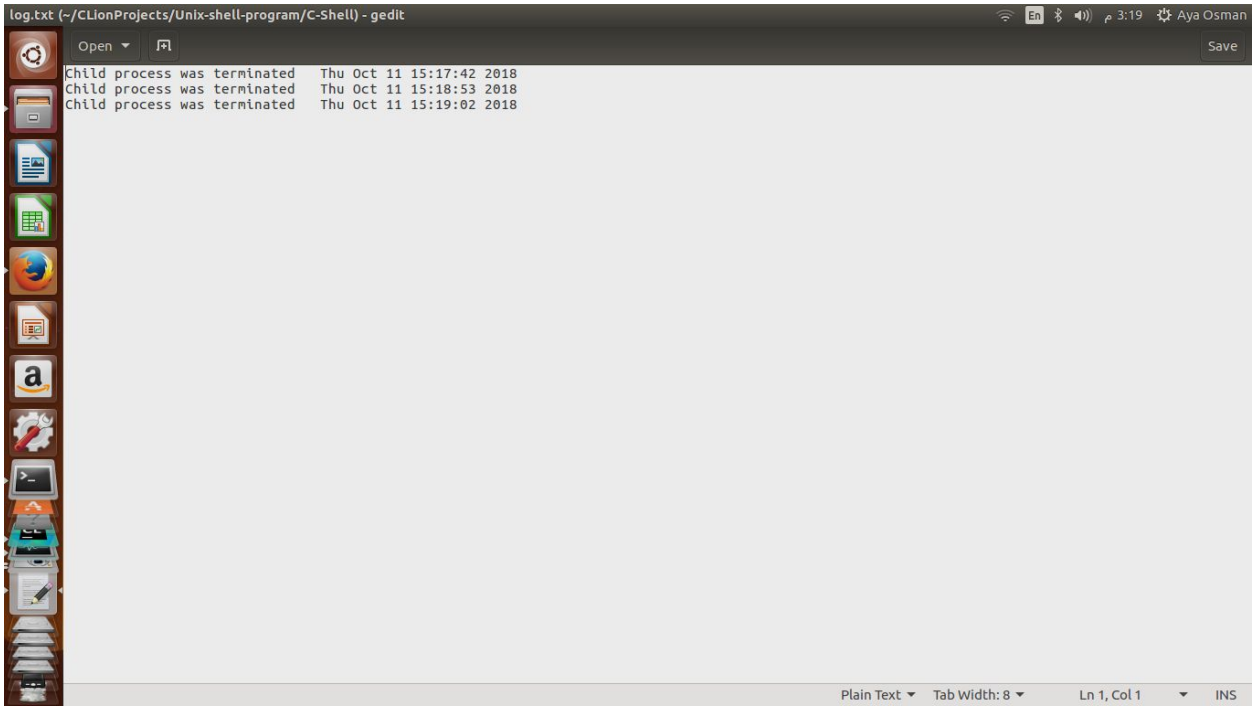

System Monitor
En
3:13
Aya Osman
Processes
Resources
File Systems
Process Name
User
% CPU
ID
Memory
Priority
gvfsd-trash aya 0 2405 996.0 KiB Normal
gvfsd-network aya 0 2756 768.0 KiB Normal
gvfsd-mtp aya 0 7894 3.1 MiB Normal
gvfsd-metadata aya 0 2433 664.0 KiB Normal
gvfsd-fuse aya 0 2117 2.5 MiB Normal
gvfsd-dnssd aya 0 2787 764.0 KiB Normal
gvfsd aya 0 2112 776.0 KiB Normal
gvfs-afc-volume-monitor aya 0 2363 1.0 MiB Normal
gpg-agent aya 0 2075 272.0 KiB Normal
gnome-terminal-server aya 0 10699 10.0 MiB Normal
bash aya 0 10706 1.9 MiB Normal
main aya 0 10724 88.0 KiB Normal
gedit aya 0 11182 7.4 MiB Normal
firefox aya 0 11171 N/A Normal
gnome-system-monitor aya 3 10635 17.7 MiB Normal
gnome-session-binary aya 0 2099 4.1 MiB Normal
zeitgeist-datahub aya 0 2417 2.4 MiB Normal
update-notifier aya 0 2602 3.6 MiB Normal
unity-fallback-mount-helper aya 0 2309 4.2 MiB Normal
polkit-gnome-authentication-agent-1 aya 0 2324 3.1 MiB Normal
nm-applet aya 0 2323 6.2 MiB Normal
nautilus aya 0 2310 29.1 MiB Normal
gnome-software aya 0 2319 22.1 MiB Normal
deja-dup-monitor aya 0 2722 2.8 MiB Normal
gconfd-2 aya 0 2682 436.0 KiB Normal
firefox aya 12 2646 943.4 MiB Normal
evolution-source-registry aya 0 2264 3.7 MiB Normal
/usr/lib/firefox/firefox
evolution-calendar-factory aya 0 2390 37.0 MiB Normal
evolution-calendar-factory-subprocess aya 0 2470 39.5 MiB Normal
evolution-calendar-factory-subprocess aya 0 2484 35.5 MiB Normal
End Process

