Programación Sobre Redes

T3: Procesos

Nicolás Mastropasqua

April 24, 2020

Instituto Industrial Luis A. Huergo

Contenidos

Repaso

Previamente...

- Sistemas multiprocesador
- Scheduling multiprocesador
- Computación paralela
- Ley de Amdahl

Procesos

¿Qué es un proceso?

Concepto

- Podemos entender a un proceso como a un programa en ejecución. Es una entidad viva
- Un programa, almacenado en algún lugar, es una entidad pasiva que no es un proceso.

Proceso vs Programa

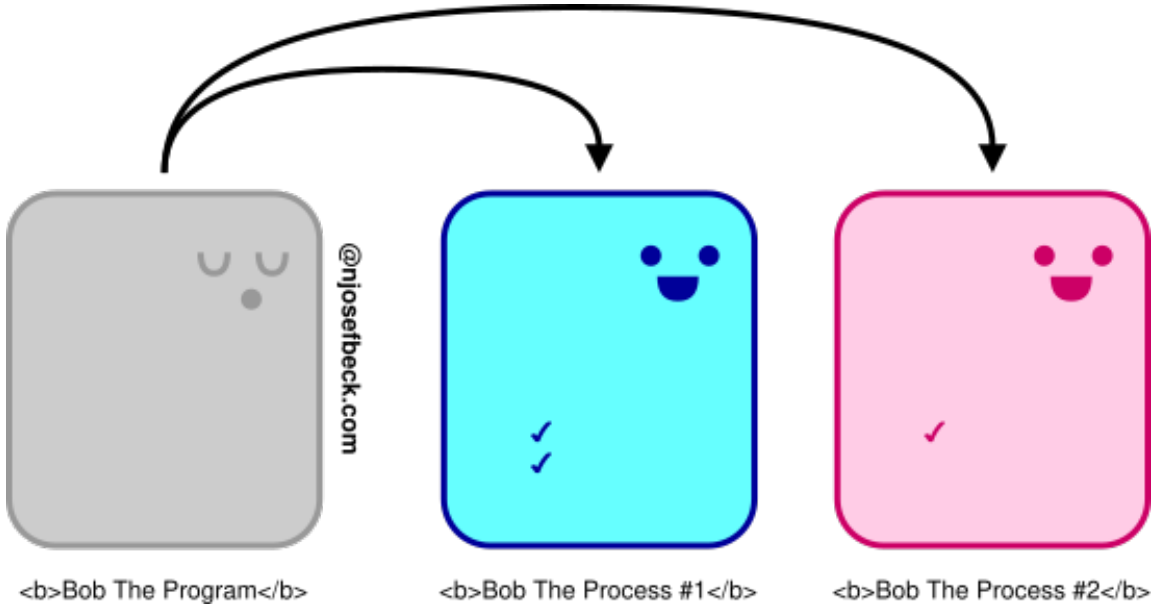


Figure 1: Procesos como entidades vivas y programas como entidades dormidas

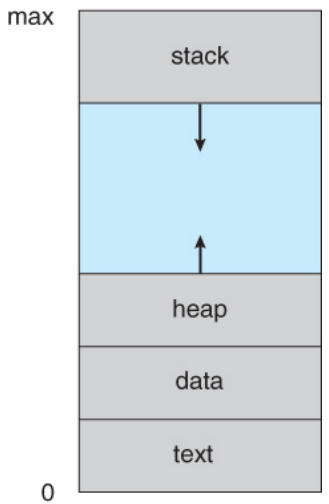
Para pensar: ¿Dos procesos pueden estar asociados al mismo programa?

