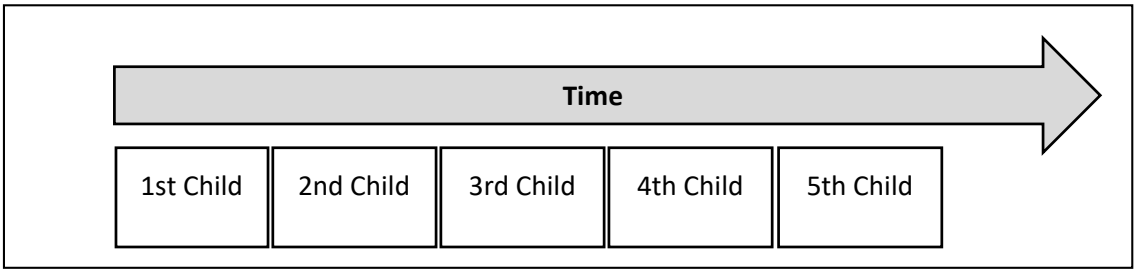
Exercises Lesson 4
(only process management)

Computers

Bachelor Degree in Data Science and Engineering

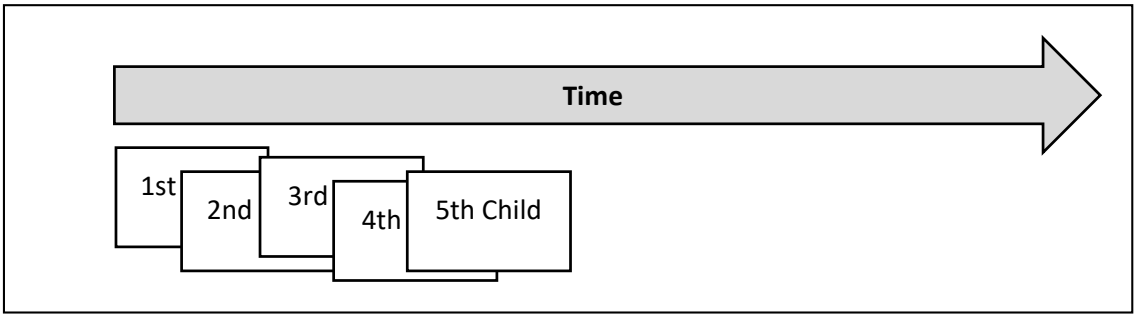
X. Martorell & X. Verdú
Computer Architecture Department
April - 2020

1. Implement a code to create the following execution timeline. You can assume child processes just print “Hello World! I am the XX child”. Besides, the parent process has to be sure not to finish before all created children have finished before.
    a. Besides, write the execution schema of the following behavior:



Code:

    b. Besides, write the execution schema of the following behavior:



Code:

2. The following two codes do not take the same amount of time to reach the end of the execution. Why? **HINT:** *The reason is related to the topics discussed in lesson 4.*

Code A

```
1.  int main(){
2.  int i; char buffer[10000];
3.  …
4.  /* Code to fill the buffer */
5.  …
6.  for (i=0; i<10000; i++)
7.     write(1, &buffer[i], 1);
8.  }
```

Code B

```
1.  int main(){
2.  int i; char buffer[10000];
3.  …
4.  /* Code to fill the buffer */
5.  …
6.
7.  write(1, &buffer, 10000);
8.  }
```

Answer:

3. Answer the following questions from the execution of the following code:

```
1.  int main(){
2.  /* Declaration of variables */
3.  …
4.  ret = fork();
5.  tmp = getpid();
6.  printf("Value of tmp %d – fork returns %d\n", tmp, ret);
7.  }
```

a. What text can you see printed in the screen?

Answer:

b. If we repeat the execution several times, are you going to see always exactly the same text? Why?

Answer:

4. Draw the process hierarchy created due to the execution of the following code and assign a PID to every process.

```
1.  int main(){
2.  /* Declaration of variables */
3.  …
4.  for (i=0; i<3; i++)
5.      fork();
6.  }
```

Answer:

5. Answer the questions related to the execution of the following code. Every question is done based from the original code. That is, the changes are not accumulated:

CODE: myprog.cpp

```
1.  int main(){
2.  /* Declaration of variables */
3.  …
4.  for (i=0; i<2; i++)
5.      if (fork() == 0){
6.          execlp("ls", "ls", "-l", NULL);
7.      else
8.          waitpid(-1, NULL, 0);
9.  }
```

a. What does this program do?

Answer:

b. What happen if line 6 changes to "*execlp("./myprog", "./myprog", NULL);*"?

Answer:

c. What happen if line 5 changes to "*if (fork() > 0) {*"?

Answer: