

AYUDANTÍA 2: SCHEDULING

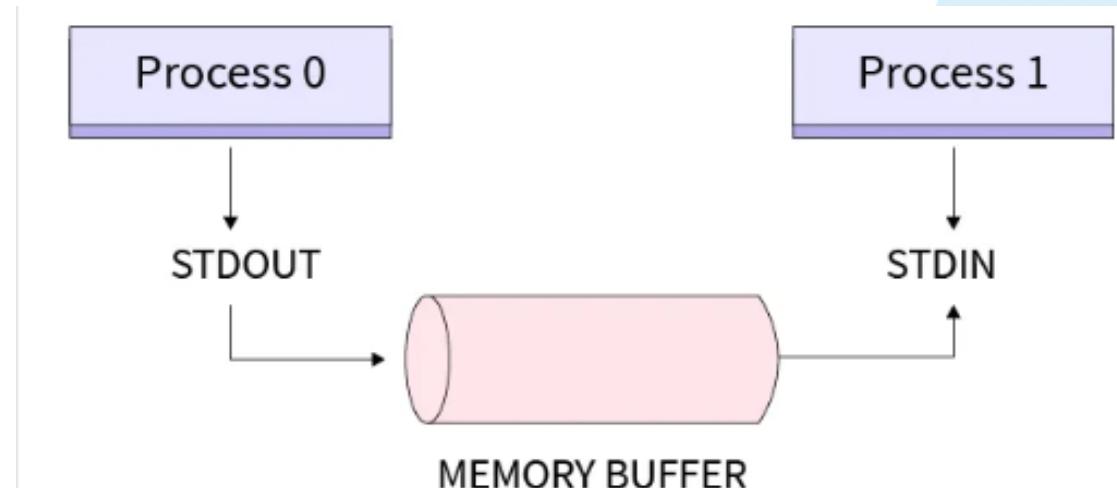


**Profesor : Jonathan Frez
Ayudante : Dante Hortuvia**

REGLAS DE LAS TAREAS



- Windows queda estrictamente prohibido en cualquier tipo de clase, ejercicio, tarea y cualquier cosa relacionada al curso
- El uso de IA generativa esta permitido, siempre y cuando no se utilice de manera abusiva
- Para la tarea 1 se deben utilizar **Named Pipes**



Scheduling

Es un proceso que realiza el sistema operativo para definir el orden a ejecutar distintos procesos

- Este ayuda a que múltiples tareas puedan funcionar de manera eficiente en el sistema
- Este busca garantizar que los recursos del sistema se usen de la manera mas efectiva

Como el sistema Operativo decide que proceso ejecutar?



Scheduling

La multiprogramación es un concepto fundamental en los sistemas operativos, ya que optimiza el uso de la CPU al mantener varios procesos en memoria. De esta manera, la CPU puede asignarse dinámicamente a aquellos procesos que estén listos para ejecutarse, evitando tiempos muertos y mejorando la eficiencia del sistema

El encargado de administrar esta tarea es el CPU Scheduler, el cual selecciona el proceso que se ejecutará en un momento dado. Para ello, utiliza diferentes algoritmos que definen la asignación más adecuada de la CPU.

