

Gestió processos

Josep Vilaplana

ETSEIB/GIE

22 de maig de 2023

① CONCEPTE DE PROCÉS

Un procés és un cas de programa en execució. És l'objecte que el sistema operatiu utilitza per gestionar l'ús de la C.P.U.

② FUNCIONS DEL S.O. PER GESTIONAR ELS PROCESOS:

- Creació i destrucció dels processos.
- Control del progrés dels processos: Cal assegurar que progressin cap a la seva conclusió a un ritme raonable.
- Actuar sobre les condicions excepcionals que surten durant l'execució d'un procés (interrupcions, errors aritmètics, etc).
- Repartir els recursos de hardware entre els processos.

① CONCEPTE DE PROCÉS

Un procés és un cas de programa en execució. És l'objecte que el sistema operatiu utilitza per gestionar l'ús de la C.P.U.

② FUNCIONS DEL S.O. PER GESTIONAR ELS PROCESOS:

- Creació i destrucció dels processos.
- Control del progrés dels processos: Cal assegurar que progressin cap a la seva conclusió a un ritme raonable.
- Actuar sobre les condicions excepcionals que surten durant l'execució d'un procés (interrupcions, errors aritmetics, etc).
- Repartir els recursos de hardware entre els processos.

① CONCEPTE DE PROCÉS

Un procés és un cas de programa en execució. És l'objecte que el sistema operatiu utilitza per gestionar l'ús de la C.P.U.

② FUNCIONS DEL S.O. PER GESTIONAR ELS PROCESOS:

- Creació i destrucció dels processos.
- Control del progrés dels processos: Cal assegurar que progressin cap a la seva conclusió a un ritme raonable.
- Actuar sobre les condicions excepcionals que surten durant l'execució d'un procés (interrupcions, errors aritmetics, etc).
- Repartir els recursos de hardware entre els processos.

① CONCEPTE DE PROCÉS

Un procés és un cas de programa en execució. És l'objecte que el sistema operatiu utilitza per gestionar l'ús de la C.P.U.

② FUNCIONS DEL S.O. PER GESTIONAR ELS PROCESOS:

- Creació i destrucció dels processos.
- Control del progrés dels processos: Cal assegurar que progressin cap a la seva conclusió a un ritme raonable.
- Actuar sobre les condicions excepcionals que surten durant l'execució d'un procés (interrupcions, errors aritmètics, etc).
- Repartir els recursos de hardware entre els processos.

① CONCEPTE DE PROCÉS

Un procés és un cas de programa en execució. És l'objecte que el sistema operatiu utilitza per gestionar l'ús de la C.P.U.

② FUNCIONS DEL S.O. PER GESTIONAR ELS PROCESSOS:

- Creació i destrucció dels processos.
- Control del progrés dels processos: Cal assegurar que progressin cap a la seva conclusió a un ritme raonable.
- Actuar sobre les condicions excepcionals que surten durant l'execució d'un procés (interrupcions, errors aritmetics, etc).
- Repartir els recursos de hardware entre els processos.

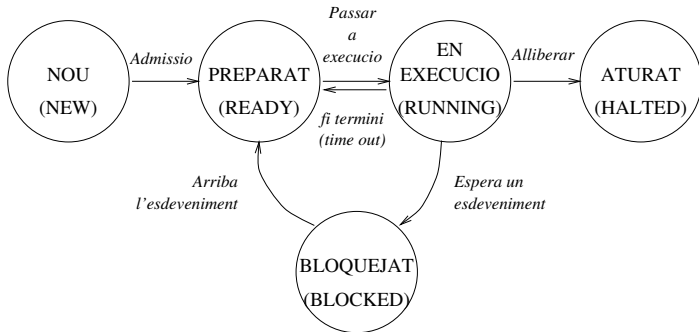
① CONCEPTE DE PROCÉS

Un procés és un cas de programa en execució. És l'objecte que el sistema operatiu utilitza per gestionar l'ús de la C.P.U.

