

Lab1 OS

Description of overall organization:

It is a simple implementation of a shell program in C programming language that can execute basic commands and shell built-ins, evaluate environment variables and also can run background processes to allow running other processes by using signal handler interrupt SIGCHLD and log the termination of child processes in a file

Description of major functions:

- **Execute_command():**
 - creates a new child process using the **fork** system call
 - executes all the given command using **execvp** system call except **cd**, **echo** and **export**
 - If foreground process parent will invoke **waitpid** system call
 - If background process will be interrupted by a SIGCHLD signal when child dies

- **execute_shell_bultin():**
 - handles the shell built-ins command **cd**, **echo** and **export**
 - **Case cd we use chdir() function**
 - **Case echo we remove the double quotes**
 - **Case export we use setenv() function**

- **Evaluate_expression():**
 - replaces the environment variables in the given command with their corresponding values using the **getenv**
 - searches for the **\$** sign in the command string and extracts the variable name following it.
 - it retrieves the value of the corresponding environment variable using **getenv** and concatenates it to the result string
 - If there are no environment variables in the command, it simply returns the original command string

- **Shell():**
 - takes the user input as a command string
 - parses the command and its parameters

Sample runs:

```
ls
```

```
a.out cmake-build-debug CMakeLists.txt main main.c myFile.txt
```

```
mkdir test
```

```
ls
```

```
a.out cmake-build-debug CMakeLists.txt main main.c myFile.txt test
```

```
ls -a -l -h
```

```
total 72K
```

```
drwxr-xr-x  5 islam islam 4.0K 04:38 9 مار .
drwxr-x--- 19 islam islam 4.0K 03:22 9 مار ..
-rwxrwxr-x  1 islam islam 16K 21:11 7 مار a.out
drwxrwxr-x  4 islam islam 4.0K 20:10 7 مار cmake-build-debug
-rw-rw-r--  1 islam islam 114 20:10 7 مار CMakeLists.txt
drwxrwxr-x  2 islam islam 4.0K 04:36 9 مار .idea
-rwxrwxr-x  1 islam islam 17K 04:38 9 مار main
-rw-rw-r--  1 islam islam 4.8K 02:49 9 مار main.c
-rw-rw-r--  1 islam islam 105 01:55 9 مار myFile.txt
drwxrwxr-x  2 islam islam 4.0K 04:38 9 مار test
```

```
export x="-a -l -h"
```

```
ls $x
```

```
total 72K
```

```
drwxr-xr-x  5 islam islam 4.0K 04:38 9 مار .
drwxr-x--- 19 islam islam 4.0K 03:22 9 مار ..
-rwxrwxr-x  1 islam islam 16K 21:11 7 مار a.out
drwxrwxr-x  4 islam islam 4.0K 20:10 7 مار cmake-build-debug
-rw-rw-r--  1 islam islam 114 20:10 7 مار CMakeLists.txt
drwxrwxr-x  2 islam islam 4.0K 04:36 9 مار .idea
-rwxrwxr-x  1 islam islam 17K 04:38 9 مار main
-rw-rw-r--  1 islam islam 4.8K 02:49 9 مار main.c
-rw-rw-r--  1 islam islam 105 01:55 9 مار myFile.txt
drwxrwxr-x  2 islam islam 4.0K 04:38 9 مار test
```

```
gedit
echo "wow"
ls
pwd
|
```

After closing gedit

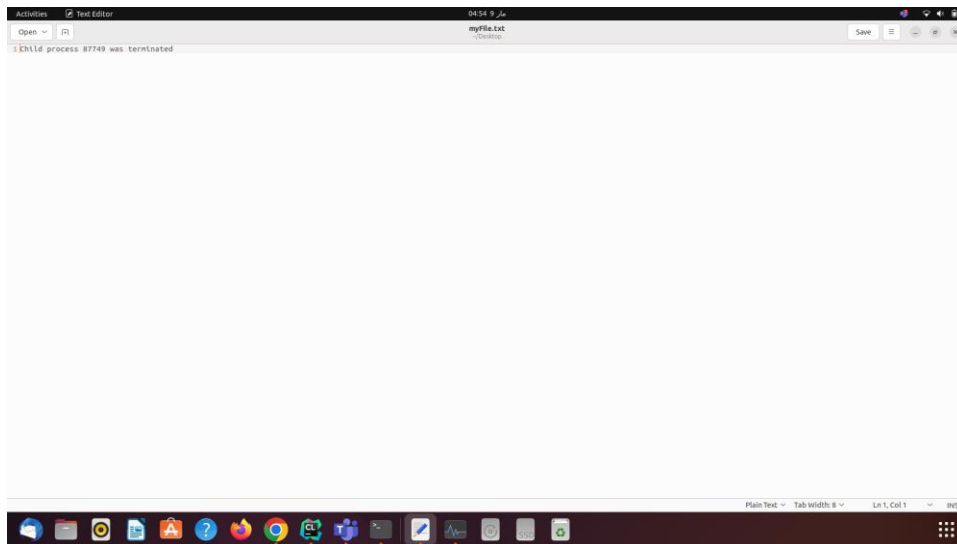
```
gedit
echo "wow"
ls
pwd
wow
a.out  cmake-build-debug  CMakeLists.txt  main  main.c  myFile.txt  test
/home/islam/Desktop
|
```

