

# Sistemas Operativos

## Crterios y algoritmos de planificación

Departamento de Ingeniería en Sistemas y Computación  
Universidad Católica del Norte, Antofagasta.

En la medida que un proceso se ejecuta, va cambiando de estado. Estos estados son:

- new: el proceso es creado
- running: las instrucciones están siendo ejecutadas
- waiting: el proceso espera que ocurra algún evento
- ready: el proceso espera que se le asigne un procesador
- terminated: el proceso terminó su ejecución

En la medida que un proceso se ejecuta, va cambiando de estado. Estos estados son:

- **new**: el proceso es creado
- **running**: las instrucciones están siendo ejecutadas
- **waiting**: el proceso espera que ocurra algún evento
- **ready**: el proceso espera que se le asigne un procesador
- **terminated**: el proceso terminó su ejecución

En la medida que un proceso se ejecuta, va cambiando de estado. Estos estados son:

- new: el proceso es creado
- running: las instrucciones están siendo ejecutadas
- waiting: el proceso espera que ocurra algún evento
- ready: el proceso espera que se le asigne un procesador
- terminated: el proceso terminó su ejecución

En la medida que un proceso se ejecuta, va cambiando de estado. Estos estados son:

- new: el proceso es creado
- running: las instrucciones están siendo ejecutadas
- waiting: el proceso espera que ocurra algún evento
- ready: el proceso espera que se le asigne un procesador
- terminated: el proceso terminó su ejecución

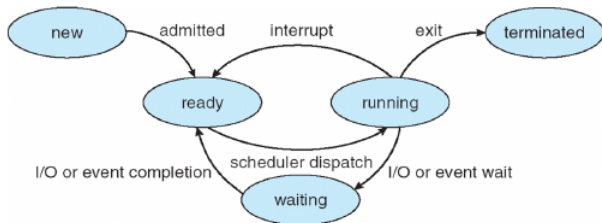
En la medida que un proceso se ejecuta, va cambiando de estado. Estos estados son:

- new: el proceso es creado
- running: las instrucciones están siendo ejecutadas
- waiting: el proceso espera que ocurra algún evento
- ready: el proceso espera que se le asigne un procesador
- terminated: el proceso terminó su ejecución

En la medida que un proceso se ejecuta, va cambiando de estado. Estos estados son:

- new: el proceso es creado
- running: las instrucciones están siendo ejecutadas
- waiting: el proceso espera que ocurra algún evento
- ready: el proceso espera que se le asigne un procesador
- terminated: el proceso terminó su ejecución

# Diagrama de estados





El SO utiliza las siguientes colas para la itineración de procesos:

- Cola de jobs: conjunto de todos los procesos del sistema
- Cola ready: conjunto de todos los procesos que están en memoria principal listos y esperando ser ejecutados
- Cola de dispositivos: procesos que esperan por E/S

Los procesos se mueven entre estas colas en su ciclo de vida.

El SO utiliza las siguientes colas para la itineración de procesos:

- Cola de jobs: conjunto de todos los procesos del sistema
- Cola ready: conjunto de todos los procesos que están en memoria principal listos y esperando ser ejecutados
- Cola de dispositivos: procesos que esperan por E/S

Los procesos se mueven entre estas colas en su ciclo de vida.

El SO utiliza las siguientes colas para la itineración de procesos:

- Cola de jobs: conjunto de todos los procesos del sistema
- Cola ready: conjunto de todos los procesos que están en memoria principal listos y esperando ser ejecutados
- Cola de dispositivos: procesos que esperan por E/S

Los procesos se mueven entre estas colas en su ciclo de vida.

El SO utiliza las siguientes colas para la itineración de procesos:

- Cola de jobs: conjunto de todos los procesos del sistema
- Cola ready: conjunto de todos los procesos que están en memoria principal listos y esperando ser ejecutados
- Cola de dispositivos: procesos que esperan por E/S

Los procesos se mueven entre estas colas en su ciclo de vida.

El SO utiliza las siguientes colas para la itineración de procesos:

- Cola de jobs: conjunto de todos los procesos del sistema
- Cola ready: conjunto de todos los procesos que están en memoria principal listos y esperando ser ejecutados
- Cola de dispositivos: procesos que esperan por E/S

Los procesos se mueven entre estas colas en su ciclo de vida.

Existen dos tipos:

- De largo plazo (job scheduler): selecciona cual proceso se debe llevar a la cola ready. Puede que en algunos SO este scheduler no exista.
- De corto plazo (CPU scheduler): selecciona el proceso a ejecutar a continuación y asigna CPU.