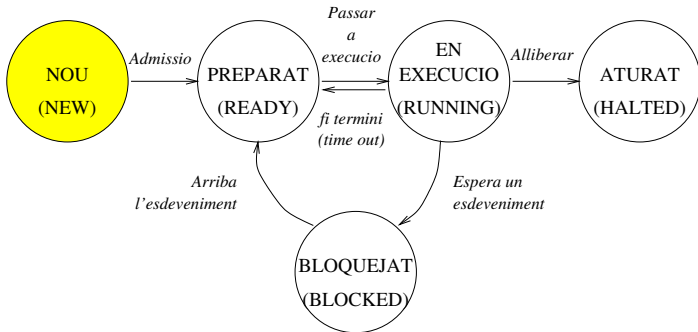
② FUNCIONS DEL S.O. PER GESTIONAR ELS PROCESSOS:

- Creació i destrucció dels processos.
- Control del progrés dels processos: Cal assegurar que progressin cap a la seva conclusió a un ritme raonable.
- Actuar sobre les condicions excepcionals que surten durant l'execució d'un procés (interrupcions, errors aritmetics, etc).
- Repartir els recursos de hardware entre els processos.

Estats d'un procés (I)

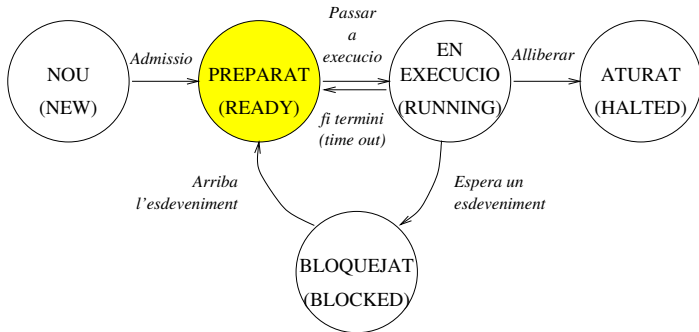


Estats d'un procés (I)



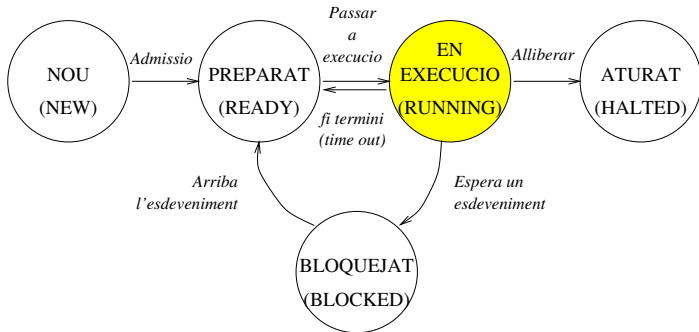
- **NOU.** El procés està creant-se per poder ser gestionat pel sistema operatiu. El procés podria no ser admés per limitacions de memòria, per motius de rendiment del sistema o altres motius.

Estats d'un procés (I)



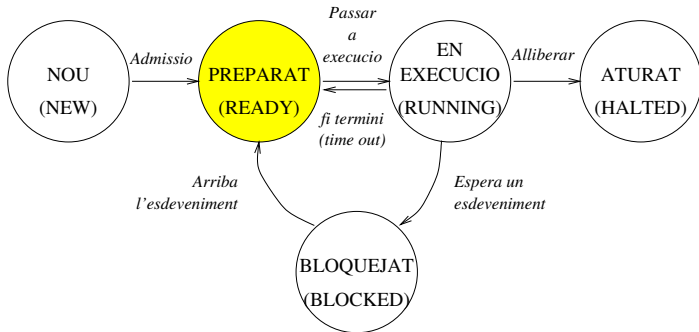
- **PREPARAT.** El procés té totes les condicions per ser executat.

