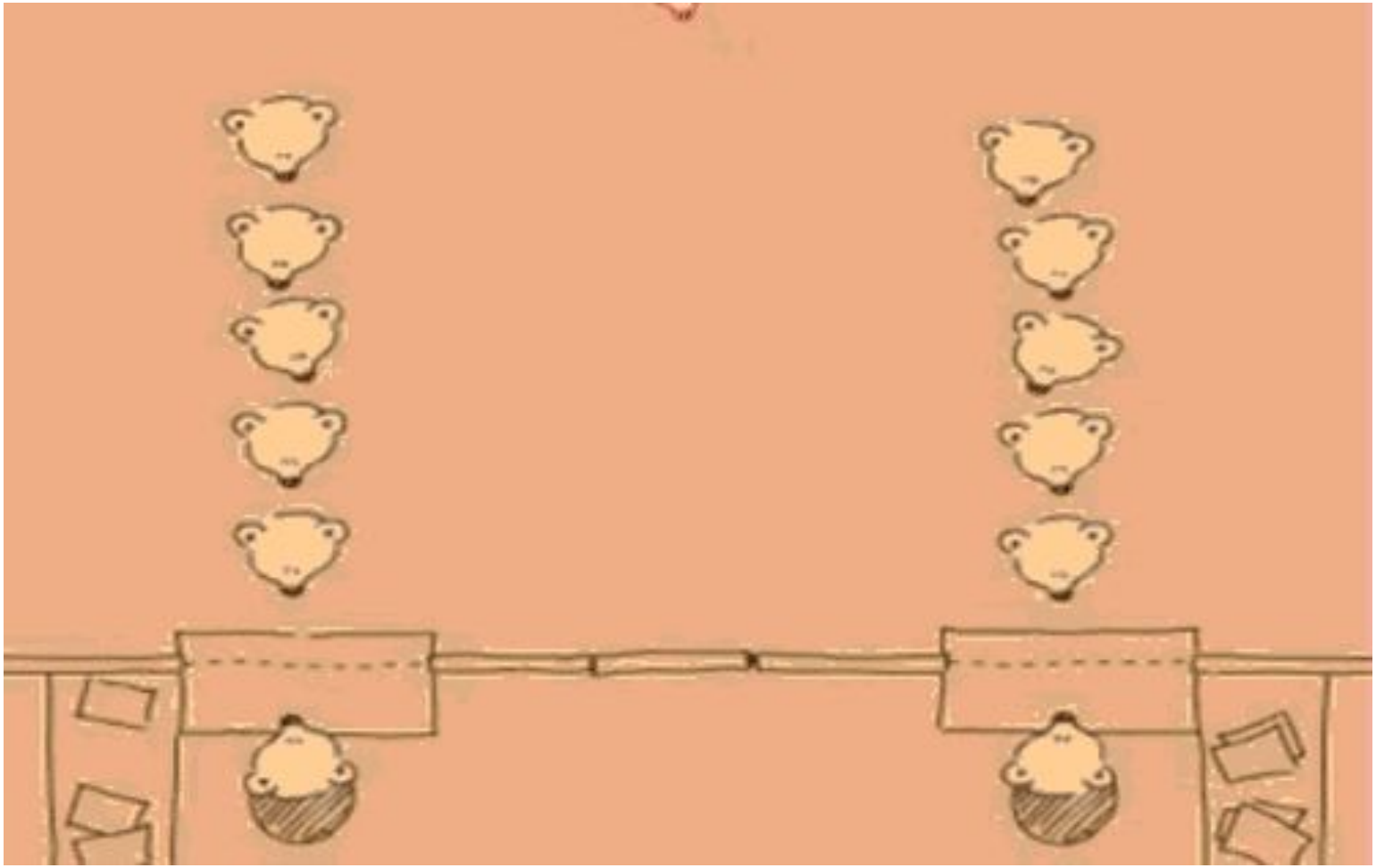
Planificación



Definición

- El término **planificación de procesos** hace referencia a un conjunto de políticas y mecanismos del SO que gobiernan el orden en que se ejecutan los procesos
- Un **planificador de procesos** es un módulo del SO que se encarga de mover los procesos entre las distintas colas de planificación
- La ejecución de un proceso consiste en una alternancia entre **ráfagas de CPU** y **ráfagas de E/S**
- Un proceso limitado por E/S (*I/O bound*) es aquél que pasa más tiempo haciendo E/S que usando la CPU (tiene ráfagas de CPU cortas)
- Un proceso limitado por CPU (*CPU bound*) es aquél que pasa más tiempo procesando que haciendo E/S (tiene ráfagas de CPU largas)

Planificador

- Escoge un proceso de entre los que están en memoria listos para ejecutarse y le asigna la CPU al proceso elegido
- La decisión de planificación puede ocurrir:
  1. Cuando un proceso deja la CPU **voluntariamente**
  2. Cuando un proceso entra a la cola de Listos
- Un planificador es ***no expropiativo o sin desalojo*** (non preemptive) cuando sólo planifica en el caso 1
- Si contempla ambos, decimos que el planificador es ***expropiativo o con desalojo*** (preemptive)

