

Benemérita Universidad Autónoma de Puebla
Facultad de Ciencias de la Computación
Sistemas Operativos 1



“Manejo de procesos: Abanicos”

*Gallegos Melchor Angélica
Hernández Fernández Saúl
Muñoz Tehuitzil Jorge Leonel
Pedraza Celón Ian Yael*

07/02/2020

Un proceso es una instancia de un programa en ejecución con datos asociados (variables y buffers) y contexto de ejecución, en donde se almacena toda la información que el CPU necesita para ejecutarlo como, por ejemplo: registros, prioridades y eventos por los que espera.

Un proceso tiene un ciclo de vida que consta de cinco estados:

New: Se ejecuta cuando el proceso es creado.

Ready: El estado ready nos indica cuando el proceso está listo para ser ejecutado, este puede ser llamado después de esperar algún otro recurso, como datos de entradas o algún evento.

Running: El proceso entra en estado running cuando se está ejecutando el proceso, desde este estado podemos indicar que necesitamos algún recurso o estamos a la espera de algún evento, como por ejemplo entrada y salida de datos.

Blocked: Este estado bloquea el proceso, lo mantiene en pausa o espera hasta que cierto evento que espere sea ejecutado. Una vez llegado este evento, vuelve a estar en estado *Ready*.

Exit: Es ejecutado cuando el proceso termina y devuelve un código de salida dependiendo si el proceso fue exitoso o no.

El PCB (*Process Control Block*) es una estructura que contiene información relevante del proceso:

- Identificación (PID, UID, ID proceso padre)
- Información del procesador (PC, CPU registros..., PSW, SP)
- Información de control (Prioridad, estado del proceso, evento por los que espera, links a otros procesos)
- Información de memoria (localización y estado de acceso de los datos usuarios)
- Información de Archivos (archivos y dispositivos abiertos)

Desarrollo:

Escriba un programa que cree una cadena de tres procesos y el último hijo un abanico de tres procesos

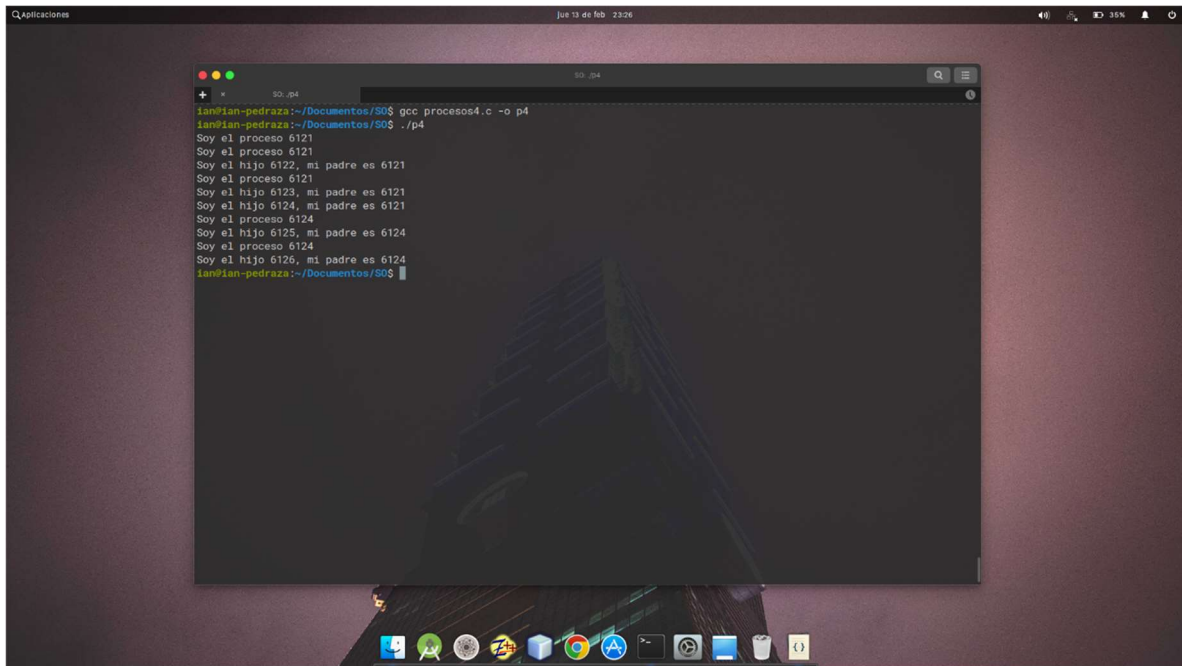
```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>

int main(int argc, char *argv[]){
    pid_t pid;
    int i, j;

    for(i = 0; i < 3; i++){
        pid = fork();
        if(pid){
            printf("Soy el proceso %d\n", getpid());
        }else{
            printf("Soy el hijo %d, mi padre es %d\n", getpid(), getppid());
            if(i == 2){
                for(j = 0; j < 2; j++){
                    pid = fork();

                    if(pid){
                        printf("Soy el proceso %d\n", getpid());
                    }else{
                        printf("Soy el hijo %d, mi padre es %d\n", getpid(), getppid());
                        exit(0);
                    }
                }
            }
            exit(0);
        }
    }

    return 0;
}
```

A screenshot of a macOS desktop environment. At the top, the menu bar shows 'Aplicaciones', the date 'Jue 13 de feb', and the time '23:26'. The desktop background is a dark, abstract image. A terminal window titled 'SO_p4' is open, displaying the following text:

```
ian@ian-pedraza:~/Documentos/SO$ gcc procesos4.c -o p4
ian@ian-pedraza:~/Documentos/SO$ ./p4
Soy el proceso 6121
Soy el proceso 6121
Soy el hijo 6122, mi padre es 6121
Soy el proceso 6121
Soy el hijo 6123, mi padre es 6121
Soy el hijo 6124, mi padre es 6121
Soy el proceso 6124
Soy el hijo 6125, mi padre es 6124
Soy el proceso 6124
Soy el hijo 6126, mi padre es 6124
ian@ian-pedraza:~/Documentos/SO$
```

The terminal window has a dark theme and standard macOS window controls. The dock at the bottom contains various application icons including Finder, Launchpad, and several web browsers.

Conclusiones:

Un sistema operativo es el encargado de proveer de servicios a las aplicaciones y administrar los recursos del sistema, y un servicio no es otra cosa que un proceso, y para nosotros como desarrolladores, es muy importante conocer el como se crean y como se manejan.

También es de mucha importancia conocer el ciclo de vida de un proceso, para poder saber como administrar las funciones que ejecuta y en que momento deben realizarse.

Bibliografía.

Bioinformatics at COMAV. (2019). Procesos. 06/02/2020, de COMAV Sitio web:
https://bioinf.comav.upv.es/courses/unix/control_procesos.html

González S.. (2019). Manual básico de administración de procesos . 06/02/2020, de
Linux Total Sitio web:
https://www.linuxtotal.com.mx/index.php?cont=info_admon_012