

Operating System Assignment 1

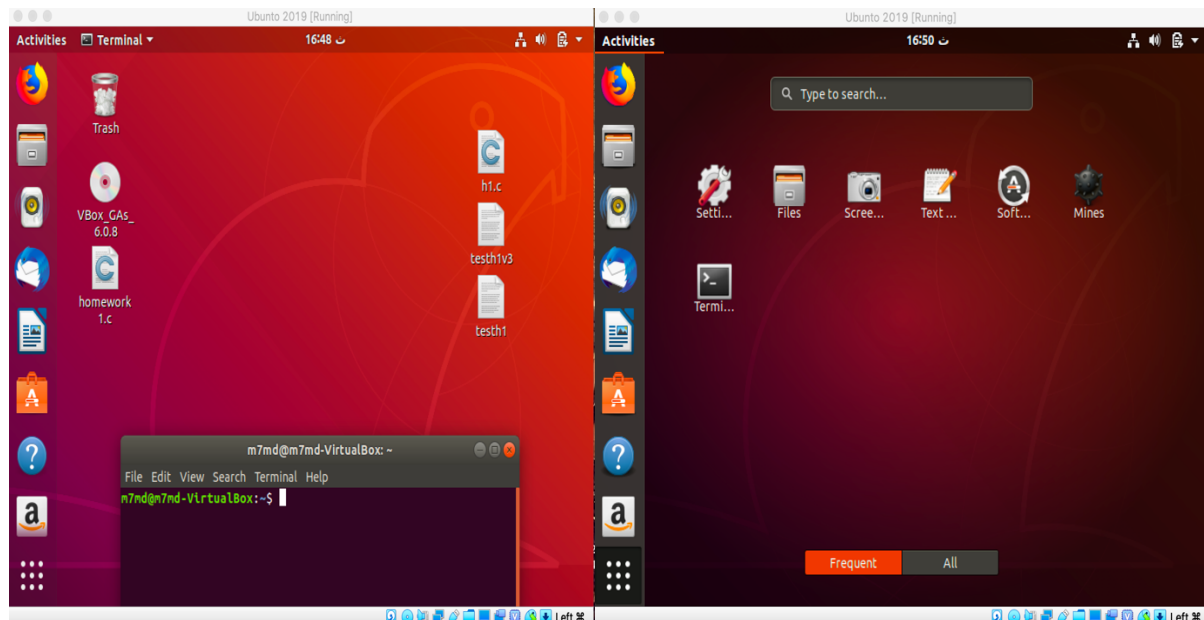
Mohammed Alaqil

Answer for Part 1 :

Firstly I chose the Virtual Box from Oracle for virtualization , it can be installed for mac and that is the most reason I chose Virtual Box in other hand Virtual Box is so popular and from a popular company as well , a lot of people try to teach how to use Virtual Box by videos and text explanation , the same thing for Linux distribution and the name of it is Ubuntu .

When I finished the installation of the app I had some problems in settings like how many memory should I put for the Linux distribution , how many processors need the distribution and It will not damage my computer generally I took a lot of time working with settings for the safety of my computer , I had problem with display it looks so small and I can't I solved it by increase the scale factor to 200% and looks better view but I can't say the problem it fully solved but it's better .

In the last when I want to write my code in C language and begin to work I have to learn some commands to use the terminal and start coding , there is some commands It couldn't work so I search about it and it was gcc library it has to be downloaded to work these command , after that I learned about C global variables , and how print statement work because it's different than java , finally it is a great experience and I will develop myself and my knowledge in this area .



Answer for Part 2 :

1)

i) Discuss the order of the output from the parent and children processes?

It's Print the the Parent first then after , print the first child then the second child

So it's in that order :

- 1- Parent
- 2- 1'st child
- 3- 2'nd child

ii) Why the value of the global variable isn't affected by the increment/decrement operations in other processes?

Because after the fork operation the value of the parent isn't shared with the another processes like every process it has their own value and it independent, so the answer it's every process has their own value and it isn't shared variable with another processes .