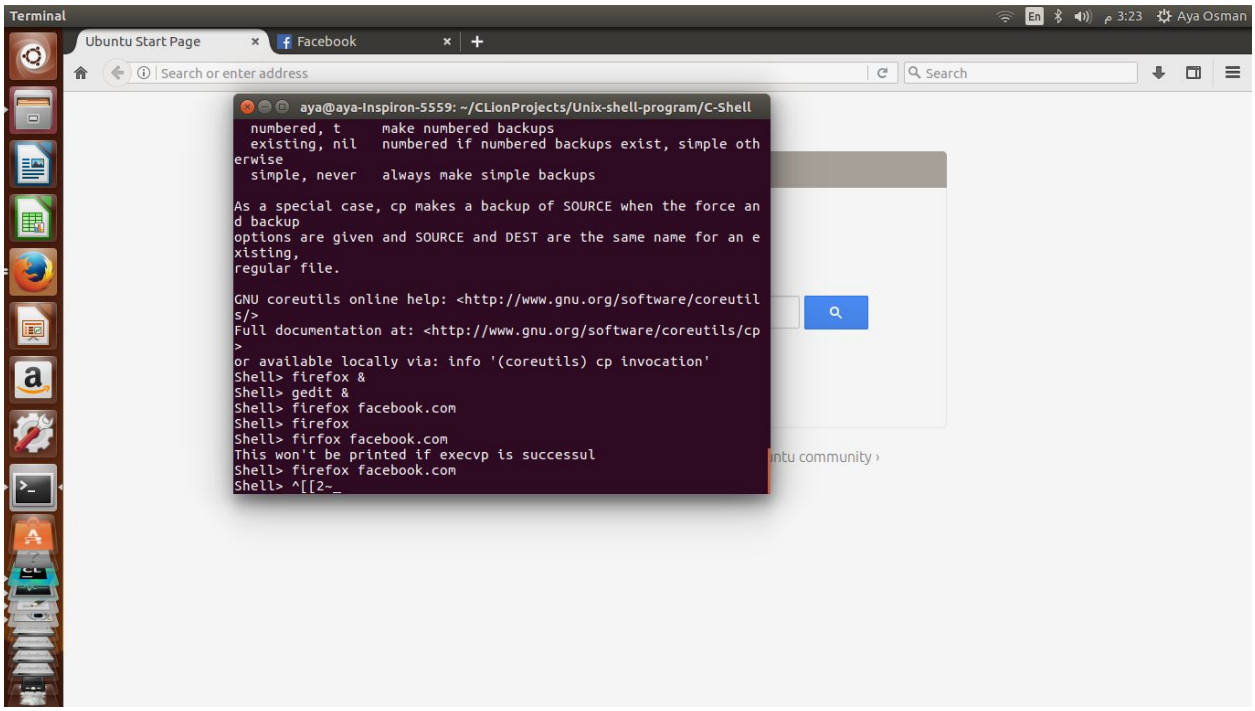

aya@aya-Inspiron-5559: ~/CLionProjects/Unix-shell-program/C-Shell
En
3:16
Aya Osman
aya@aya-Inspiron-5559:~/CLionProjects/Unix-shell-program/C-Shell$ gcc -o main main.c
aya@aya-Inspiron-5559:~/CLionProjects/Unix-shell-program/C-Shell$ ./mainShell> ls
cmake-build-debug CMakeLists.txt main main.c main.h parser.cpp parser.h
Shell> ls -l
total 40
drwxrwxr-x 3 aya aya 4096 ك11 14:47 cmake-build-debug
-rw-rw-r-- 1 aya aya 140 ك 7 18:46 CMakeLists.txt
-rwxrwxr-x 1 aya aya 14112 ك11 14:56 main
-rw-rw-r-- 1 aya aya 1239 ك11 14:46 main.c
-rw-rw-r-- 1 aya aya 361 ك11 14:49 main.h
-rw-rw-r-- 1 aya aya 2102 ك11 13:35 parser.cpp
-rw-rw-r-- 1 aya aya 248 ك11 12:54 parser.h
Shell> ls -lha
total 52K
drwxrwxr-x 4 aya aya 4.0K ك11 14:56 .
drwxrwxr-x 5 aya aya 4.0K ك 7 18:18 ..
drwxrwxr-x 3 aya aya 4.0K ك11 14:47 cmake-build-debug
-rw-rw-r-- 1 aya aya 140 ك 7 18:46 CMakeLists.txt
drwxrwxr-x 3 aya aya 4.0K ك 7 18:18 .idea
-rwxrwxr-x 1 aya aya 14K ك11 14:56 main
-rw-rw-r-- 1 aya aya 1.3K ك11 14:46 main.c
-rw-rw-r-- 1 aya aya 361 ك11 14:49 main.h
-rw-rw-r-- 1 aya aya 2.1K ك11 13:35 parser.cpp
-rw-rw-r-- 1 aya aya 248 ك11 12:54 parser.h
Shell> gedit
Shell> firfox
This won't be printed if execvp is successul
Shell>
Shell> firefox
Shell> firefox &
Shell> gedit
Shell> pwd
s/Unix-shell-program/C-Shell
Ubuntu Software Center
aya@aya-Inspiron-5559:~/CLionProjects/Unix-shell-program/C-Shell$ _