Estats d'un procés (I)



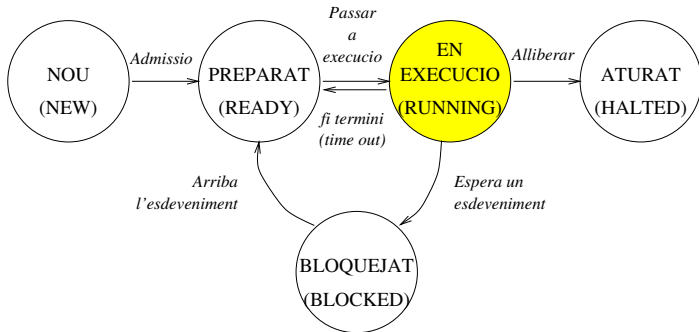
- **EN EXECUCIÓ.** El procés està executant-se.

Estats d'un procés (I)



- **PREPARAT.** El procés té totes les condicions per ser executat.

Estats d'un procés (I)

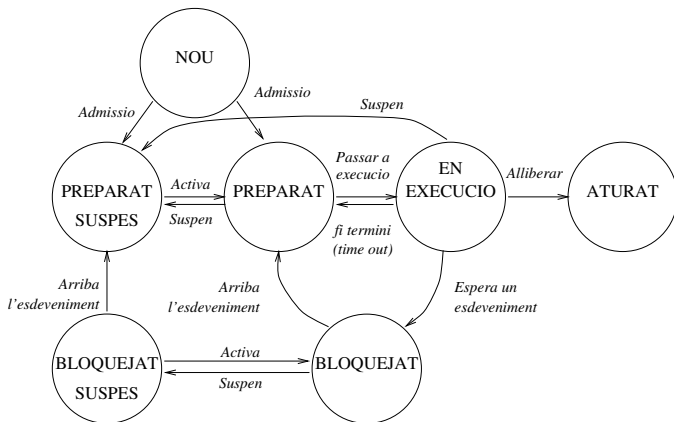


- **EN EXECUCIÓ.** El procés està executant-se.

- Pot passar que un procés no hi càpiga a memòria principal a conseqüència d'altres processos que competeixen per la CPU. El problema es pot solucionar enmagatzamant a la memòria secundària alguns processos que en un instant donat no estan executant-se. La operació de passar processos entre la memòria secundària i la memòria principal s'anomena *intercanvi (swapping)*.
- El disc és el dispositiu més ràpid i adient per ser usat com a memòria secundària.

- Pot passar que un procés no hi càpiga a memòria principal a conseqüència d'altres processos que competeixen per la CPU. El problema es pot solucionar enmagatzamant a la memòria secundària alguns processos que en un instant donat no estan executant-se. La operació de passar processos entre la memòria secundària i la memòria principal s'anomena *intercanvi (swapping)*.
- El disc és el dispositiu més ràpid i adient per ser usat com a memòria secundària.

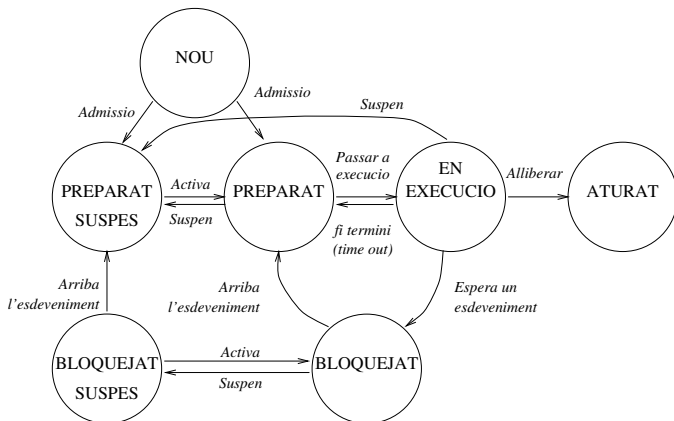
Estats d'un procés (III)



NOUS ESTATS:

- **PREPARAT I SUSPÈS**. Procés preparat en memòria secundària.
- **BLOQUEJAT I SUSPÈS**. Procés bloquejat en memòria secundària.