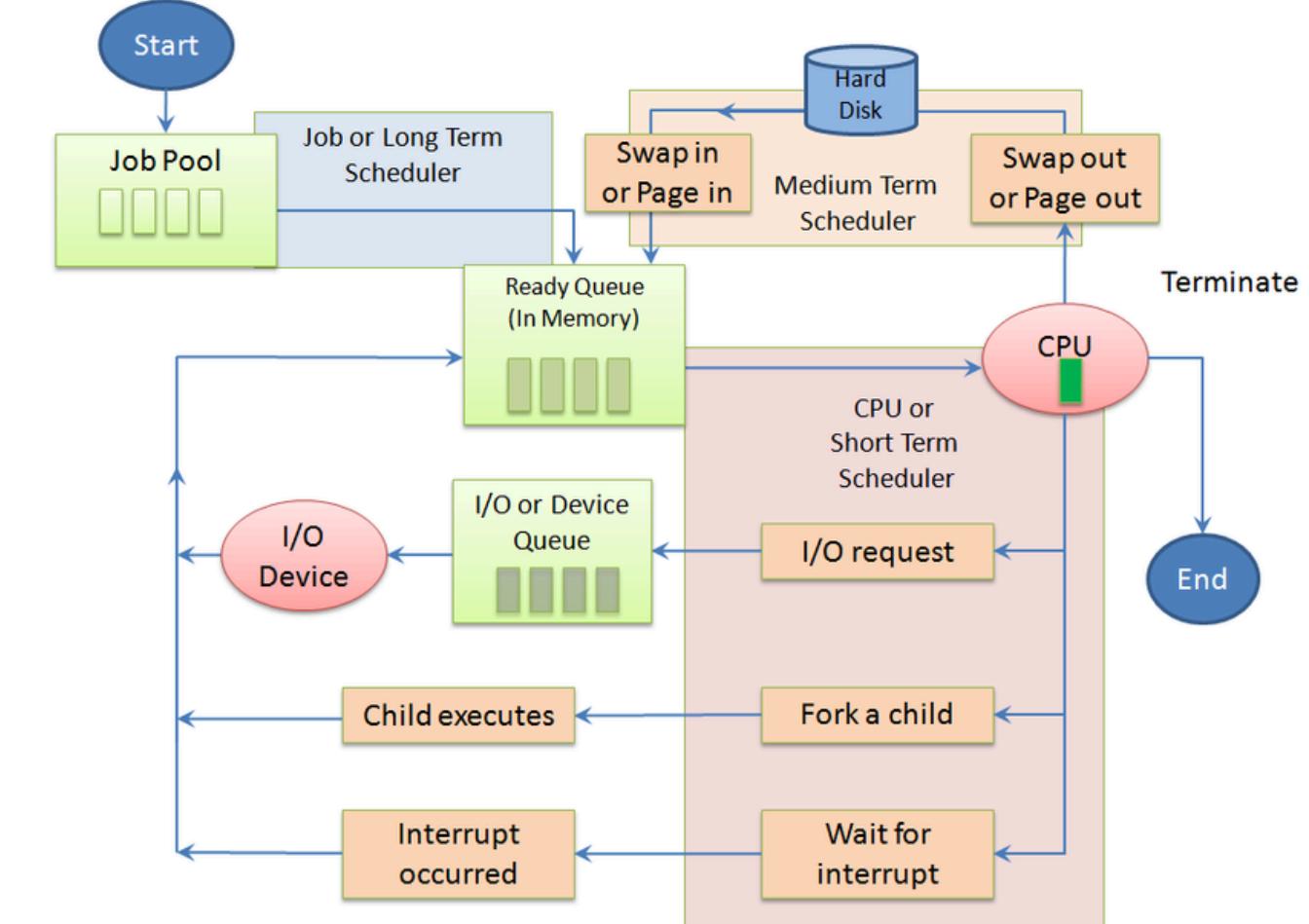
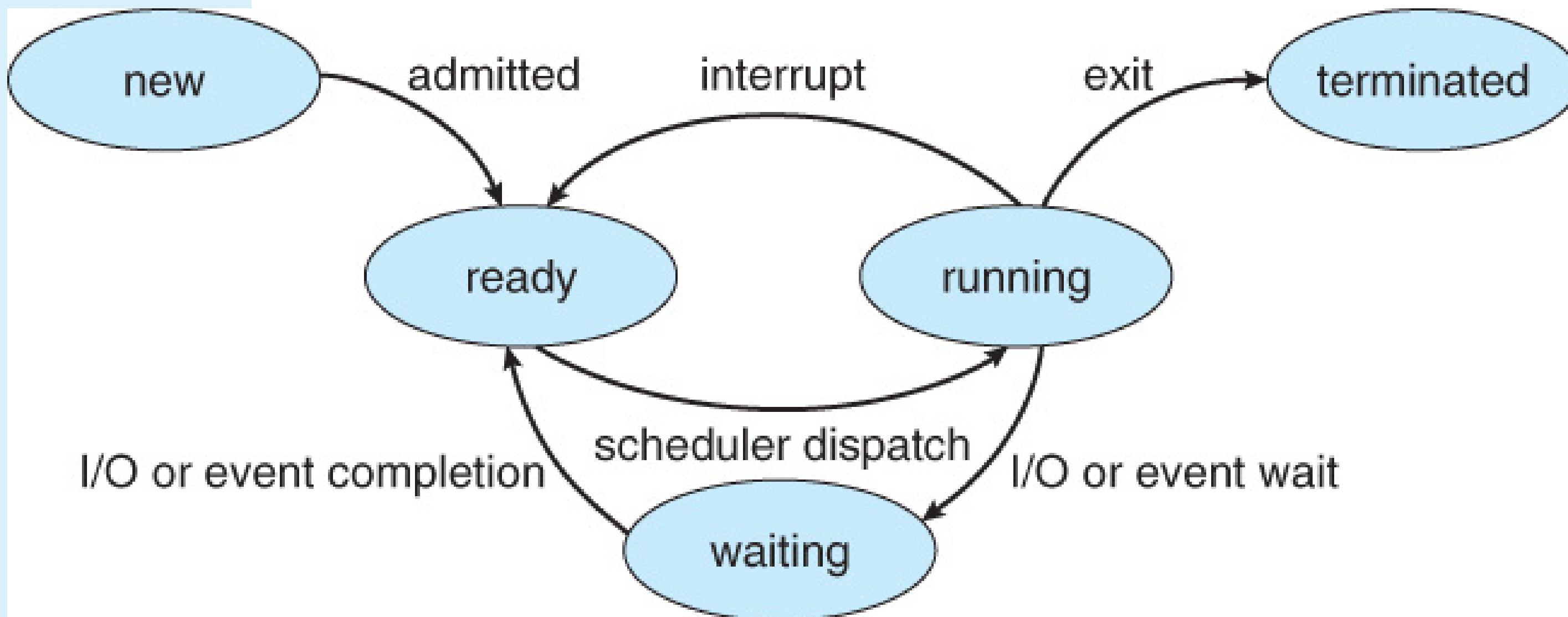


