

## Disciplina: Sistemas Operacionais I

<https://github.com/diogobranquinho>

### Exercício 1

1. Elabore um programa em Linux que crie um processo e execute um segundo programa (Aula 07 – Slide 17).

```
processos.c x processos.python x
#include <sys/types.h>
#include <stdio.h>
#include <unistd.h>

int main (void){
    pid_t pid;

    pid = fork();

    if (pid<0){
        fprintf(stderr, "Falha no fork");
        exit(-1);
    } else if (pid == 0){
        printf("Pai chama o filho para exec \n\n\r");
        execlp("/bin/ls","ls", NULL);
    } else {
        wait(NULL);
        printf("\n\rFilho completou a exec \n\r");
        exit(0);
    }
}
```

- a. Explique o código fonte utilizado

```
ubuntu@ubuntu-VirtualBox: ~/Documents
ubuntu@ubuntu-VirtualBox:~$ ls
Desktop  Downloads  Music  Public  Videos
Documents  examples.desktop  Pictures  Templates
ubuntu@ubuntu-VirtualBox:~$ cd Documents
ubuntu@ubuntu-VirtualBox:~/Documents$ ls
processos.c  processos.python  Untitled Document 3
processos.c~  processos.python~
ubuntu@ubuntu-VirtualBox:~/Documents$ gcc -o processos processos.C
gcc: error: processos.C: No such file or directory
gcc: fatal error: no input files
compilation terminated.
ubuntu@ubuntu-VirtualBox:~/Documents$ gcc -o processos processos.c
processos.c: In function 'main':
processos.c:12:10: warning: incompatible implicit declaration of built-in function 'exit' [enabled by default]
processos.c:19:10: warning: incompatible implicit declaration of built-in function 'exit' [enabled by default]
ubuntu@ubuntu-VirtualBox:~/Documents$ ls
processos  processos.c~  processos.python~
processos.c  processos.python  Untitled Document 3
ubuntu@ubuntu-VirtualBox:~/Documents$ ./processos
Pai chama o filho para exec

processos  processos.c~  processos.python~
processos.c  processos.python  Untitled Document 3

Filho completou a exec
ubuntu@ubuntu-VirtualBox:~/Documents$
```

- b. Descreva os resultados obtidos
  - 2. Elabore um programa em Windows que crie um processo e execute um segundo programa (Aula 07 – Slide 19).
    - a. Explique o código fonte utilizado
- Código do exercício:

```

[*] Sem Título1
2  #include <windows.h>
3
4  int main(VOID)
5  {
6      STARTUPINFO si;
7      PROCESS_INFORMATION pi;
8
9      //allocate memory
10     ZeroMemory(&si, sizeof(si));
11     si.cb = sizeof(si);
12     ZeroMemory (&pi, sizeof(pi));
13
14     //create child process
15     if (!CreateProcess(NULL, //use command line
16         "C:\\WINDOWS\\system\\mspaint.exe", //command line
17         NULL, //don't inherit process handle
18         NULL, //don't inherit thread handle
19         FALSE, //disable handle inheritance
20         0, //no creation flags
21         NULL, //use parent's environment block
22         NULL, //use parent's existing directory
23         &si,
24         &pi))
25     {
26         fprintf(stderr, "Create Process Failed");
27         return -1;
28     }
29     //parent will wait for the child to complete
30     WaitForSingleObject(pi.hProcess, INFINITE);
31     printf("Child Complete");
32
33     //close handles
34     CloseHandle(pi.hProcess);
35     CloseHandle(pi.hThread);
36 }
37

```

- b. Descreva os resultados obtidos

The screenshot shows a C++ IDE with two windows. The top window is a console titled "C:\Users\User\Documents\Sem Título1.exe" displaying the following text:

```
Create Process Failed
-----
Process exited after 0.2102 seconds with return value 4294967295
Pressione qualquer tecla para continuar. . .
```

The bottom window is the IDE editor titled "C:\Users\User\Documents\Sem Título1.cpp - Dev-C++ 5.11". It contains the following C++ code:

```
1 #include <stdio.h>
2 #include <windows.h>
3
4 int main(VOID)
5 {
6     STARTUPINFO si;
7     PROCESS_INFORMATION pi;
8
9     //allocate memory
10    ZeroMemory(&si, sizeof(si));
11    si.cb = sizeof(si);
12    ZeroMemory(&pi, sizeof(pi));
13
14    //create child process
15    if (!CreateProcess(NULL, //use command line
16        "C:\\WINDOWS\\system\\nspaint.exe", //command line
17        NULL, //don't inherit process handle
18        NULL, //don't inherit thread handle
19        FALSE, //disable handle inheritance
20        0, //no creation flags
21        NULL, //use parent's environment block
22        NULL, //use parent's existing directory
23        &si,
24        &pi))
25    {
26        fprintf(stderr, "Create Process Failed");
27        return -1;
28    }
29    //parent will wait for the child to complete
30    WaitForSingleObject(pi.hProcess, INFINITE);
31    printf("child complete");
32
33    //close handles
34    CloseHandle(pi.hProcess);
35    CloseHandle(pi.hThread);
36
37 }
```

The status bar at the bottom indicates: "Linha: 37 Col: 1 Set: 0 Linhas: 37 Tamanho: 902 Inserir Done parsing in 7,941 seconds".

