

Operating System

Lab1

Name : Mahmoud Ebrahim Elsayed Mahmoud Manfy

Id : 57



Content:

- Lab description
- Basic idea
- Main functions
- Sample runs
- screenshots for the processes hierarchy in System Monitor.

Lab description

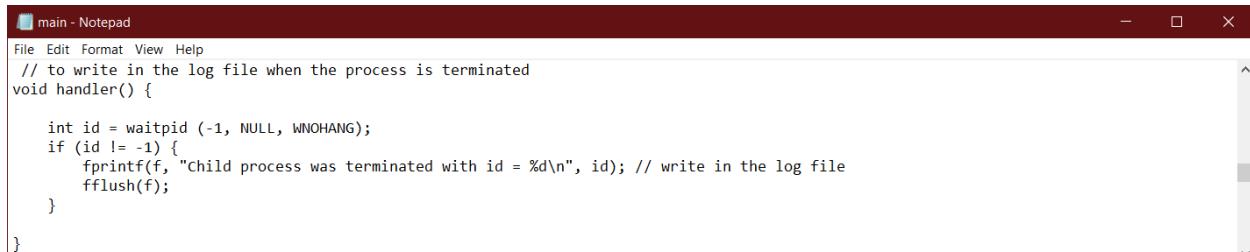
It is required to implement a Unix shell program. A shell is simply a program that conveniently allows you to run other programs. The shell program should support:

1. The internal shell command “exit” which terminates the shell.
2. A command with no arguments like: ls, cp, rm ...etc.
3. A command with arguments like: ls -l.
4. A command, with or without arguments, executed in the background using &.

Basic idea

- The infinite loop which corresponding the parent process and terminate when the user “exit” command.
- Each iteration has main five main steps:
 1. Takes the input from the user.
 2. Parsing the input and split it. If the input command starts with cd or the input command is empty line, the program will return to step number one.
 3. The program creates a new child process from the parent process using fork() function.
 4. If the process is the child process, the program will send the inputs to execvp() function.
 5. If the process is the parent process and the command isn't terminated with character '&', the program will wait the child to terminate first to continue the execution.

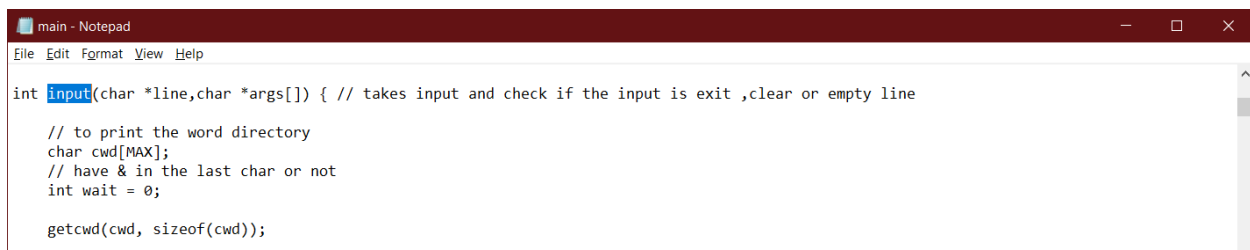
Main functions



```
main - Notepad
File Edit Format View Help
// to write in the log file when the process is terminated
void handler() {

    int id = waitpid (-1, NULL, WNOHANG);
    if (id != -1) {
        fprintf(f, "Child process was terminated with id = %d\n", id); // write in the log file
        fflush(f);
    }
}
```

- When the child process is terminated, it sends SIGCHLD to the parent process and this function handles this action.
- The function will check if there is a zombie process.
- If the zombie process is found, the function will clean up the zombie process and receive its id and print the received id in the log file.



```
main - Notepad
File Edit Format View Help
int input(char *line, char *args[]) { // takes input and check if the input is exit ,clear or empty line

    // to print the word directory
    char cwd[MAX];
    // have & in the last char or not
    int wait = 0;

    getcwd(cwd, sizeof(cwd));
}
```

- Takes the input from the user and splits it.
- It will return -1 if the input is 'exit'.
- It will return 1 if the input ended with '&'.
- It will return 2 if the input is an empty line or command cd.
- It will return 0 otherwise.
- The input function prints the absolute path of where it is executing.
- It will handle the command cd and execute it.

```
main - Notepad
File Edit Format View Help
int main() {

    char line[MAX],*args[MAX/2 + 1];

    f = fopen ("data.log", "w");

    while (1) {

        int in = input(&line,&args);

        if (in == -1) {
            break;
        } else if (in > 1) {
            continue;
        }

        signal(SIGCHLD, handler); // when the command is finished

        pid_t pid = fork();

        if (pid == -1) {

            printf("Failed forking child..\n");
            return 1;

        } if (pid == 0) {

            int error = execvp(args[0], args);
            if (error == -1) {
                printf("%s: command not found\n",line); // if there is error
            }
            exit(0);

        } else {
            if (!in) {
                waitpid (pid, NULL, 0); // wait the current child process
            }
        }
    }
}
```

- Has the main infinite loop which the program work around.
- Call fork() function which divides the parent process to two identical processes.
- Call execvp() which have the path of the executable files.

Sample runs

From the terminal:

```
manfi@admin: ~/CLionProjects/Lab1
manfi@admin:~/CLionProjects/Lab1$ ./a.out
/home/manfi/CLionProjects/Lab1> ls
a.out  cmake-build-debug  CMakeLists.txt  data.log  logs.txt  main.c
/home/manfi/CLionProjects/Lab1> lss
lss: command not found
/home/manfi/CLionProjects/Lab1> mkdir file
/home/manfi/CLionProjects/Lab1> ls
a.out  cmake-build-debug  CMakeLists.txt  data.log  file  logs.txt  main.c
/home/manfi/CLionProjects/Lab1> rm -r file
/home/manfi/CLionProjects/Lab1> ls
a.out  cmake-build-debug  CMakeLists.txt  data.log  logs.txt  main.c
/home/manfi/CLionProjects/Lab1> ls -l
total 36
-rwxrwxr-x 1 manfi manfi 17624 Nov 10 17:57 a.out
drwxrwxr-x 3 manfi manfi 4096 Nov 10 17:43 cmake-build-debug
-rw-rw-r-- 1 manfi manfi 107 Nov 8 15:59 CMakeLists.txt
-rw-rw-r-- 1 manfi manfi 0 Nov 10 17:57 data.log
-rw-rw-r-- 1 manfi manfi 516 Nov 9 20:15 logs.txt
-rw-rw-r-- 1 manfi manfi 2778 Nov 10 17:52 main.c
/home/manfi/CLionProjects/Lab1> echo "Mahmoud Ebrahim Elsayed Mahmoud Manfi"
"Mahmoud Ebrahim Elsayed Mahmoud Manfi"
/home/manfi/CLionProjects/Lab1> cd /
/> firefox
/> zoom
/> zoom &
/> which ls
/usr/bin/ls
/> cd /home/manfi
/home/manfi> ls
bin  CLionProjects  Desktop  Documents  Downloads  IdeaProjects  Music  Pictures  Public  PycharmProjects  snap  Templates  Videos
/home/manfi> exit
manfi@admin:~/CLionProjects/Lab1$
```

The log file:

```
data.log
~/CLionProjects/Lab1/cmake-build-debu
1 Child process was terminated with id = 42878
2 Child process was terminated with id = 42906
3 Child process was terminated with id = 42924
```

screenshots for the processes hierarchy in System Monitor

The screenshot displays the System Monitor application with the 'Processes' tab selected. The process list is organized into a hierarchy, showing parent processes and their children. The terminal window in the foreground shows the execution of a C program named 'main.c' which includes a 'main' function. The program uses 'getline' to read input from the user and prints it back. The terminal output shows the program running successfully, with some warnings about incompatible pointer types.