Diagrama de Scheduling



Recordando Organizacion de los procesos:

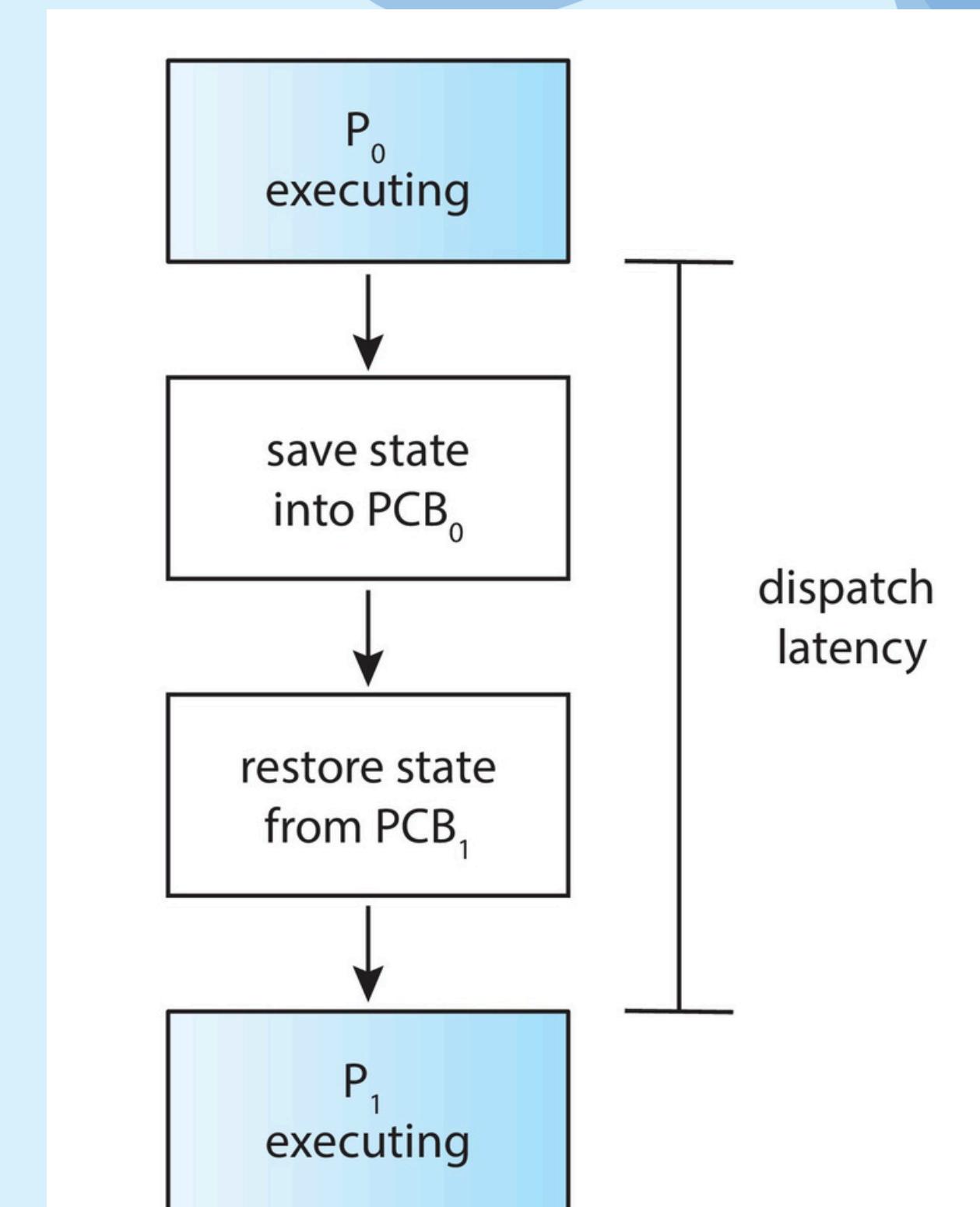


Dispatcher

El dispatcher es un componente esencial en la administración de procesos de un sistema operativo. Una vez que el CPU Scheduler selecciona qué proceso debe ejecutarse, el dispatcher se encarga de poner en marcha esa decisión.

Su función principal es realizar el cambio de contexto, lo que implica guardar el estado del proceso que estaba ejecutándose y cargar el estado del nuevo proceso. Además, transfiere el control de la CPU al proceso elegido para que pueda comenzar o continuar su ejecución.

El tiempo que tarda en completar este procedimiento se denomina latencia del dispatcher, y mientras más bajo sea, mayor será la eficiencia del sistema, ya que se reduce el tiempo improductivo en la CPU.