Dispatcher

- El dispatcher es un módulo que cede la CPU al proceso elegido por el planificador de CPU. Para ello el dispatcher tiene que:
  - Realizar una **conmutación de contexto**
  - Cambiar la máquina a **modo usuario** (no privilegiado)
  - **Saltar al punto apropiado** del programa para continuar con su ejecución
- El tiempo que tarda el dispatcher en detener un proceso y poner otro en ejecución se denomina *latencia del dispatcher*.
  - Debe ser lo más pequeña posible

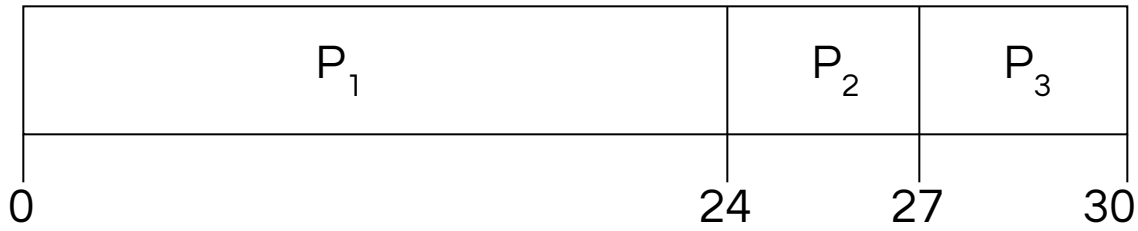
Criterios de planificación

- **Utilización de la CPU** – mantener la CPU tan ocupada como sea posible (maximizar)
- **Rendimiento** – número de procesos que se completan por unidad de tiempo (maximizar)
- **Tiempo de retorno** – tiempo transcurrido desde que se presenta el proceso hasta que se completa (minimizar)
- **Tiempo de espera** – tiempo que un proceso pasa en la cola de procesos listos esperando la CPU (minimizar)
- **Tiempo de respuesta** – tiempo que tarda un proceso desde que se le presenta una solicitud hasta que produce la primera respuesta (minimizar / hacerlo previsible)

FCFS / FIFO

Procesos	Ráfaga de CPU (ms)
P_1	24
P_2	3
P_3	3

- Los procesos llegan en el orden: P_1, P_2, P_3. La planificación es:



- Tiempo de espera para P_1 = 0; P_2 = 24; P_3= 27
- Tiempo de espera medio: (0 + 24 + 27)/3 = 17
- ¿Qué hubiese sucedido si llegaban en otro orden?

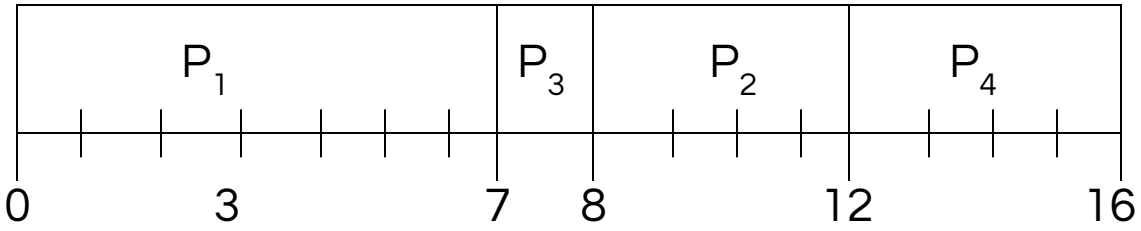
SJF – Shortest job first

- También conocido como Shortest Remaining Time Next (SRTN)
- Asigna la CPU al proceso cuya siguiente ráfaga de CPU es más corta. Si dos procesos empatan, se resuelve el empate por FCFS.
- Dos posibilidades:
  - No expropiativo – cuando se asigna la CPU a un proceso no se puede expropiar hasta que completa su ráfaga de CPU
  - Expropiativo – si llega un proceso a la cola de listos con una ráfaga de CPU más corta que el tiempo restante, se expropia. El SJF expropiativo se conoce también como Shortest Remaining Time First (SRTF)
- SJF es óptimo – da el mínimo tiempo de espera medio para un conjunto de procesos dado.
- Requiere conocer de antemano la duración de la siguiente ráfaga de CPU
- Puede producir starvation