Estats d'un procés (III)



NOUS ESTATS:

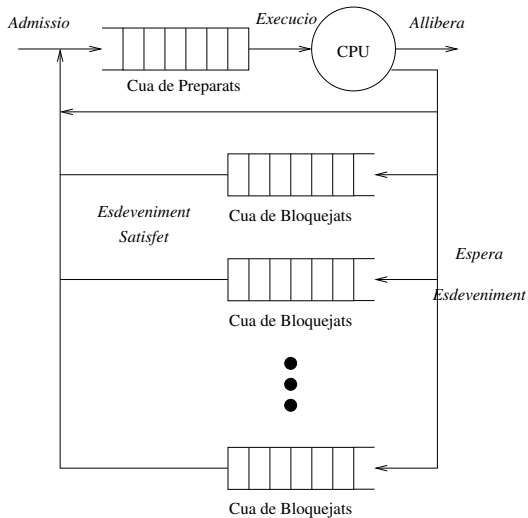
- **PREPARAT I SUSPÈS**. Procés preparat en memòria secundària.
- **BLOQUEJAT I SUSPÈS**. Procés bloquejat en memòria secundària.

IDENTIFICACIÓ DEL PROCÉS Identificador del procés Identificador del procés pare Identificador de l'usuari
ESTAT DEL PROCESSADOR Comptador de programa Registres CPU
CONTROL DEL PROCÉS Estat del procés Prioritat Informació de planificació (temps d'espera/execució, etc) Esdeveniment que espera Privilegis sobre els recursos Gestió Memòria Propietat i ús dels recursos (fixers, memòria, etc)

- Anomenarem *Planificació (Scheduling)* al conjunt d'estratègies i mecanismes contruïts en el sistema operatiu que governen l'ordre en que la feina a fer es realitzarà.
- L'objectiu bàsic del planificador és optimitzar el rendiment del sistema.
- El planificador manté cues de processos segons el seu estat.

- Anomenarem *Planificació (Scheduling)* al conjunt d'estratègies i mecanismes contruïts en el sistema operatiu que governen l'ordre en que la feina a fer es realitzarà.
- L'objectiu bàsic del planificador és optimitzar el rendiment del sistema.
- El planificador manté cues de processos segons el seu estat.

- Anomenarem *Planificació (Scheduling)* al conjunt d'estratègies i mecanismes contruïts en el sistema operatiu que governen l'ordre en que la feina a fer es realitzarà.
- L'objectiu bàsic del planificador és optimitzar el rendiment del sistema.
- El planificador manté cues de processos segons el seu estat.



- *Ús de la CPU*: Fracció mitja del temps en que la CPU està ocupada. Inclou el temps perdut pel sistema operatiu.
- *Productivitat*: Nombre de treballs completats per unitat de temps.
- *Temps de retorn*: L'interval de temps que passa entre l'enviament del treball i la seva finalització. És la suma dels períodes perduts per estar en memòria, esperar en la cua de preparats, executar sobre la CPU i, realitzar les E/S.
- *Temps d'espera*: El temps que un treball espera en la cua de preparats.
- *Temps de resposta*: És el temps entre l'enviament d'una petició i la primera resposta produïda. Una mesura millor pels sistemes interactius.

- *Ús de la CPU*: Fracció mitja del temps en que la CPU està ocupada. Inclou el temps perdut pel sistema operatiu.
- *Productivitat*: Nombre de treballs completats per unitat de temps.
- *Temps de retorn*: L'interval de temps que passa entre l'enviament del treball i la seva finalització. És la suma dels períodes perduts per estar en memòria, esperar en la cua de preparats, executar sobre la CPU i, realitzar les E/S.
- *Temps d'espera*: El temps que un treball espera en la cua de preparats.
- *Temps de resposta*: És el temps entre l'enviament d'una petició i la primera resposta produïda. Una mesura millor pels sistemes interactius.