Tipos de Scheduler



Long term scheduler

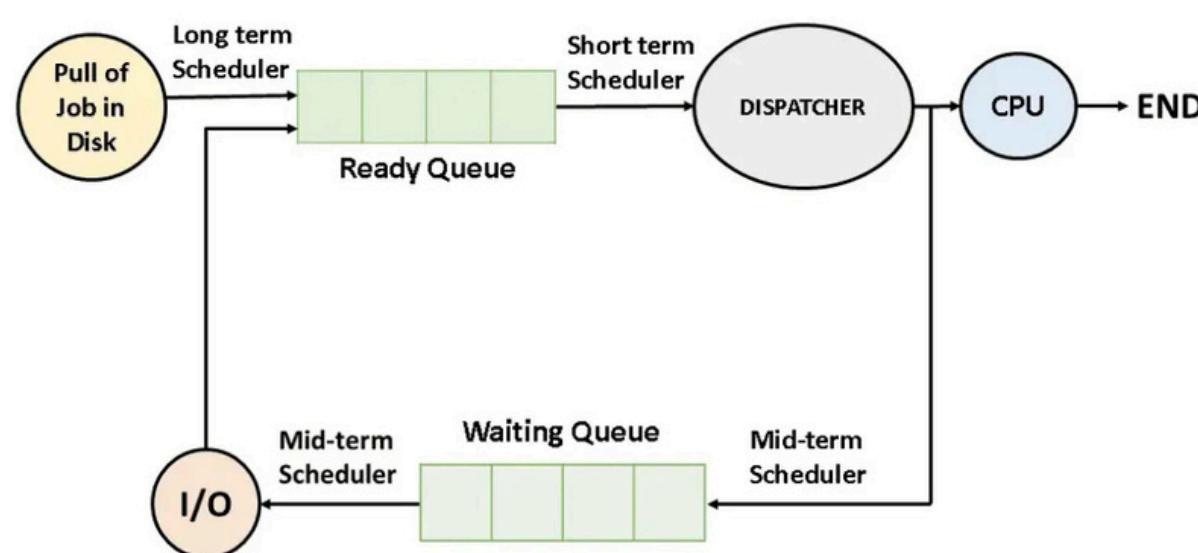
Procesos que quedan en la ready queue, esto define la cantidad también.

Medium term scheduler

Modifica temporalmente el grado de multiprogramación y hace swapping (RAM a HDD y viceversa).

Short term scheduler

Envía procesos de la ready queue a CPU, realiza cambio de contexto.



Tipos de Scheduling

Expropiativo:

- El sistema operativo puede suspender un proceso en ejecución y reasignar la CPU a otro con mayor prioridad.

No expropiativo:

- Una vez que un proceso obtiene la CPU, no puede ser interrumpido por el sistema operativo; continúa hasta finalizar o entrar en espera por una operación de E/S.

Batch Scheduling:

- Agrupa procesos en lotes y los ejecuta de forma secuencial, buscando maximizar la eficiencia global del sistema.