Proceso en memoria

Además del código (sección de texto), el proceso cuenta con:

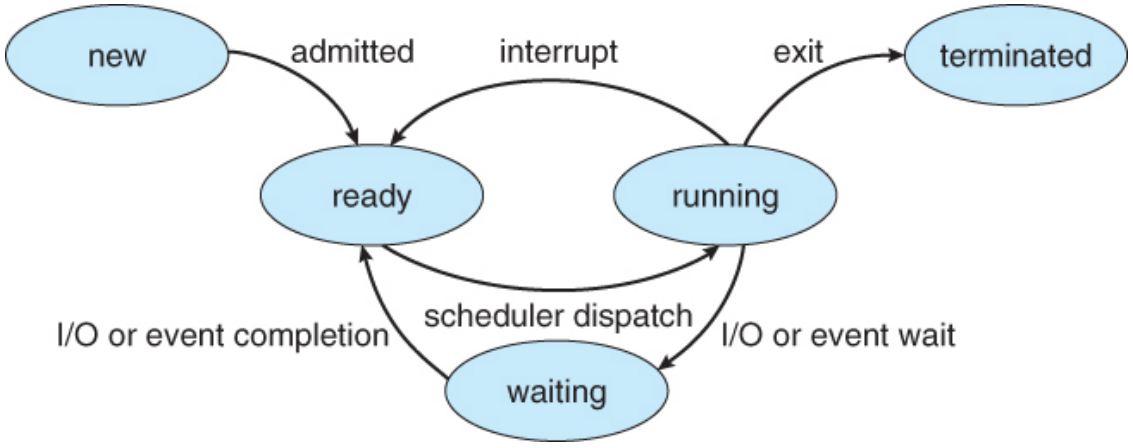
Partes de un proceso

- Program Counter
- Stack
- Sección de datos (variables globales)
- Heap



Veamos un ejemplo!

Estados de un proceso

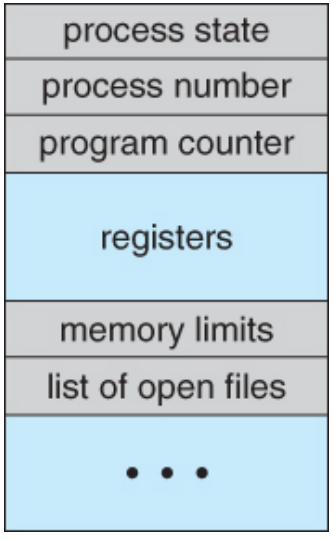


Máquina de estados

El proceso solo puede estar en un estado.

Las flechas indican los eventos que deben ocurrir para transicionar de un estado a otro

Representación de un proceso: PCB



PCB

- El **PCB** (Process Control Block) es un repositorio que almacena información relevante del proceso para que el mismo pueda ejecutar y reanudar su ejecución en caso de ser desalojado.
- Recordar que solo un proceso puede ejecutar al mismo tiempo (asumiendo single core)

