

# *Introducción a los Sistemas Operativos*

## Procesos - III

### **Profesores:**

Lía Molinari

Juan Pablo Pérez

Nicolás del Río



✓ Versión: Septiembre 2019

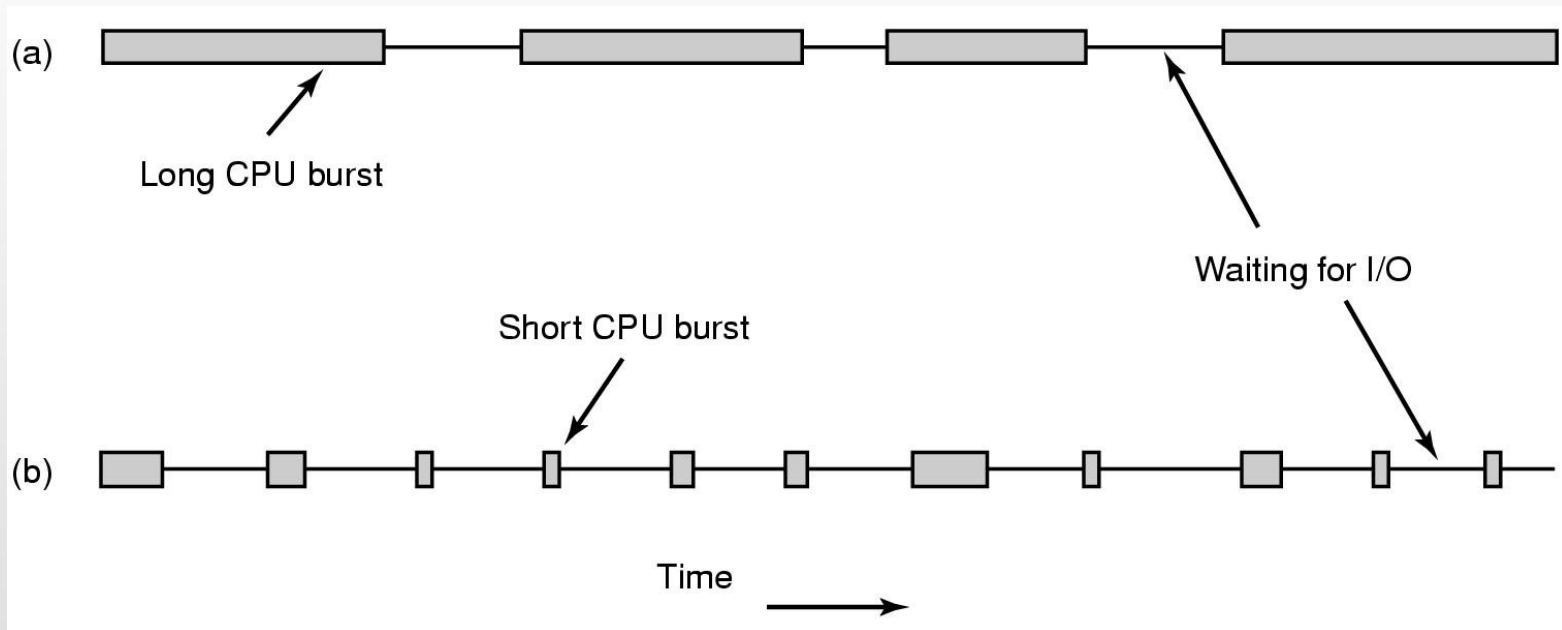
Palabras Claves: Procesos, Planificación, FCFS, SJF, Round Robin, SRTF, Prioridades, Algoritmos Apropiativos y Algoritmos No Apropiativos

Los temas vistos en estas diapositivas han sido mayormente extraídos del libro de Andrew S. Tanenbaum (Sistemas Operativos Modernos)



# Comportamiento de los procesos

- ✓ Procesos alternan ráfagas de CPU y de I/O.



# Comportamiento de los procesos (cont.)

## ☑ CPU-bound

- ✓ Mayor parte del tiempo utilizando la CPU

## ☑ I/O-bound (I/O = E/S)

- ✓ Mayor parte del tiempo esperando por I/O

## ☑ La velocidad de la CPU es mucho mas rápida que la de los dispositivos de E/S

- ✓ Pensar: Necesidad de atender rápidamente procesos I/O-bound para mantener el dispositivo ocupado y aprovechar la CPU para procesos CPU-bound



# Planificación

## ☑ Planificación:

- Necesidad de determinar cual de todos los procesos que están listos para ejecutarse, se ejecutará a continuación en un ambiente multiprogramado

## ☑ Algoritmo de Planificación

- Algoritmo utilizado para realizar la planificación del sistema



# Algoritmos Apropiativos y No Apropiativos

- ✓ En los algoritmos Apropiativos (preemptive) existen situaciones que hacen que el proceso en ejecución sea expulsado de la CPU
- ✓ En los algoritmos No Apropiativo (nonpreemptive) los procesos se ejecutan hasta que el mismo (por su propia cuenta) abandone la CPU
  - Se bloquea por E/S o finaliza
  - No hay decisiones de planificación durante las interrupciones de reloj