Interactive Scheduling:

- Orientado a sistemas donde la interacción con el usuario es fundamental, asegurando respuestas rápidas y fluidas.

Real-Time Scheduling:

- Diseñado para garantizar que los procesos se ejecuten dentro de plazos estrictos y predefinidos, siendo crítico en sistemas de tiempo real.

Algoritmos de Scheduling

No-Expropiativos

First come first served (Batch)
Shortes job first (Batch)

Expropiativos

Earliest deadline first (Real Time)
Round Robin (Interactive)
Priority (Interactive)

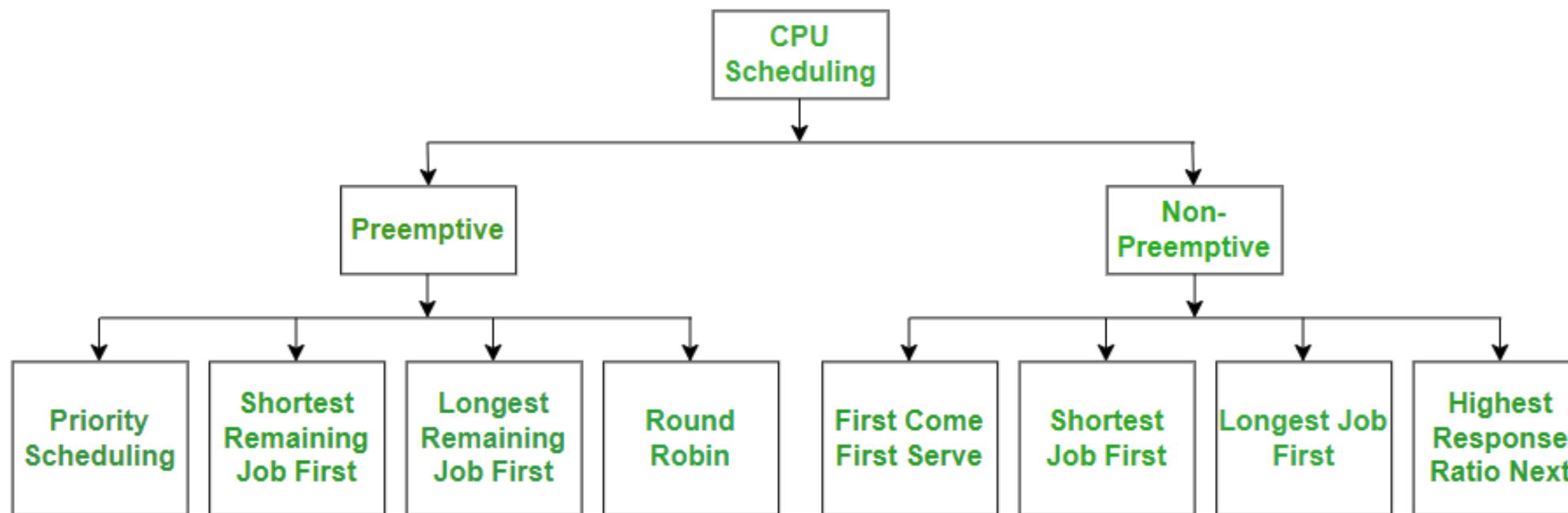


Diagrama Algoritmos de Scheduler.

Métricas de planificación de procesos

- Arrival Time: Momento en que el proceso llega a la ready queue.
- CPU Burst: Tiempo que un proceso requiere de la CPU para ejecutarse.
- End Time: Instante en que el proceso finaliza su ejecución.
- Turnaround Time: Tiempo total de permanencia de un proceso en el sistema, desde su llegada hasta su finalización. ($\text{Turnaround} = \text{End Time} - \text{Arrival Time}$)
- Execution Start: Momento en que el proceso comienza a ejecutarse en la CPU.
- Response Time: Tiempo que transcurre desde la llegada del proceso hasta que inicia su primera ejecución en la CPU. ($\text{Response} = \text{Execution Start} - \text{Arrival Time}$)

Ejercicios



Ejercicio 1:

Considere la siguiente tabla de proceso::

Proceso	CPU Burst	Arrival Time
P_0	3	7
P_1	10	1
P_2	3	0
P_3	6	5



Determinar el Turnaround y end time, considerando para ello el algoritmo Short Job First no expropiativo y Round-Robin con un quantum de 2.

Ejercicio 2:

Considere la siguiente tabla de proceso:

Proceso	Arrival Time	CPU Burst	Deadline
P_0	1	2	28
P_1	6	4	27
P_2	2	7	35
P_3	0	6	32
P_4	7	5	23

Para cada uno de los procesos, determinar el Turnaround y end time, considerando para ello el algoritmo Round-Robin con un quantum de 3.