Context swtich

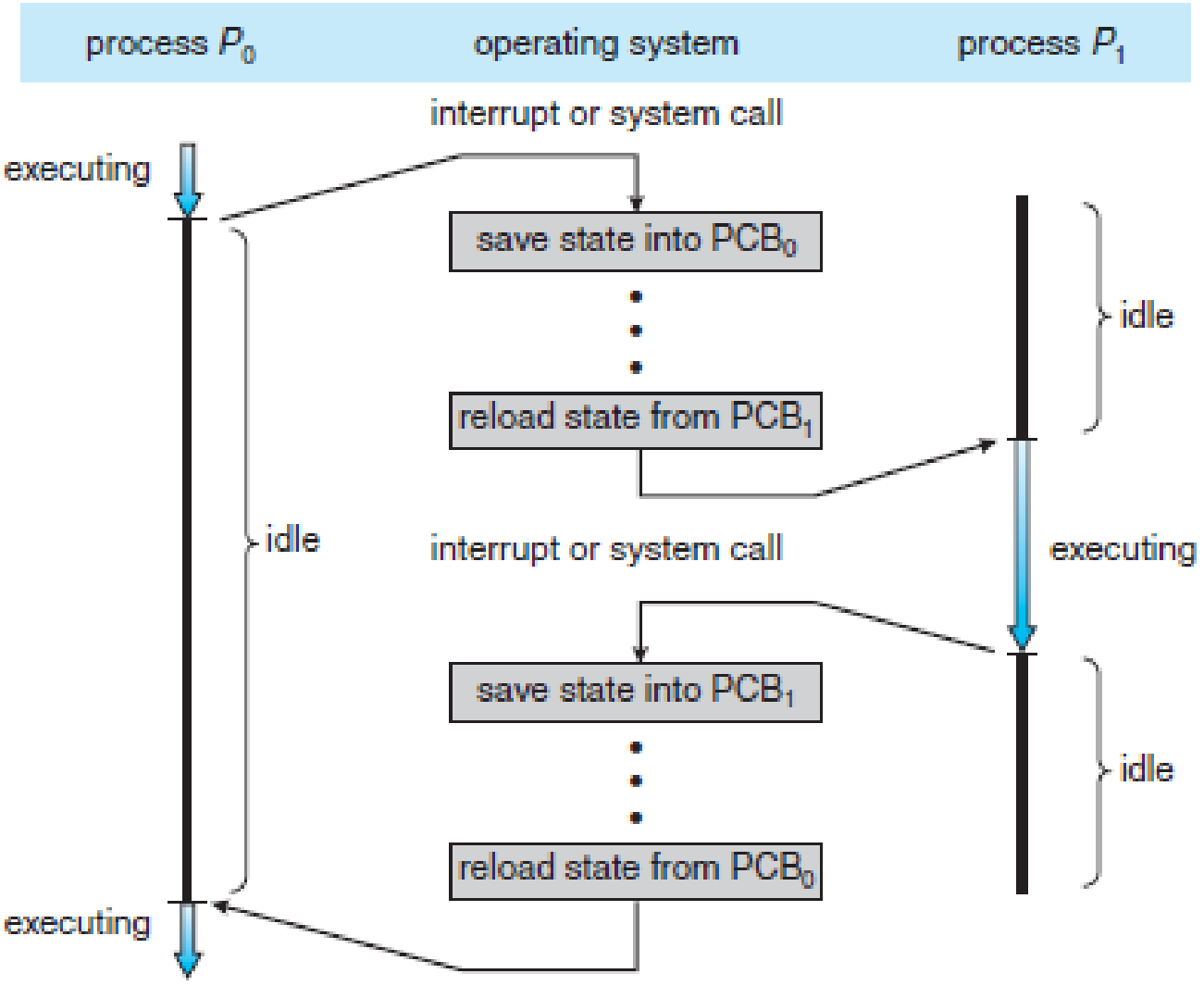
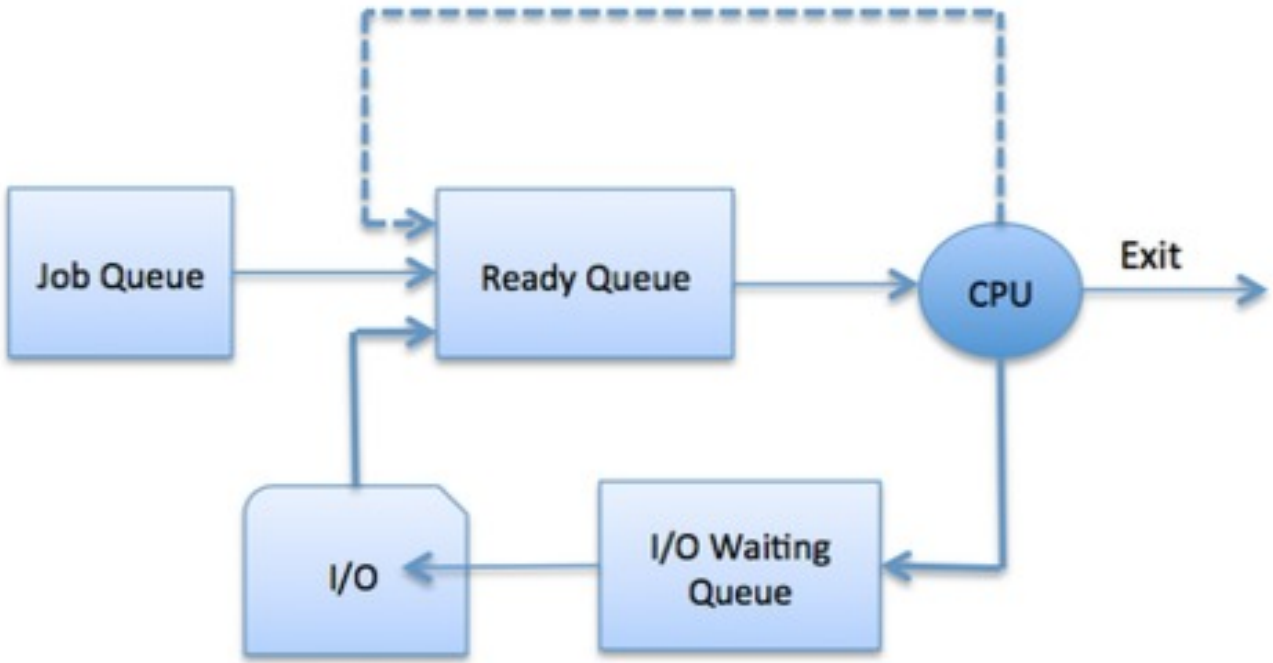