aya@aya-Inspiron-5559: ~/CLionProjects/Unix-shell-program/C-Shell
En 3:21 Aya Osman
aya@aya-Inspiron-5559:~/CLionProjects/Unix-shell-program/C-Shell$ gcc -o main main.c
aya@aya-Inspiron-5559:~/CLionProjects/Unix-shell-program/C-Shell$ ./main
Shell> ls
cmake-build-debug CMakeLists.txt main main.c main.h parser.cpp parser.h
Shell> pwd
/home/aya/CLionProjects/Unix-shell-program/C-Shell
Shell> gedit &
Shell> cp
cp: missing file operand
Try 'cp --help' for more information.
Shell> cp --help
Usage: cp [OPTION]... [-T] SOURCE DEST
or: cp [OPTION]... SOURCE... DIRECTORY
or: cp [OPTION]... -t DIRECTORY SOURCE...
Copy SOURCE to DEST, or multiple SOURCE(s) to DIRECTORY.
Mandatory arguments to long options are mandatory for short options too.
-a, --archive same as -dR --preserve=all
--attributes-only don't copy the file data, just the attributes
--backup[=CONTROL] make a backup of each existing destination file
-b like --backup but does not accept an argument
--copy-contents copy contents of special files when recursive
-d same as --no-dereference --preserve=links
-f, --force if an existing destination file cannot be
opened, remove it and try again (this option
is ignored when the -n option is also used)
-i, --interactive prompt before overwrite (overrides a previous -n
option)
-H follow command-line symbolic links in SOURCE
-l, --link hard link files instead of copying
-L, --dereference always follow symbolic links in SOURCE
-n, --no-clobber do not overwrite an existing file (overrides
a previous -i option)
-P, --no-dereference never follow symbolic links in SOURCE
-p same as --preserve=mode,ownership,timestamps
--preserve[=ATTR_LIST] preserve the specified attributes (default:
mode,ownership,timestamps), if possible
additional attributes: context, links, xattr,
all
--no-preserve=ATTR_LIST don't preserve the specified attributes
--parents use full source file name under DIRECTORY
-R, -r, --recursive copy directories recursively
--reflink[=WHEN] control clone/CoW copies. See below


aya@aya-Inspiron-5559: ~/CLionProjects/Unix-shell-program/C-Shell
En 3:21 Aya Osman
-t, --target-directory=DIRECTORY copy all SOURCE arguments into DIRECTORY
-T, --no-target-directory treat DEST as a normal file
-u, --update copy only when the SOURCE file is newer
than the destination file or when the
destination file is missing
-v, --verbose explain what is being done
-x, --one-file-system stay on this file system
-Z set SELinux security context of destination
file to default type
--context[=CTX] like -Z, or if CTX is specified then set the
SELinux or SMACK security context to CTX
--help display this help and exit
--version output version information and exit
By default, sparse SOURCE files are detected by a crude heuristic and the
corresponding DEST file is made sparse as well. That is the behavior
selected by --sparse=auto. Specify --sparse=always to create a sparse DEST
file whenever the SOURCE file contains a long enough sequence of zero bytes.
Use --sparse=never to inhibit creation of sparse files.
When --reflink[=always] is specified, perform a lightweight copy, where the
data blocks are copied only when modified. If this is not possible the copy
fails, or if --reflink=auto is specified, fall back to a standard copy.
The backup suffix is '~', unless set with --suffix or SIMPLE_BACKUP_SUFFIX.
The version control method may be selected via the --backup option or through
the VERSION_CONTROL environment variable. Here are the values:
none, off never make backups (even if --backup is given)
numbered, t make numbered backups
existing, nil numbered if numbered backups exist, simple otherwise
simple, never always make simple backups
As a special case, cp makes a backup of SOURCE when the force and backup
options are given and SOURCE and DEST are the same name for an existing,
regular file.
GNU coreutils online help: <http://www.gnu.org/software/coreutils/>
Full documentation at: <http://www.gnu.org/software/coreutils/cp>
or available locally via: info '(coreutils) cp invocation'
Shell> firefox &
Shell> gedit &
Shell> _

Log File





Reference:

https://www.geeksforgeeks.org/signals-c-language/

https://brennan.io/2015/01/16/write-a-shell-in-c/

http://alumni.cs.ucr.edu/~kishore/cs153/signals.c

https://indradhanush.github.io/blog/writing-a-unix-shell-part-2/

http://webhotel4.ruc.dk/~keld/teaching/CAN_e14/Readings/How%20to%20Compile%20and%20Run%20a%20C%20Program%20on%20Ubuntu%20Linux.pdf