- *Ús de la CPU*: Fracció mitja del temps en que la CPU està ocupada. Inclou el temps perdut pel sistema operatiu.
- *Productivitat*: Nombre de treballs completats per unitat de temps.
- *Temps de retorn*: L'interval de temps que passa entre l'enviament del treball i la seva finalització. És la suma dels períodes perduts per estar en memòria, esperar en la cua de preparats, executar sobre la CPU i, realitzar les E/S.
- *Temps d'espera*: El temps que un treball espera en la cua de preparats.
- *Temps de resposta*: És el temps entre l'enviament d'una petició i la primera resposta produïda. Una mesura millor pels sistemes interactius.

- *Ús de la CPU*: Fracció mitja del temps en que la CPU està ocupada. Inclou el temps perdut pel sistema operatiu.
- *Productivitat*: Nombre de treballs completats per unitat de temps.
- *Temps de retorn*: L'interval de temps que passa entre l'enviament del treball i la seva finalització. És la suma dels períodes perduts per estar en memòria, esperar en la cua de preparats, executar sobre la CPU i, realitzar les E/S.
- *Temps d'espera*: El temps que un treball espera en la cua de preparats.
- *Temps de resposta*: És el temps entre l'enviament d'una petició i la primera resposta produïda. Una mesura millor pels sistemes interactius.

- *Ús de la CPU*: Fracció mitja del temps en que la CPU està ocupada. Inclou el temps perdut pel sistema operatiu.
- *Productivitat*: Nombre de treballs completats per unitat de temps.
- *Temps de retorn*: L'interval de temps que passa entre l'enviament del treball i la seva finalització. És la suma dels períodes perduts per estar en memòria, esperar en la cua de preparats, executar sobre la CPU i, realitzar les E/S.
- *Temps d'espera*: El temps que un treball espera en la cua de preparats.
- *Temps de resposta*: És el temps entre l'enviament d'una petició i la primera resposta produïda. Una mesura millor pels sistemes interactius.

- **Maximitzar Ús de la CPU i Productivitat.**
- Minimitzar Temps de retorn, Temps d'espera i Temps de resposta.
- En molts casos s'optimitza la mitja. Altres casos s'optimitzan els màxims o mínims.
- En el cas de sistemes operatius de temps compartit, es valora minimitzar la variància del temps de resposta en comptes de minimitzar el temps mig de resposta. Un sistema amb un temps de resposta raonable i predictable pot considerarse millor que un de temps mig ràpid però excessivament variable.

- Maximitzar Ús de la CPU i Productivitat.
- Minimitzar Temps de retorn, Temps d'espera i Temps de resposta.
- En molts casos s'optimitza la mitja. Altres casos s'optimitzan els màxims o mínims.
- En el cas de sistemes operatius de temps compartit, es valora minimitzar la variància del temps de resposta en comptes de minimitzar el temps mig de resposta. Un sistema amb un temps de resposta raonable i predictable pot considerarse millor que un de temps mig ràpid però excessivament variable.

- Maximitzar Ús de la CPU i Productivitat.
- Minimitzar Temps de retorn, Temps d'espera i Temps de resposta.
- En molts casos s'optimitza la mitja. Altres casos s'optimitzan els màxims o mínims.
- En el cas de sistemes operatius de temps compartit, es valora minimitzar la variància del temps de resposta en comptes de minimitzar el temps mig de resposta. Un sistema amb un temps de resposta raonable i predictable pot considerarse millor que un de temps mig ràpid però excessivament variable.