O resultado e esta resposta acima, pois ao rodar e feito a execução do tempo em segundos e a velocidade do mesmo.

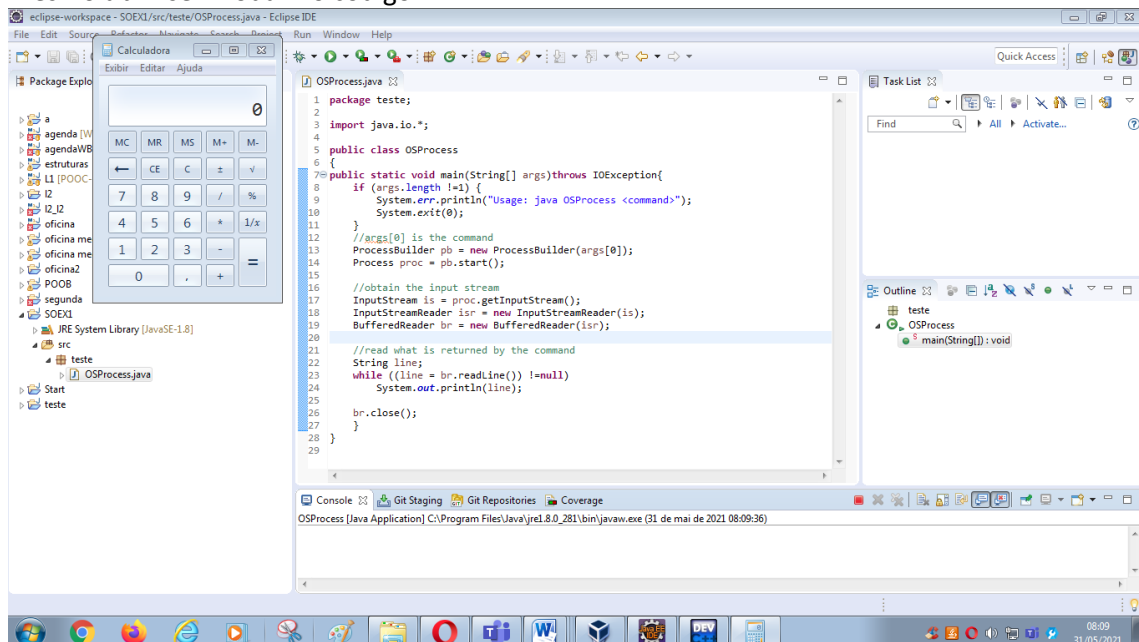
3. Elabore um programa em Java que crie um processo e execute um segundo programa (Aula 07 – Slide 20).

a. Explique o código fonte utilizado

```
OSProcess.java
1 package teste;
2
3 import java.io.*;
4
5 public class OSProcess
6 {
7     public static void main(String[] args) throws IOException{
8         if (args.length != 1) {
9             System.err.println("Usage: java OSProcess <command>");
10            System.exit(0);
11        }
12        //args[0] is the command
13        ProcessBuilder pb = new ProcessBuilder(args[0]);
14        Process proc = pb.start();
15
16        //obtain the input stream
17        InputStream is = proc.getInputStream();
18        InputStreamReader isr = new InputStreamReader(is);
19        BufferedReader br = new BufferedReader(isr);
20
21        //read what is returned by the command
22        String line;
23        while ((line = br.readLine()) != null)
24            System.out.println(line);
25
26        br.close();
27    }
28 }
29
```

b. Descreva os resultados obtidos

Quando é realizada a configuração dentro do run, depois em Java Application e posteriormente em OSProcess, uma mudança de sentido é realizada com a execução do código e dentro da parte de comentários é feito utilizando o exemplo de calc.exe na parte de argumentos, assim ao rodar automaticamente abre um aplicativo que neste caso é a calculadora para ser exibido o que será feito, e segue como um programa qualquer, mesmo abrir sem rodar no código.



4. Elabore um programa em Python que crie um processo e execute um segundo programa.
- a. Explique o código fonte utilizado

```
processos.python (~/.Documents) - gedit
Open Save Undo
processos.c x processos.python x
# Python code to create child process
import os
from datetime import datetime

n = os.fork()
# n greater than 0 means parent process
if n > 0:
    print("Parent process id is : ", os.getpid())
# n equals to 0 means child process
else:
    print("Child process id is : ", os.getpid())

print(datetime.now())
```

B. Descreva os resultados obtidos.

```
processos      processos.c~      processos.python~
processos.c    processos.python  Untitled Document 3

Filho completou a exec
ubuntu@ubuntu-VirtualBox:~/Documents$ python processos.python
('Parent process id is : ', 4596)
('Child process id is : ', 4597)
2021-06-01 12:51:22.097440
2021-06-01 12:51:22.092993
ubuntu@ubuntu-VirtualBox:~/Documents$
```

PADRÃO DE ENTREGA:

NomeEX1\_SO1AouB.pdf