My Code :--

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

int v = 7;

int main () {

pid_t pid = fork();

if (pid == 0 ) {
v++;
printf("FirstChild=%d\n",v );
return 0 ;
} else if (pid > 0 ) {

pid = fork();

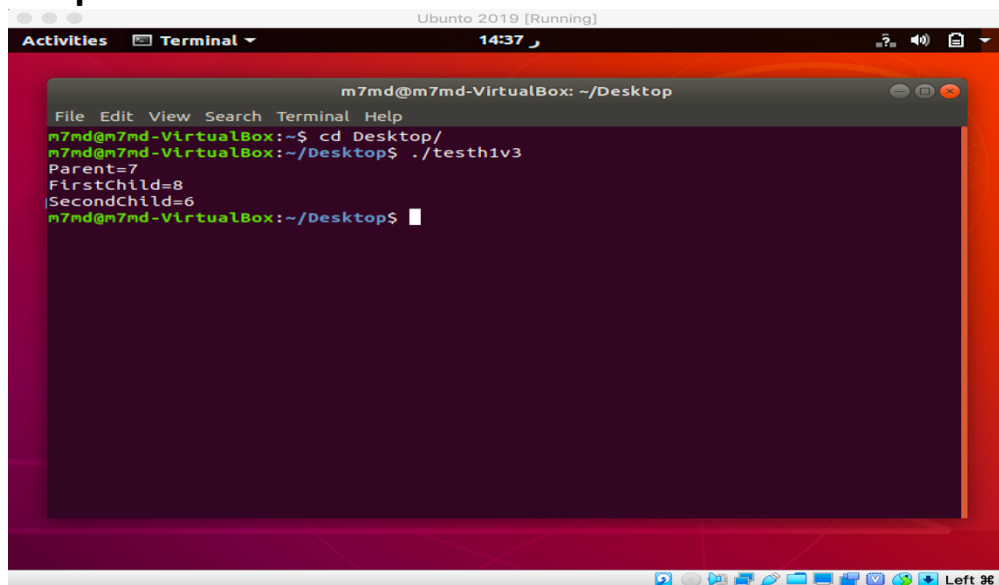
if (pid == 0) {
v--;
printf("SecondChild=%d\n",v );
return 0 ;
} else if ( pid > 0 ) {
//wait(NULL);
```

```
//wait(NULL) ;
printf("Parent=%d\n",v) ;
return 0 ;
}

}

}
```

Output :--



```
m7md@m7md-VirtualBox: ~/Desktop
File Edit View Search Terminal Help
m7md@m7md-VirtualBox:~$ cd Desktop/
m7md@m7md-VirtualBox:~/Desktop$ ./testh1v3
Parent=7
FirstChild=8
SecondChild=6
m7md@m7md-VirtualBox:~/Desktop$
```

2)

- 1- brk() to control the amount of memory allocated to the data segment of the process. These functions are typically called from a higher-level memory management library function such as malloc.
- 2- access() checks whether the calling process can access the file pathname. If pathname is a symbolic link, it is dereferenced.
- 3- fstat() is identical to stat(), except that the file to be stat-ed is specified by the file descriptor fd. stat() and fstatat() retrieve information about the file pointed to by pathname
- 4- mmap() creates a new mapping in the virtual address space of the calling process.
- 5- close() closes a file descriptor, so that it no longer refers to any file and may be reused.

6- read() attempts to read up to count bytes from file descriptor fd into the buffer starting at buf.

7- mprotect() changes the access protections for the calling process's memory pages .

8- arch_prctl() sets architecture-specific process or thread state.

9- The munmap() system call deletes the mappings for the specified address range, and causes further references to addresses within the range to generate invalid memory references.

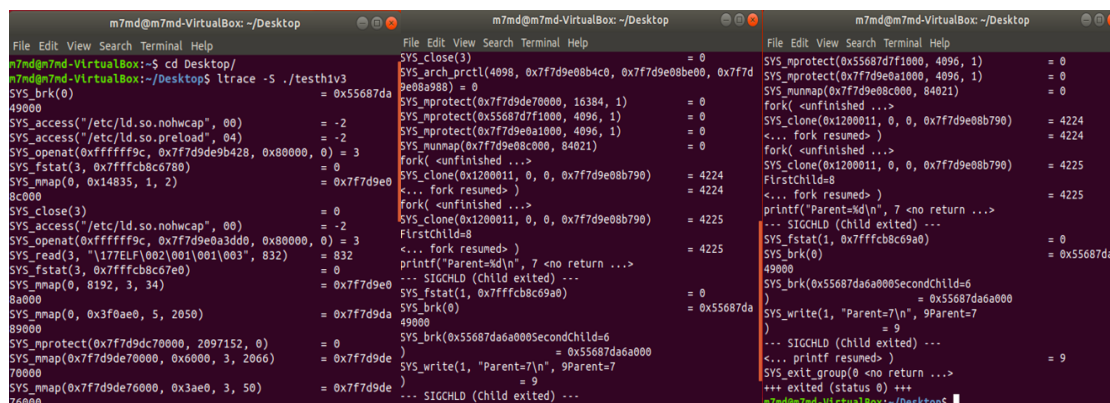
10- fork() creates a new process by duplicating the calling process. The new process is referred to as the child process. The calling process is referred to as the parent process.

11- clone() to implement threads: multiple threads of control in a program that run concurrently in a shared memory space.

12- write() writes up to count bytes from the buffer starting at buf to the file referred to by the file descriptor fd.

13- exit_group() This system call is equivalent to exit() except that it terminates not only the calling thread, but all threads in the calling process's thread group.

**** file descriptor :** s an abstract indicator (handle) used to access a file or other input/output resource