Figure 3.4 Diagram showing CPU switch from process to process.

Scheduling de procesos... ¿se acuerdan?



Scheduling de procesos

Cuando el proceso es despachado desde la ready queue al CPU, comienza a ejecutar. Pueden ocurrir lo siguientes eventos:

- El proceso requiere una operación de E/S, por lo tanto es desalojado y colocado en la I/O queue correspondiente
- El proceso crea un subproceso y decide esperar su terminación
- El proceso es desalojado forzosamente por la aparición de una interrupción

Eventualmente, en los dos primeros casos, el proceso volverá a al estado de ready

Abandonando el estado de ready

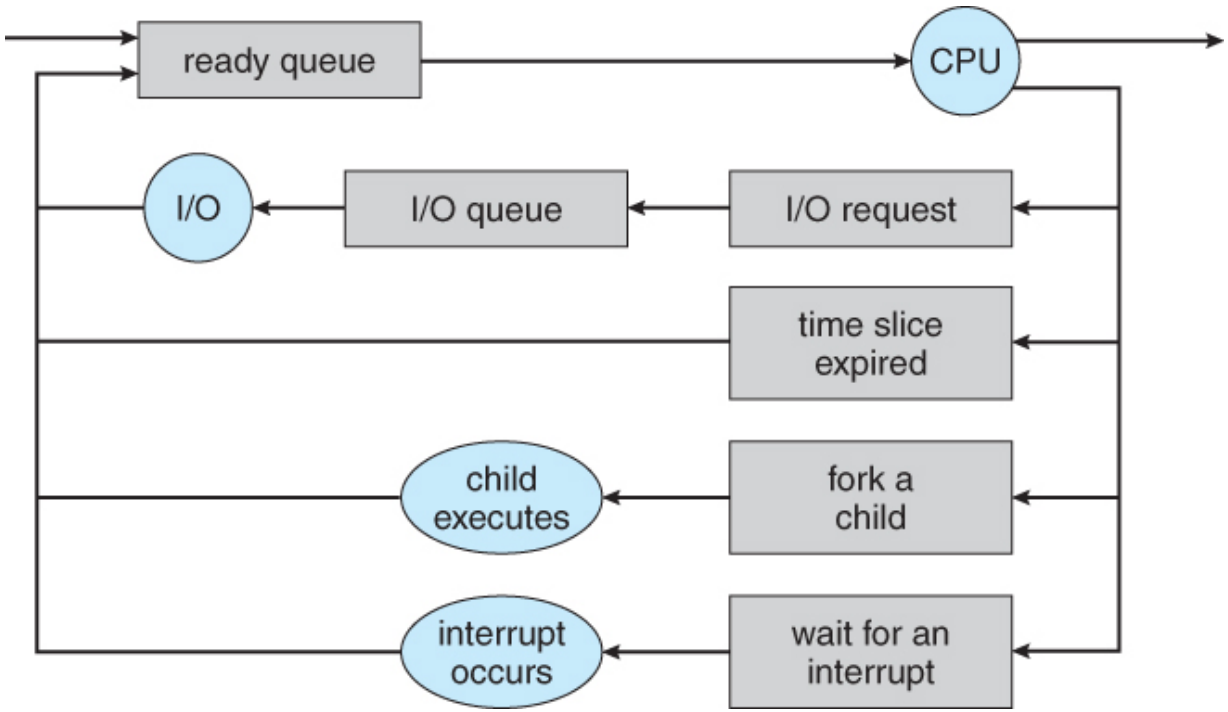


Figure 2: En la figura, los rectángulos representan colas distintas (a diferencia de lo que veníamos asumiendo)

Procesos... ¿para qué?