# *Categorías de los Algoritmos de Planificación*

- ☑ Según el ambiente es posible requerir algoritmos de planificación diferentes, con diferentes metas:
  - ✓ Equidad: Otorgar una parte justa de la CPU a cada proceso
  - ✓ Balance: Mantener ocupadas todas las partes del sistema
- ☑ Ejemplos:
  - ✓ Procesos por lotes (batch)
  - ✓ Procesos Interactivos
  - ✓ Procesos en Tiempo Real



# Procesos Batch

- ☑ No existen usuarios que esperen una respuesta en una terminal.
- ☑ Se pueden utilizar algoritmos no apropiativos
- ☑ Metas propias de este tipo de algoritmos:
  - ✓ Rendimiento: Maximizar el número de trabajos por hora
  - ✓ Tiempo de Retorno: Minimizar los tiempos entre el comienzo y la finalización
  - ✓ El Tiempo es espera se puede ver afectado
  - ✓ Uso de la CPU: Mantener la CPU ocupada la mayor cantidad de tiempo posible





# Procesos Batch (cont.)

- ✓ Ejemplos de Algoritmos:
  - ✓ FCFS – First Come First Served
  - ✓ SJF – Shortest Job First

8	4	4	4
A	B	C	D

(a)

4	4	4	8
B	C	D	A

(b)



# Procesos Interactivos

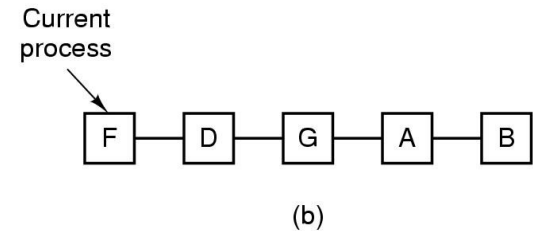
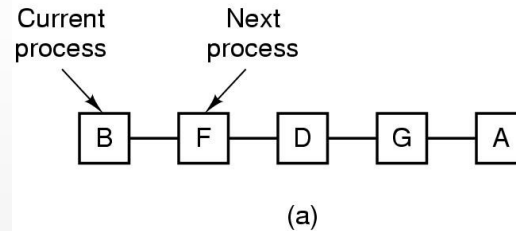
- ☑ No solo interacción con los usuarios
  - ✓ Un servidor, necesita de varios procesos para dar respuesta a diferentes requerimientos
- ☑ Son necesarios algoritmos apropiativos para evitar que un proceso acapare la CPU
- ☑ Metas propias de este tipo de algoritmos:
  - ✓ Tiempo de Respuesta: Responder a peticiones con rapidez
  - ✓ Proporcionalidad: Cumplir con expectativas de los usuarios
    - ♦ Si el usuario le pone STOP al reproductor de música, que la música deje de ser reproducida en un tiempo considerablemente corto.



# Procesos Interactivos (cont.)

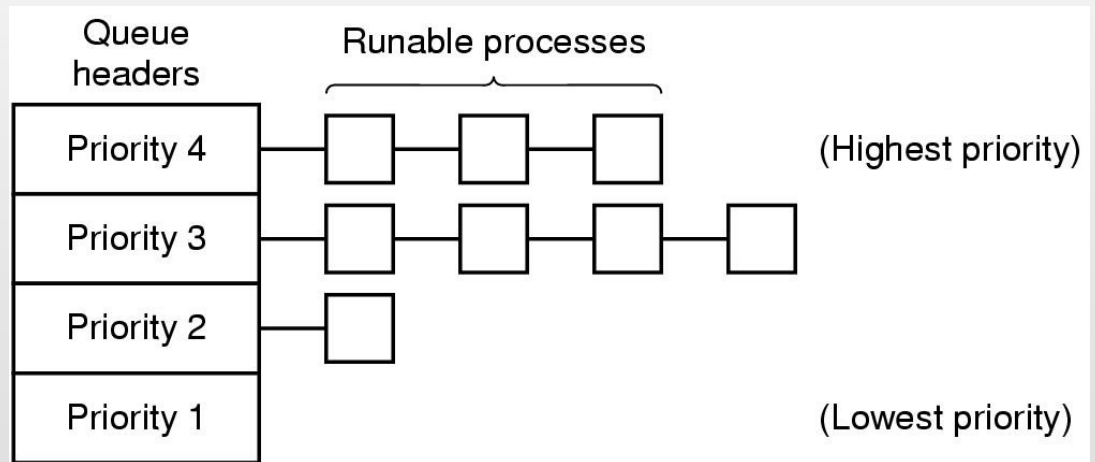
## ✓ Ejemplos de Algoritmos:

### ✓ Round Robin



### ✓ Prioridades

### ✓ Colas Multinivel



### ✓ SRTF – Shortest remaining time first



# *Política Versus Mecanismo*

- ✓ Existen situaciones en las que es necesario que la planificación de uno o varios procesos se comporte de manera diferente
- ✓ El algoritmo de planificación debe estar parametrizado, de manera que los procesos/usuarios pueden indicar los parámetros para modificar la planificación



# Política Versus Mecanismo (cont.)

- ✓ El Kernel implementa el mecanismo
- ✓ El usuario/proceso/administrador utiliza los parámetros para determinar la Política
- ✓ Ejemplo:
  - ✓ Un algoritmo de planificación por prioridades y una System Call que permite modificar la prioridad de un proceso (man nice)
  - ✓ Un proceso puede determinar las prioridades de los procesos que el crea, según la importancia de los mismos.