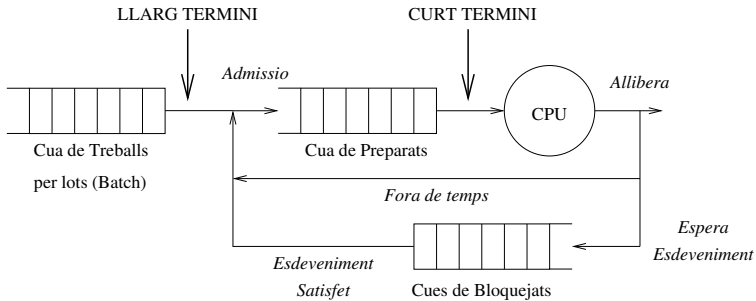
- Maximitzar Ús de la CPU i Productivitat.
- Minimitzar Temps de retorn, Temps d'espera i Temps de resposta.
- En molts casos s'optimitza la mitja. Altres casos s'optimitzan els màxims o mínims.
- En el cas de sistemes operatius de temps compartit, es valora minimitzar la variància del temps de resposta en comptes de minimitzar el temps mig de resposta. Un sistema amb un temps de resposta raonable i predictable pot considerarse millor que un de temps mig ràpid però excessivament variable.

- CURT TERMINI (PLANIFICADOR DE CPU)
 - Selecciona els treballs preparats a ser executats que estan a la memòria principal.
 - Aquesta selecció és molt freqüent ja que poden haver processos que executen només uns quants milisegons abans de demanar una operació d'E/S. Sovint el planificador s'activa un cop cada 10ms i ha de decidir ràpidament el següent procés a executar (1ms per un 91% d'aprofitament del temps de CPU).

- CURT TERMINI (PLANIFICADOR DE CPU)
 - Selecciona els treballs preparats a ser executats que estan a la memòria principal.
 - Aquesta selecció és molt freqüent ja que poden haver processos que executen només un quants milisegons abans de demanar una operació d'E/S. Sovint el planificador s'activa un cop cada 10ms i ha de decidir ràpidament el següent procés a executar (1ms per un 91% d'aprofitament del temps de CPU).

- CURT TERMINI (PLANIFICADOR DE CPU)
 - Selecciona els treballs preparats a ser executats que estan a la memòria principal.
 - Aquesta selecció és molt freqüent ja que poden haver processos que executen només un quants milisegons abans de demanar una operació d'E/S. Sovint el planificador s'activa un cop cada 10ms i ha de decidir ràpidament el següent procés a executar (1ms per un 91% d'aprofitament del temps de CPU).

Tipus de planificadors (II): Curt termini



- LLARG TERMINI.

- Determina quins treballs intensius en recursos s'admeten per ser processats. Sovint aquests treballs tenen una prioritat baixa i provenen d'una cua de treballs per lots (batch).
- Els treballs per lots mantenen ocupats els recursos del sistema en els períodes d'activitat baixa per part dels treballs interactius. En períodes d'alta activitat de treballs interactius haurà de aminorar el nombre de treballs intensius per tal de no enlentir el temps de resposta.
- Un coneixement anticipat sobre el funcionament del treball a executar facilita la tasca del planificador a llarg termini que ha de combinar treballs intensius en CPU amb treballs intensius en E/S. La decisió dels treballs a executar triga més (minuts).
- El planificador a llarg termini controla el *grau de multiprogramació* (el nombre de processos en memòria): El nombre mig de treballs que entren al sistema hauria de ser el nombre mig de treballs que surten. El planificador és cridat cada cop que un treball deixa el sistema.

- LLARG TERMINI.