- Para ejecutar tareas concurrentemente
- Para modularizar, por ej. al desarrollar un sistema
- Para paralelizar

Creación de procesos

- Un proceso puede crear varios procesos hijos.
- De esta forma, se obtiene una estructura en forma de árbol

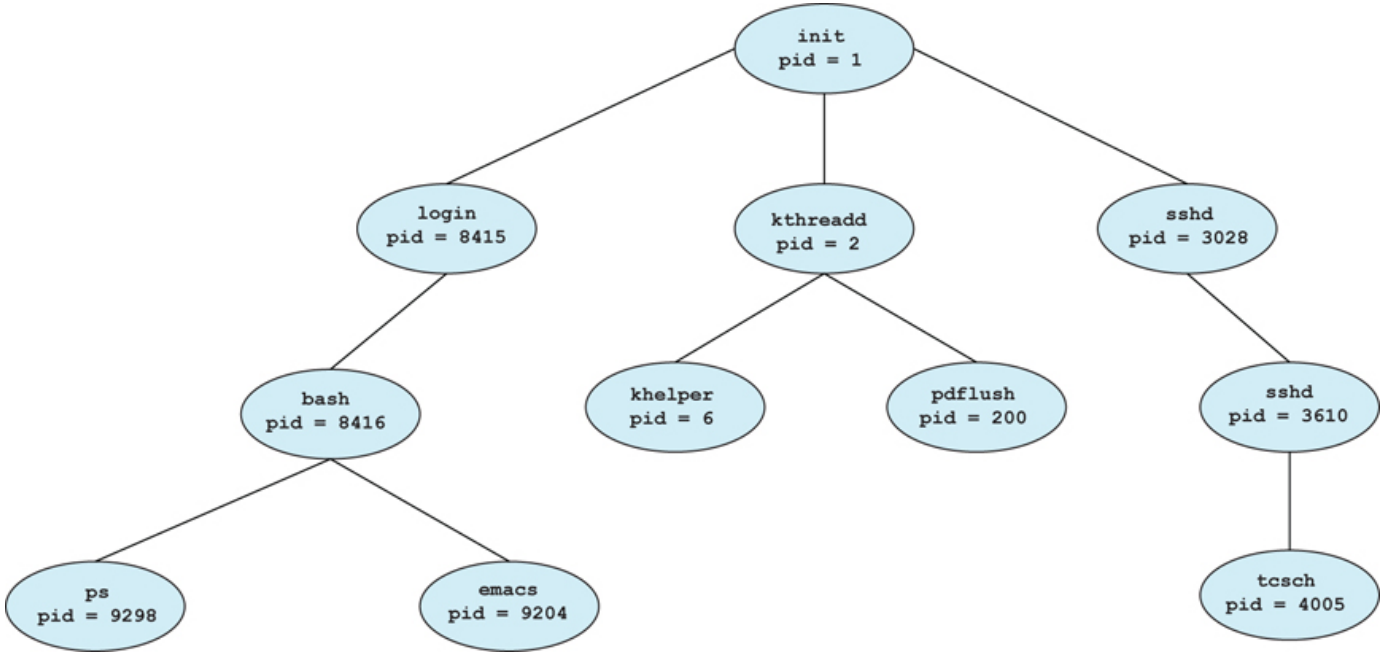


Figure 3: Ejemplo arbol de procesos en UNIX

Sugerencia: Jugar con **pstree** en linux y **ps** en linux. Pueden ver el siguiente link para el anterior

Bibiliografía

Abraham Silberschatz, Peter Baer Galvin, Greg Gagne. ***Operating System Concepts Eigth Edition****, Capítulo 3*