SJF – Shortest job first

Procesos	Llegada	Ráfaga CPU (ms)
P_1	0	7
P_2	2	4
P_3	4	1
P_4	5	4

- **SJF (no expropiativo)**

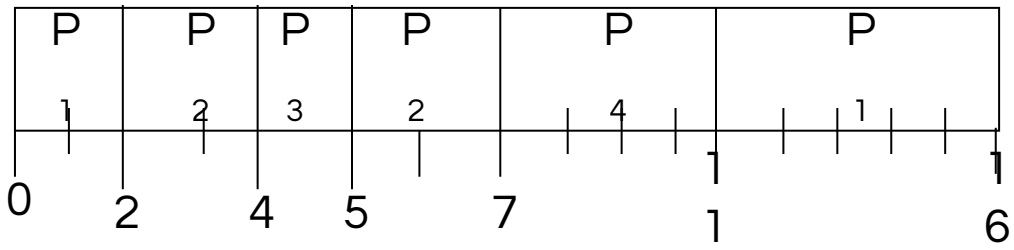


- Tiempo de espera medio = (0 + 6 + 3 + 7)/4 = 4

SJF – Shortest job first

Procesos	Llegada	Ráfaga CPU (ms)
P_1	0	7
P_2	2	4
P_3	4	1
P_4	5	4

- **SJF (expropiativo)**



- Tiempo de espera medio = (9 + 1 + 0 +2)/4 = 3


NO LO SE RICK, PARECE FALSO
memegenerator.net

SJF - Estimación

- Lo habitual es que no se conozca, así que sólo se puede estimar
- Se hace usando la duración de las ráfagas de CPU anteriores, usando un promedio exponencial

1. t_n = longitud de la n – ésima ráfaga de CPU
2. τ_{n+1} = valor predicho para la siguiente ráfaga de CPU
3. $\alpha, 0 \leq \alpha \leq 1$
4. Expresión :

$$\tau_{n+1} = \alpha\, t_n + (1-\alpha)\tau_n.$$


IF YOU WANT MY FUTURE, FORGET MY PAST.
VERSWAND
×
SPICEGIRLS

Prioridades

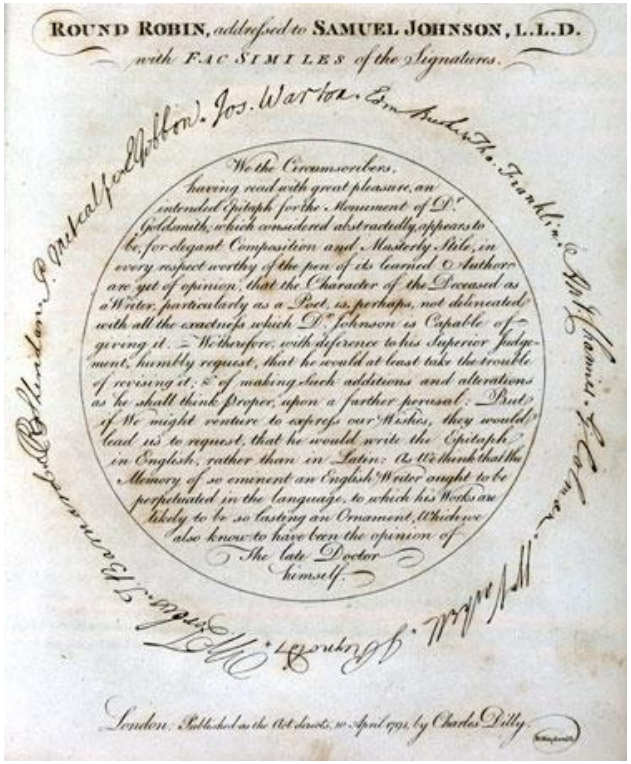
- Se asocia con cada proceso una prioridad (número entero)
- La CPU se asigna al proceso con la prioridad más alta (consideramos número pequeño ≡ prioridad alta)
- Tenemos dos posibilidades:
  - Expropiativo
  - No expropiativo
- SJF se puede ver como un algoritmo de planificación por prioridad en el que la prioridad es la duración predicha para la siguiente ráfaga de CPU
- Problema: Inanición (*starvation*) – los procesos de más baja prioridad podrían no ejecutarse nunca
- Solución: Envejecimiento (*aging*) – conforme el tiempo pasa aumentar la prioridad de los procesos que esperan mucho en el sistema

HRRN

- Es una adaptación del SJF que permite romper la inanición.
- Su nombre, Highest Response Ratio Next, indica su comportamiento. Se prioriza a los procesos con mayor Response Ratio.
- R.R. = (S + W) / S
  - S = Tiempo de servicio (próxima ráfaga)
  - W = Espera
- Cuanto mayor sea la espera, respecto al tamaño de su ráfaga, mayor será su prioridad para ejecutar

Round Robin

- Cada proceso obtiene la CPU durante un breve espacio de tiempo (*cuanto o quantum de tiempo*). Cuando el tiempo pasa, el proceso es expropiado e insertado al final de la cola de listos.
- Si hay *n* procesos en la cola de listos y el quantum es *q*, cada proceso recibe 1/*n* del tiempo de CPU en intervalos de *q* unidades de tiempo como mucho. Ningún proceso espera más de (*n*-1)*q* unidades de tiempo.
- ¿Qué tamaño debería tener el quantum?



Otros algoritmos

- Virtual Round Robin
- Colas multinivel
  - Feedback / Realimentación
  - Prioridades absolutas

Preguntas?