Existen dos tipos:

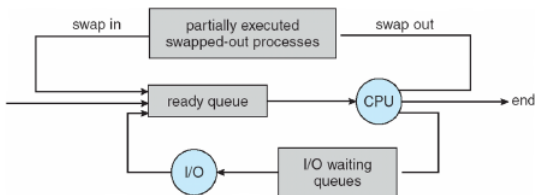
- De largo plazo (job scheduler): selecciona cual proceso se debe llevar a la cola ready. Puede que en algunos SO este scheduler no exista.
- De corto plazo (CPU scheduler): selecciona el proceso a ejecutar a continuación y asigna CPU.





# Itinerador de mediano plazo

- Este esquema se denomina swapping
- Reduce el grado de multiprogramación (jobs en memoria principal)



- El itinerador de corto plazo es invocado frecuentemente (orden de milisegundos). Debe ser rápido.
- El itinerador de largo plazo se invoca en forma poco frecuente (orden de segundos o minutos). Puede ser lento
- El itinerador de mediano plazo controla el grado de multiprogramación

- El itinerador de corto plazo es invocado frecuentemente (orden de milisegundos). Debe ser rápido.
- El itinerador de largo plazo se invoca en forma poco frecuente (orden de segundos o minutos). Puede ser lento
- El itinerador de mediano plazo controla el grado de multiprogramación

- El itinerador de corto plazo es invocado frecuentemente (orden de milisegundos). Debe ser rápido.
- El itinerador de largo plazo se invoca en forma poco frecuente (orden de segundos o minutos). Puede ser lento
- El itinerador de mediano plazo controla el grado de multiprogramación

Los procesos se pueden clasificar en dos tipos:

- Limitados por E/S: pasan más tiempo haciendo E/S que cálculos. Muchas ráfagas cortas de CPU.
- Limitados por CPU: pasan más tiempo haciendo cálculos. Pocas ráfagas largas de CPU.

Los procesos se pueden clasificar en dos tipos:

- Limitados por E/S: pasan más tiempo haciendo E/S que cálculos. Muchas ráfagas cortas de CPU.
- Limitados por CPU: pasan más tiempo haciendo cálculos. Pocas ráfagas largas de CPU.

Los procesos se pueden clasificar en dos tipos:

- Limitados por E/S: pasan más tiempo haciendo E/S que cálculos. Muchas ráfagas cortas de CPU.
- Limitados por CPU: pasan más tiempo haciendo cálculos. Pocas ráfagas largas de CPU.

## Algoritmos utilizados:

- FCFS: First Come First Served, se ejecuta primero el que llega primero.
- SJF: Shortest Job First, se ejecuta primero el proceso de duración más corta.
- Prioridad: Los procesos pueden tener prioridad especial por sobre los demás.



## Algoritmos utilizados:

- FCFS: First Come First Served, se ejecuta primero el que llega primero.
- SJF: Shortest Job First, se ejecuta primero el proceso de duración más corta.
- Prioridad: Los procesos pueden tener prioridad especial por sobre los demás.

## Algoritmos utilizados:

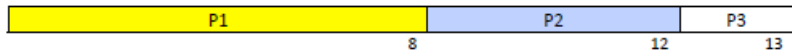
- FCFS: First Come First Served, se ejecuta primero el que llega primero.
- SJF: Shortest Job First, se ejecuta primero el proceso de duración más corta.
- Prioridad: Los procesos pueden tener prioridad especial por sobre los demás.

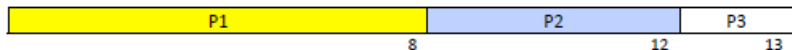
# Ejemplo

Considere los procesos P1, P2 y P3 con sus respectivos tiempos de llegada y tiempo que toma su ejecución (tiempo de ráfaga).

Proceso	Tiempo de llegada	"Burst Time"
P1	0.0	8
P2	0.4	4
P3	1.0	1

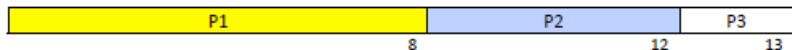
La carta gantt de los procesos, al utilizar FCFS es:





Cantidades de interés:

- Tiempo espera promedio:  $(0+8+12)/3$
- Tiempo de ejecución promedio:  $(8 + (12-0.4) + (13-1) )/3$



Cantidades de interés:

- Tiempo espera promedio:  $(0+8+12)/3$
- Tiempo de ejecución promedio:  $(8 + (12-0.4) + (13-1) )/3$

- Calcule el tiempo de espera promedio, y el tiempo de ejecución promedio si se utiliza SJF.
- Cómo cambia esta última respuesta, si la CPU se deja inactiva la primera unidad de tiempo?

Considere el grupo de procesos P1, P2, P3, P4 y P5, llegando todos en el tiempo  $t = 0$ .

Proceso	"Burst Time"	Prioridad
P1	5	4
P2	3	3
P3	10	1
P4	1	4
P5	1	2

Dibuje la carta gantt, calcule el tiempo promedio de espera, y el tiempo de ejecución promedio al utilizar SJF, o Prioridad.