5.2.1 System Calls with Linux

The UNIX system provides several system calls to create and end program, to send and receive software interrupts, to allocate memory, and to do other useful jobs for a process. The following exercises are using process and file system calls.

Use the following code Linux System Calls Colab link and solve the TODOs.

---

1. Process System Calls - myshell.c

What are the values that need to be placed in the TODOs numbered 1,2,3,4?

Question 1 1 / 1

TODO-1: the parent is waiting on the child thread.

- (•) -1
- ( ) 1
- ( ) 2
- ( ) 0

Question 2 1 / 1

TODO-2: Determine the character required to replace the space character from the read line. Enter the char.

- (•) carriage return
- ( ) line feed
- ( ) tab
- ( ) null character

Question 3 2 / 2

TODO-3: Fill the args array according to the man page of the system call

- ( ) -1
- (•) 0
- ( ) 1
- ( ) 2

Question 4 2 / 2

TODO-4: Execvp takes two paramters. Considering the array in args, set the parameters accordingly.

- ( ) char* todo_4_1 = args[-1]; char** todo_4_2 = args - 1;
- (•) char* todo_4_1 = args[0]; char** todo_4_2 = args + 0;
- ( ) char* todo_4_1 = args[1]; char** todo_4_2 = args + 1;
- ( ) char* todo_4_1 = args[2]; char** todo_4_2 = args + 2;

2. File System Calls - filesystem.c

Question 5 4 / 4

What are the values that need to be placed in the TODOs numbered 1,2,3,4?

- ( ) int todo_1 = O_RDWR | O_CRET | O_EXCL; int todo_2 = sizeof(message); int todo_3 = 0L; int todo_4 = sizeof(message);
- ( ) int todo_1 = O_RDWR | O_CREAT | O_EXCL; int todo_2 = sizeof(message); int todo_3 = 1L; int todo_4 = sizeof(message + 1);
- ( ) int todo_1 = 2_RDWR | O_CREAT | O_EXCL; int todo_2 = sizeof(message); int todo_3 = 2L; int todo_4 = sizeof(message + 2);
- (•) int todo_1 = O_RDWR | O_CREAT | O_EXCL; int todo_2 = sizeof(message); int todo_3 = 0L; int todo_4 = sizeof(message);