Process Name	User	Stat	% CPU	ID	Memory	Disk read tot	Disk write tot	Disk read	Disk write	Priority
gnome-shell	manfi	Slee	0	2717	772.0 KIB	28.0 KIB	N/A	N/A	N/A	Normal
gsd-screensaver-proxy	manfi	Slee	0	2716	632.0 KIB	N/A	N/A	N/A	N/A	Normal
gsd-rfkill	manfi	Slee	0	2715	1.4 MiB	44.0 KIB	N/A	N/A	N/A	Normal
gsd-print-notifications	manfi	Slee	0	2856	1.9 MiB	N/A	N/A	N/A	N/A	Normal
gsd-power	manfi	Slee	0	2712	10.5 MiB	100.0 KIB	N/A	N/A	N/A	Normal
gsd-media-keys	manfi	Slee	0	2711	11.0 MiB	340.0 KIB	392.0 KIB	N/A	N/A	Normal
gsd-keyboard	manfi	Slee	0	2709	9.8 MiB	4.0 KIB	N/A	N/A	N/A	Normal
gsd-housekeeping	manfi	Slee	0	2707	820.0 KIB	528.0 KIB	N/A	N/A	N/A	Normal
gsd-datetime	manfi	Slee	0	2705	1.9 MiB	N/A	N/A	N/A	N/A	Normal
gsd-color	manfi	Run	0	2704	10.2 MiB	8.0 KIB	N/A	N/A	N/A	Normal
gsd-a11y-settings	manfi	Slee	0	2703	604.0 KIB	N/A	N/A	N/A	N/A	Normal
goa-identity-service	manfi	Slee	0	2370	1.1 MiB	112.0 KIB	N/A	N/A	N/A	Normal
goa-daemon	manfi	Slee	0	2362	44.5 MiB	1.4 MiB	N/A	N/A	N/A	Normal
gnome-terminal-server	manfi	Run	0	42270	12.3 MiB	N/A	N/A	N/A	N/A	Normal
bash	manfi	Slee	0	42278	1.7 MiB	N/A	40.0 KIB	N/A	N/A	Normal
a.out	manfi	Slee	0	42297	64.0 KIB	N/A	4.0 KIB	N/A	N/A	Normal
zoom	manfi	Slee	0	42601	260.0 KIB	N/A	4.0 KIB	N/A	N/A	Normal
sh	manfi	Slee	0	42603	72.0 KIB	N/A	N/A	N/A	N/A	Normal
zoom	manfi	Run	0	42604	54.5 MiB	N/A	76.0 KIB	N/A	N/A	Normal
firefox	manfi	Run	0	42335	141.1 MiB	972.0 KIB	8.6 MiB	N/A	N/A	Normal
WebExtension	manfi	Slee	0	42454	45.2 MiB	N/A	N/A	N/A	N/A	Normal
Web Content	manfi	Run	0	42395	73.2 MiB	N/A	N/A	N/A	N/A	Normal
Web Content	manfi	Slee	0	42524	17.5 MiB	N/A	N/A	N/A	N/A	Normal
Privileged Con	manfi	Slee	0	42508	36.2 MiB	N/A	N/A	N/A	N/A	Normal
chrome-gnom	manfi	Slee	0	42507	14.1 MiB	N/A	N/A	N/A	N/A	Normal
gnome-shell-calendar-ser	manfi	Slee	0	2612	5.2 MiB	5.1 MiB	N/A	N/A	N/A	Normal
gnome-shell	manfi	Run	1	2555	246.8 MiB	937.8 MiB	36.2 MiB	N/A	N/A	Normal
mailspring	manfi	Slee	0	15880	56.0 MiB	41.3 MiB	1.2 MiB	N/A	N/A	Normal
ibus-daemon	manfi	Slee	0	2583	1.6 MiB	148.0 KIB	4.0 KIB	N/A	N/A	Normal
gnome-system-monitor	manfi	Run	0	28131	16.3 MiB	1.0 KIB	4.0 KIB	N/A	N/A	Normal
Discord--no-sandbox	manfi	Run	0	40312	41.6 MiB	N/A	88.0 KIB	N/A	N/A	Normal

```
manfi@admin: ~/CLionProjects/Lab1
manfi@admin:~/CLionProjects/Lab1$ cd CLionProjects/
manfi@admin:~/CLionProjects$ cd Lab1/
manfi@admin:~/CLionProjects/Lab1$ gcc main.c
main.c: In function 'main':
main.c:84:24: warning: passing argument 1 of 'input' from incompatible pointer type [-Wincompatible-pointer-types]
      int in = input(&line,&args);
                        ^~~~~~
                        |
                        | char (*)(sizetype)MAX]
main.c:13:17: note: expected 'char **' but argument is of type 'char (*)(sizetype)MAX]'
      int input(char *line,char *args[]) { // takes input and check if the input is exit ,clear or empty
      line
      |
      | char * (*)(sizetype)((unsigned int)MAX / 2 + 1)}
main.c:13:28: note: expected 'char **' but argument is of type 'char * (*)(sizetype)((unsigned int)MAX / 2 + 1)}'
      int input(char *line,char *args[]) { // takes input and check if the input is exit ,clear or empty
      line
      |
      |
manfi@admin:~/CLionProjects/Lab1$ ./a.out
/home/nanfi/CLionProjects/Lab1>> firefox &
/home/nanfi/CLionProjects/Lab1>> zoom &
/home/nanfi/CLionProjects/Lab1>> ls
a.out cmake-build-debug CMakeLists.txt data.log logs.txt main.c
/home/nanfi/CLionProjects/Lab1>> cd //
```