# Sistemas Operativos 2021/2022
# Gestión de procesos y Archivos

Rana Abadi

October 22, 2021

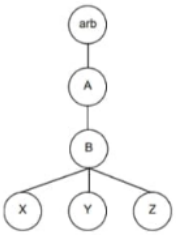
## Contents

**1 Ejercicio 1: Gestión básica de procesos** **2**

**2 Ejercicio 3: Comunicación entre procesos: memoria compartida** **3**

# 1 Ejercicio 1: Gestión básica de procesos

**En este ejercicio se nos pide que realicemos un programa que genere un árbol de procesos que se indica y que tenga como funcionalidad que tras pasar unos segundos (indicado por argumento) ejecute el comando pstree o el comando ls, esto dependerá de proceso sea. También tendremos que controlar la muerte del árbol, ya que un padre no puede morir antes que un hijo.**




Como se puede apreciar en la foto de ejemplo del ejercicio podemos ver que se trata de un árbol que es fijo, que tiene dos hijos del proceso padre y otros 3 subhijos por así decirlo.

Este ejercicio se fundamenta en que vamos a tener muchos PIDS por lo que tendremos que decidir el tipo de datos a usar, podría ser interesante usar arrays de ints o simplemente ints, es algo un poco indiferente. La lógica del programa se basa en que cada vez que creamos un nuevo proceso tenemos que mostrar por pantalla el pid que le corresponde a ese proceso y el pid de su padre. Como se ha mencionado por argumento se nos pasa el valor de los segundos hasta que se ejecute el comando correspondiente. Finalmente hay que mostrar por pantalla cada PID de cada procesos e ir matándolos poco a poco de la forma adecuada.

Prueba de ejecución, vamos a ejecutar con 5 segundos de retraso hasta que se ejecute el comando:

```
programador@programador-pc:~/Música/Prac1$ ./ejec 5
Soy el proceso ejec: Mi pid es: 7363
Soy el proceso A: Mi pid es: 7364. Mi padre es: 7363
Soy el proceso B: Mi pid es: 7365. Mi padre es: 7364, mi abuelo es: 7363
Soy el proceso X: Mi pid es: 7366. Mi padre es 7365, abuelo 7364, mi bisabuelo es: 7363
Soy el proceso Y: Mi pid es: 7367. Mi padre es 7365, abuelo 7364, mi bisabuelo es: 7363
Soy el proceso Z: Mi pid es: 7368. Mi padre es 7365, abuelo 7364, mi bisabuelo es: 7363
```

Tras los 5 segundos tenemos el siguiente resultado:

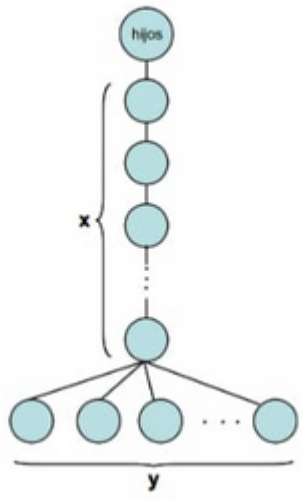
```
    ├─gvfsd
    │  ├─gvfsd-trash --spawner :1.3 /org/gtk/gvfs/exec_spaw/0
    │  │  └─2*[{gvfsd-trash}]
    │  └─2*[{gvfsd}]
    ├─gvfsd-fuse /run/user/1003/gvfs -f -o big_writes
    │  └─5*[{gvfsd-fuse}]
    ├─gvfsd-metadata
    │  └─2*[{gvfsd-metadata}]
    ├─obexd
    └─pulseaudio --daemonize=no --log-target=journal
       └─3*[{pulseaudio}]
  ├─systemd-journal
  ├─systemd-logind
  ├─systemd-resolve
  ├─systemd-timesyn
  │  └─{systemd-timesyn}
  ├─systemd-udevd
  ├─udisksd
  │  └─4*[{udisksd}]
  ├─upowerd
  │  └─2*[{upowerd}]
  └─wpa_supplicant -u -s -O /run/wpa_supplicant
Soy Z (7368) y muero
Soy Y (7367) y muero
Soy X (7366) y muero
Soy B(7365) y muero
Soy A(7364) y muero
Soy ejec(7363) y muero
```

En este ejercicio he tenido problemas a la hora de mostrar el otro comando, sólo he podido plantearlo con pstree.

## 2 Ejercicio 3: Comunicación entre procesos: memoria compartida

**Tenemos que diseñar un programa que cree un árbol de procesos según la siguiente estructura a partir de los datos pasados por parámetro. También tiene que mostrar unos mensajes por la salida estándar.**

```
$hijos x y
```




El funcionamiento o la lógica de este ejercicio es parecido al del primer ejercicio, la única gran diferencia es que ahora la altura y anchura puede variar, en si lo que hay que hacer es ir guardando el PID de cada uno de los procesos para poder mostrarlos en la salida estándar. Lo que se hace en primer lugar es obtener los datos de la anchura y altura que están en los argumentos del programa.
Ahora realizaremos una prueba de ejecución:

```
programador@programador-pc:~/Música/Prac1$ ./hijos 2 3
Soy el superpadre(13609): mis hijos finales son: 13613 Soy el superpadre(13609): mis hijos finales son: 13614 .
Soy el subhijo 13612, mis padres son: Soy el superpadre(13609): mis hijos finales son:  13611 13611.
13615 13616 .
Soy el subhijo 13614, mis padres son:  13611 13611.
programador@programador-pc:~/Música/Prac1$ ./hijos 3 2
Soy el superpadre(13649): mis hijos finales son: 13654 .
Soy el subhijo 13653, mis padres son: Soy el superpadre(13649): mis hijos finales son:  13651 13651 13651.
13655 .
Soy el subhijo 13654, mis padres son:  13651 13651 13651.
programador@programador-pc:~/Música/Prac1$
```