```
m7md@m7md-VirtualBox: ~/Desktop
File Edit View Search Terminal Help
m7md@m7md-VirtualBox:~$ cd Desktop/
m7md@m7md-VirtualBox:~/Desktop$ ltrace -S ./testhiv3
SYS_brk(0) = 0x55687da49000
SYS_access("/etc/ld.so.nohwcap", 00) = -2
SYS_access("/etc/ld.so.preload", 04) = -2
SYS_openat(0xfffff9c, 0x7f7d9de9b428, 0x80000, 0) = 3
SYS_fstat(3, 0x7ffcb8c6780) = 0
SYS_mmap(0, 0x14835, 1, 2) = 0x7f7d9e8c000
SYS_close(3) = 0
SYS_access("/etc/ld.so.nohwcap", 00) = -2
SYS_openat(0xfffff9c, 0x7f7d9e0a3dd0, 0x80000, 0) = 3
SYS_read(3, "\177ELF\001\001\003", 832) = 832
SYS_fstat(3, 0x7ffcb8c67e0) = 0
SYS_mmap(0, 8192, 3, 34) = 0x7f7d9e0a000
SYS_mmap(0, 0x3f0ae, 5, 2050) = 0x7f7d9da49000
SYS_mprotect(0x7f7d9dc70000, 2097152, 0) = 0
SYS_mmap(0x7f7d9de70000, 0x6000, 3, 2066) = 0x7f7d9de70000
SYS_mmap(0x7f7d9de70000, 0x3ae0, 3, 50) = 0x7f7d9de70000
SYS_close(3) = 0
SYS_arch_prctl(4096, 0x7f7d9e08b4c0, 0x7f7d9e08be00, 0x7f7d9e08a988) = 0
SYS_mprotect(0x7f7d9de70000, 16384, 1) = 0
SYS_mprotect(0x55687d7f1000, 4096, 1) = 0
SYS_mprotect(0x7f7d9e0a1000, 4096, 1) = 0
SYS_munmap(0x7f7d9e08c000, 84021) = 0
fork( <unfinished ...> = 0
fork( <unfinished ...> = 4224
SYS_clone(0x1200011, 0, 0, 0x7f7d9e08b790) = 4224
<... fork resumed> = 4224
fork( <unfinished ...> = 4225
SYS_clone(0x1200011, 0, 0, 0x7f7d9e08b790) = 4225
<... fork resumed> = 4225
printf("Parent=%d\n", 7 <no return ...>
--- SIGCHLD (child exited) ---
SYS_fstat(1, 0x7ffcb8c69a0) = 0
SYS_brk(0) = 0x55687da49000
SYS_brk(0x55687da6a000SecondChild=6) = 0x55687da6a000
SYS_write(1, "Parent=7\n", 9Parent=7) = 9
--- SIGCHLD (child exited) ---
<... printf resumed> = 9
SYS_exit_group(0 <no return ...>
+++ exited (status 0) +++
m7md@m7md-VirtualBox:~/Desktop$
```