Background process

```
gedit &
ls
a.out  cmake-build-debug  CMakeLists.txt  main  main.c  myFile.txt  test
pwd
/home/islam/Desktop
```

After closing background process

```
gedit &
ls
a.out  cmake-build-debug  CMakeLists.txt  main  main.c  myFile.txt  test
pwd
/home/islam/Desktop
process 87749 terminated with return code: 0
|
```



Exiting the shell

```
hey  
Wrong command!  
exit  
  
Process finished with exit code 0  
|
```

ion Control ▶ Run ☰ TODO ⓘ Problems 📄 Term

Code snippets:

```

1  #include <stdio.h>
2  #include <unistd.h>
3  #include <string.h>
4  #include <stdlib.h>
5  #include <sys/wait.h>
6  #include <ctype.h>
7
8  #define Max_Size 100
9
10 int status;
11 int flag = 0; //flag to determine whether it is the first time to open a file or no
12
13 void execute_command(char* args[], int foreground) {
14     pid_t pid = fork();
15     if(pid == 0) { //child process
16         execvp( file: args[0], argv: args);
17         printf( format: "Wrong command!\n"); //if execution fails
18         exit( status: 0);
19     }
20     if(pid > 0 && foreground) //if parent process and foreground process
21         waitpid(pid, stat_loc: &status, options: 0);
22     else if(pid < 0)
23         printf( format: "failed to fork");
24 }

```

```

25
26 void execute_shell_bultin(char* args[]) { //case cd, echo, export
27     if(strcmp(args[0], "cd") == 0) { // if the command is cd
28         if(args[1] == NULL) //case cd
29             chdir( path: "/home");
30         else if(strcmp(args[1], "~") == 0) //case cd ~
31             chdir( path: ".");
32         else
33             chdir( path: args[1]); //remaining cases
34     }
35     else if(strcmp(args[0], "echo") == 0) { //if the command is echo
36         args[1]++;
37         char* LastChar = args[1]+strlen(s: args[1])-1;
38         *LastChar = '\0'; //eliminate the double quotes at the start and the end of the string
39         printf( format: "%s\n", args[1]);
40     }
41     else { //if the command is export
42         char* variable = strtok(s: args[1], delim: "="); //split the string and store the variable name
43         char* val = strtok(s: NULL, delim: "="); //store the variable value in val
44         if(*val == '"') { //if double quotes exists eliminate them
45             val++;
46             val[strlen(s: val)-1] = '\0';
47         }
48         setenv( name: variable, value: val, replace: 1); //set the name and the value of the environment variable and enable replacing

```

```

49     }
50 }
51
52 //function to replace the environment variables if exists by their value
53 void evaluate_expression(char command[]) {
54     char var[20] = "";
55     char result[100] = "";
56     char* val;
57     for (int i = 0; i < strlen(s: command); i++) {
58         if (command[i] == '$') {
59             // if it is '$' sign then get the variable name that follows it
60             int j = 0;
61             while (isalnum(command[i+1+j])) { // check if the next character is alphanumeric
62                 var[j] = command[i+1+j];
63                 j++;
64             }
65             var[j] = '\0'; // terminate the string
66             char* variablePointer = &var;
67             val = getenv(name: variablePointer);
68
69             // concatenate the variable value to the result string
70             strcat(dest: result, src: val);
71
72             i += j; // skip the remaining characters
73         } else {
74             // if it's not a '$' sign then copy the character to the result string
75             char temp[2];
76             temp[0] = command[i];
77             temp[1] = '\0'; // add a null terminator to make it a valid string
78             strcat(dest: result, src: temp); // concatenate the character to the result string
79         }
80     }
81     strcpy(dest: command, src: result);
82 }
83
84 void shell() {
85     char command[Max_Size];
86     while(1) {
87         fgets(s: command, n: Max_Size, stream: stdin); //read line
88         command[strcspn(s: command, reject: "\n")] = 0; //remove newline character at the end
89         int foreground = 1; //flag to determine foreground or not
90         if(strcmp(command, "exit") == 0) //if exit command
91             exit(status: 0);
92         if(command[strlen(s: command) - 1] == '&') //if not foreground command
93             foreground = 0;
94         evaluate_expression(command);
95         char* pointer = strtok(s: command, delim: " "); //split the command
96         char* args[Max_Size]; //will hold the command name and its parameters

```

```

97     args[0] = pointer; //command name in index 0
98     if(strcmp(args[0],"cd") == 0 || strcmp(args[0],"echo") == 0 || strcmp(args[0],"export") == 0) {
99         args[1] = strtok(s:NULL, delim:""); //parameter of the command in index 1
100         execute_shell_builtin(args);
101         continue;
102     }
103     int i;
104     for(i = 1; pointer != NULL; i++) { //if still there is a command parameter
105         pointer = strtok(s:NULL, delim:" ");
106         args[i] = pointer; //command parameter is in index i
107     }
108     execute_command(args, foreground);
109 }
110 }
111 //signal handler function
112 void on_child_exit_handler() {
113     pid_t pid;
114     while ((pid = waitpid(pid: -1, stat_loc: &status, options: WNOHANG)) > 0) { //if any child process termin
115         printf(format: "process %d terminated with return code: %d\n", pid, status);
116         FILE* myFile;
117         // log the termination of the child in a file
118         if(flag) myFile = fopen(filename: "myFile.txt", modes: "a"); //if not first time to open the file
119         else {myFile = fopen(filename: "myFile.txt", modes: "w"); flag = 1;} //if it is the first time to
120         fprintf(stream: myFile, format: "Child process %d was terminated\n", pid);
121         fclose(stream: myFile);
122     }
123 }
124
125 int main() {
126     signal(sig: SIGCHLD, handler: on_child_exit_handler); //initialize signal handler
127     shell(); // initialize the shell
128     return 0;
129 }

```

System monitor:

06/29 9 مارس										
Activities System Monitor										
Processes Resources File Systems										
Q										
Process Name	User	% CPU	ID	Memory	Disk read tot:	Disk write tot:	Disk read	Disk write	Priority	
gjs	islam	0.00	2055	5.3 MB	N/A	N/A	N/A	N/A	N/A	Normal
gjs	islam	0.00	2319	5.3 MB	N/A	N/A	N/A	N/A	N/A	Normal
gnome-session-binary	islam	0.00	1913	3.1 MB	6.9 MB	4.1 kB	N/A	N/A	N/A	Normal
atspi-bus-launcher	islam	0.00	1971	675.8 kB	N/A	N/A	N/A	N/A	N/A	Normal
dbus-daemon	islam	0.00	1979	483.3 kB	N/A	N/A	N/A	N/A	N/A	Normal
evolution-alarm-notify	islam	0.00	2123	14.9 MB	3.2 MB	N/A	N/A	N/A	N/A	Normal
gsd-disk-utility-notify	islam	0.00	2125	1.4 MB	24.6 kB	N/A	N/A	N/A	N/A	Normal
update-notifier	islam	0.00	4104	8.5 MB	2.9 MB	46.0 MB	N/A	N/A	N/A	Normal
gnome-session-ctl	islam	0.00	1901	442.4 kB	20.5 kB	N/A	N/A	N/A	N/A	Normal
gnome-shell	islam	2.16	1972	247.7 MB	255.0 MB	118.4 MB	N/A	N/A	N/A	Normal
chrome	islam	0.00	18250	114.8 MB	10.1 MB	2.3 GB	N/A	N/A	N/A	Normal
cat	islam	0.00	18257	90.1 kB	N/A	N/A	N/A	N/A	N/A	Normal
cat	islam	0.00	18256	94.2 kB	N/A	N/A	N/A	N/A	N/A	Normal
chrome-type=utility-utility-sub-type=audio.mojom.AudioService-lang=en-US-service=sandbox-type=none-crashpad-handler-pid=18259-enable-crash	islam	0.00	19247	11.9 MB	N/A	N/A	N/A	N/A	N/A	Normal
chrome-type=utility-utility-sub-type=network.mojom.NetworkService-lang=en-US-service=sandbox-type=none-crashpad-handler-pid=18259-enable-	islam	0.03	18297	25.8 MB	122.9 kB	349.7 MB	N/A	N/A	N/A	Normal
chrome-type=zygote-crashpad-handler-pid=18259-enable-crash-reporter=-change-stack-guard-on-fork=enable	islam	0.00	18268	11.5 MB	N/A	N/A	N/A	N/A	N/A	Normal
chrome-type=zygote-crashpad-handler-pid=18259-enable-crash-reporter=-change-stack-guard-on-fork=enable	islam	0.00	18272	11.6 MB	10.5 MB	162.8 MB	N/A	N/A	N/A	Normal
chrome-type=renderer-crashpad-handler-pid=18259-enable-crash-reporter=5441ea8d-d924-4133-8f6a-1902b4e9db2a,-change-stack-guard-on-fc	islam	0.00	58331	39.5 MB	N/A	323.6 kB	N/A	N/A	N/A	Normal
chrome-type=renderer-crashpad-handler-pid=18259-enable-crash-reporter=5441ea8d-d924-4133-8f6a-1902b4e9db2a,-change-stack-guard-on-fc	islam	0.00	85111	23.8 MB	N/A	569.3 kB	N/A	N/A	N/A	Normal
chrome-type=renderer-crashpad-handler-pid=18259-enable-crash-reporter=5441ea8d-d924-4133-8f6a-1902b4e9db2a,-change-stack-guard-on-fc	islam	0.00	85138	317.5 MB	N/A	9.9 MB	N/A	N/A	N/A	Normal
chrome-type=renderer-crashpad-handler-pid=18259-enable-crash-reporter=5441ea8d-d924-4133-8f6a-1902b4e9db2a,-change-stack-guard-on-fc	islam	0.00	85151	22.4 MB	N/A	151.6 kB	N/A	N/A	N/A	Normal
chrome-type=renderer-crashpad-handler-pid=18259-enable-crash-reporter=5441ea8d-d924-4133-8f6a-1902b4e9db2a,-change-stack-guard-on-fc	islam	0.00	85192	17.8 MB	N/A	N/A	N/A	N/A	N/A	Normal
chrome-type=renderer-crashpad-handler-pid=18259-enable-crash-reporter=5441ea8d-d924-4133-8f6a-1902b4e9db2a,-change-stack-guard-on-fc	islam	0.00	96871	16.0 MB	N/A	N/A	N/A	N/A	N/A	Normal
chrome-type=utility-utility-sub-type=storage.mojom.StorageService-lang=en-US-service=sandbox-type=utility-crashpad-handler-pid=18259-en	islam	0.00	18308	13.5 MB	8.2 kB	6.4 MB	N/A	N/A	N/A	Normal
nad_helper	islam	0.00	18269	426.0 kB	N/A	N/A	N/A	N/A	N/A	Normal
chrome-type=zygote-no-zygote-sandbox-crashpad-handler-pid=18259-enable-crash-reporter=-change-stack-guard-on-fork=enable	islam	0.00	18267	11.5 MB	N/A	N/A	N/A	N/A	N/A	Normal
chrome-type=gpu-process-crashpad-handler-pid=18259-enable-crash-reporter=-change-stack-guard-on-fork=enable-gpu-preferences=WAAAAA	islam	0.00	18296	94.1 MB	9.4 MB	6.4 MB	N/A	N/A	N/A	Normal
gjs	islam	0.00	54960	114.2 MB	145.4 kB	8.2 kB	N/A	N/A	N/A	Normal
gnome-system-monitor	islam	1.15	85321	31.9 MB	8.6 MB	8.2 kB	N/A	N/A	N/A	Normal
java	islam	0.14	11599	2.7 GB	296.7 MB	588.8 MB	N/A	N/A	N/A	Normal
fsnotifier	islam	0.00	11827	147.5 kB	12.3 kB	N/A	N/A	N/A	N/A	Normal
ion.clangd.main	islam	0.00	11887	11.6 MB	17.6 MB	89.4 MB	N/A	N/A	N/A	Normal
main	islam	0.00	101725	94.2 kB	N/A	N/A	N/A	N/A	N/A	Normal
firefox	islam	0.00	101732	159.5 MB	6.3 MB	11.4 MB	N/A	N/A	N/A	Normal
gedit	islam	0.00	102209	14.6 MB	90.1 kB	N/A	N/A	N/A	N/A	Normal
obs	islam	0.56	96233	382.3 MB	26.3 MB	1.3 MB	N/A	N/A	N/A	Normal
teams-for-linux-no-sandbox-enable-crashpad	islam	0.00	4171	55.0 MB	98.5 MB	176.3 MB	N/A	N/A	N/A	Normal

Firefox and gedit are childs for main