Ejercicio 3:

Las oficinas de los académicos fueron remodeladas y ahora son completamente de vidrio. Un académico de la EITle tiene un miedo patológico al vidrio. Por ello, es que el académico necesita organizarse de modo de pasar el menor tiempo posible en su nueva oficina y el mayor tiempo posible en algún laboratorio que están muchas veces ocupados por otros profes malévolos de la EIT que lo usan para trabajar con sus alumnos o avanzar en investigación. La tabla de uso de los laboratorios son las siguientes:

Lab 1 :

Profesor	Tiempo de llegada	Tiempo de lab	Prioridad
P1	60	120	90
P2	120	180	110
P3	90	60	40
VF	0	240	0

Lab 2:

Profesor	Tiempo de llegada	Tiempo de lab	Prioridad
P1	60	30	25
P2	120	45	55
P3	90	180	80
VF	0	240	0

- Suponiendo que la reserva de los laboratorios es expropiativa, indique qué laboratorio debiese preferir el profesor VidrioFóbico (VF) si es que prima que no deba salir y volver al mismo laboratorio (prima la continuidad de permanencia en el laboratorio elegido).
- ¿Podría el profesor VF escapar totalmente del vidrio refugiándose en ambos laboratorios? Explique.
- De ser cierta la respuesta b), ¿qué debiesen hacer los otros profesores para forzar a que el profesor VF volviese a su oficina? De ser falsa la respuesta a la parte b), ¿Qué debe hacer el profesor VF para poder escapar totalmente de su oficina de vidrio?

Ejercicio 4:

Supongamos que hay tres grupos de personas realizando un experimento en un laboratorio donde solo hay un computador disponible. Cada grupo necesita utilizar este computador durante cierto tiempo para completar su experimento . Aquí están los datos de tiempo que cada grupo necesita para su experimento:

- Grupo 1: 300 minutos
- Grupo 2: 250 minutos
- Grupo 3: 330 minutos

Dados estos datos, de acuerdo a las siguientes estrategias, indique el tiempo de término, turnaround, inicio de ejecución y tiempo de respuesta:

a) Round Robin ($q=80\text{min}$). Orden de llegada, Grupo 1: min 0, Grupo 2: min 40, Grupo 3: min 160.

b) Shortest Job First (Expropiativo). Orden de llegada: Grupo 1: min 42, Grupo 2: min 120, Grupo 3: min 0.

c) Priority expropiativo, con las siguientes prioridades, Grupo 1: 30, Grupo 2: 20, Grupo 3: 50. Orden de llegada: Grupo 1: min 100, Grupo 2: min 0, Grupo 3: min 50.



Ejercicio 5:

La Reserva Natural KH organiza visitas de investigadores a sus estaciones de monitoreo. Cada visita tiene:

- Un tiempo de llegada (momento en que se solicita el acceso)
- Una duración estimada (tiempo para realizar la visita)
- Prioridad asignada (un número entero donde un valor menor indica mayor prioridad)

Para optimizar los recursos, la reserva cuenta con dos equipos que pueden trabajar en paralelo, procesando las visitas simultáneamente. Es importante destacar que cada vez que un equipo está libre, se le asigna un grupo de visitas y por otro lado las visitas no pueden ser interrumpidas.

Considere las siguientes visitas con sus tiempos de llegada, duración y prioridad:

En base a lo anterior, realice la planificación completa de los equipos, indicando: tiempo de inicio, tiempo de término, tiempo de espera y tiempo total en sistema (desde llegada hasta término).

Visita	Tiempo de llegada	Duración	Prioridad
v1	0	20	3
v2	15	20	2
v3	40	40	4
v4	30	55	8
v5	50	10	1
v6	40	70	5
v7	15	30	4

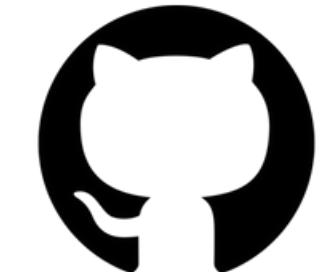
Contacto



DANTE.HORTUVIA@MAIL.UDP.CL



+56 9 2236 9606



[HTTPS://GITHUB.COM/DOSHUERTOS/AYUDANTIAS](https://github.com/doshuertos/ayudantias)

SO_2_2025.GIT



Gracias

