CS314 : Operating Systems Lab

Lab 2

Sourabh Bhosale (200010004)

January 15, 2023

1 Part 1

Problem

Print the string “Hello World” on screen. Each character must be printed by a different process. The process that prints the ith letter must have been spawned by the process that printed the (i-1)th letter. Each process must first print its designated character, as well as its own process ID, second, sleep for a random number of seconds (from 1 to 4 seconds), and then, do anything else it must do to achieve the given task.

Answer

Minimum lines of C code with which you can achieve is around 5 to 10. We can call `fork()` then run the loop till iterator travels through ’Hello World’ and whenever child process is there we call sleep, print the character and for parent process do `wait()` .

Now for execution process do following :
In host machine:

1. First navigate to folder where the zip file(containing .c make file) is present.
2. run following command for transferring the file to minix. (change file_name ip address accordingly.) `scp 200010004_lab2_part1.zip root@10.196.7.20:/root`

In minix system:

1. Navigate to ‘root‘ (preferably root here.)
2. extract zip file contents and then gmake will produce output. (which compiles with clang compiler, done.)

2 Part 2

Problem

Write a collection of programs twice, half, square such that they execute sequentially with the same process-id, and each program should also print its PID. (process id) The user should be able to invoke any combination of these programs, to achieve the required functionality. Using execvp system calls.

Answer

We get the arguments from command line and then we make new array of strings args, which gets updates as we iterate, and also the number's value changes accordingly.

```
# gmake
clang twice.c -o twice
clang half.c -o half
clang square.c -o square
# ./twice ./square ./half ./twice ./half 10
Twice: Current process id: 412, Current result: 20
Square: Current process id: 412, Current result: 400
Half: Current process id: 412, Current result: 200
Twice: Current process id: 412, Current result: 400
Half: Current process id: 412, Current result: 200
# _
```

Figure 1: `Execution output in minix system`

3 Part 3

Problem

Modify the Minix3 source code such that:

A message “Minix: PID <pid> created” is printed, whenever a process is created. (Let us follow the convention throughout this course that anything printed by the Operating System code will be prepended by the string “Minix: “.)

A message “Minix: PID <pid> exited” is printed, whenever a process ends.

Answer

In our programs, we can observe that while running commands like ls and so, it states that process created with its process id and then it shows the output of command and then it says process exited. Later on, I ran part_1 and part_2, in that also, similar observations were made.

```
112   /* Find a free pid for the child and put it in the table. */
113   new_pid = get_free_pid();
114   rmc->mp_pid = new_pid;     /* assign pid to child */
115   printf("Minix: PID %d created\n", rmc->mp_pid);  // pid created
116
```

Figure 2: `Changes made in code`

```
244   else {
245       exit_proc(mp, m_in.m_lc_pm_exit.status, FALSE /*dump_core*/);
246       printf("Minix: PID %d exited\n", mp->mp_pid);  // pid exited
247   }
248   return(SUSPEND);      /* can't communicate from beyond the grave */
```

Figure 3: `Changes made in code`

```
[minix/servers/pm $ scp forkexit.c root@10.196.6.193:/usr/src/minix/servers/pm
[root@10.196.6.193's password:
Permission denied, please try again.
[root@10.196.6.193's password:
forkexit.c                                    100%   25KB   1.4MB/s   00:00
minix/servers/pm $
```

Figure 4: `Transferring changed code files using scp`

Figure 5: `Execution output in minix system`