- Determina quins treballs intensius en recursos s'admeten per ser processats. Sovint aquests treballs tenen una prioritat baixa i provenen d'una cua de treballs per lots (batch).
- Els treballs per lots mantenen ocupats els recursos del sistema en els períodes d'activitat baixa per part dels treballs interactius. En períodes d'alta activitat de treballs interactius haurà de minimitzar el nombre de treballs intensius per tal de no enlentir el temps de resposta.
- Un coneixement anticipat sobre el funcionament del treball a executar facilita la tasca del planificador a llarg termini que ha de combinar treballs intensius en CPU amb treballs intensius en E/S. La decisió dels treballs a executar triga més (minuts).
- El planificador a llarg termini controla el *grau de multiprogramació* (el nombre de processos en memòria): El nombre mig de treballs que entren al sistema hauria de ser el nombre mig de treballs que surten. El planificador és cridat cada cop que un treball deixa el sistema.

- LLARG TERMINI.

- Determina quins treballs intensius en recursos s'admeten per ser processats. Sovint aquests treballs tenen una prioritat baixa i provenen d'una cua de treballs per lots (batch).
- Els treballs per lots mantenen ocupats els recursos del sistema en els períodes d'activitat baixa per part dels treballs interactius. En períodes d'alta activitat de treballs interactius haurà de aminorar el nombre de treballs intensius per tal de no enlentir el temps de resposta.
- Un coneixement anticipat sobre el funcionament del treball a executar facilita la tasca del planificador a llarg termini que ha de combinar treballs intensius en CPU amb treballs intensius en E/S. La decisió dels treballs a executar triga més (minuts).
- El planificador a llarg termini controla el *grau de multiprogramació* (el nombre de processos en memòria): El nombre mig de treballs que entren al sistema hauria de ser el nombre mig de treballs que surten. El planificador és cridat cada cop que un treball deixa el sistema.

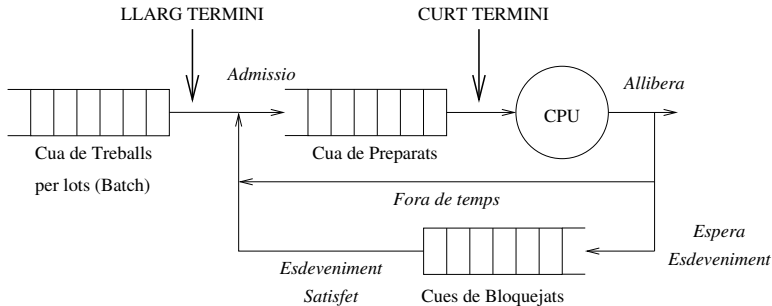
- LLARG TERMINI.

- Determina quins treballs intensius en recursos s'admeten per ser processats. Sovint aquests treballs tenen una prioritat baixa i provenen d'una cua de treballs per lots (batch).
- Els treballs per lots mantenen ocupats els recursos del sistema en els períodes d'activitat baixa per part dels treballs interactius. En períodes d'alta activitat de treballs interactius haurà de aminorar el nombre de treballs intensius per tal de no enlentir el temps de resposta.
- Un coneixement anticipat sobre el funcionament del treball a executar facilita la tasca del planificador a llarg termini que ha de combinar treballs intensius en CPU amb treballs intensius en E/S. La decisió dels treballs a executar triga més (minuts).
- El planificador a llarg termini controla el *grau de multiprogramació* (el nombre de processos en memòria): El nombre mig de treballs que entren al sistema hauria de ser el nombre mig de treballs que surten. El planificador és cridat cada cop que un treball deixa el sistema.

- LLARG TERMINI.

- Determina quins treballs intensius en recursos s'admeten per ser processats. Sovint aquests treballs tenen una prioritat baixa i provenen d'una cua de treballs per lots (batch).
- Els treballs per lots mantenen ocupats els recursos del sistema en els períodes d'activitat baixa per part dels treballs interactius. En períodes d'alta activitat de treballs interactius haurà de minimitzar el nombre de treballs intensius per tal de no enlentir el temps de resposta.
- Un coneixement anticipat sobre el funcionament del treball a executar facilita la tasca del planificador a llarg termini que ha de combinar treballs intensius en CPU amb treballs intensius en E/S. La decisió dels treballs a executar triga més (minuts).
- El planificador a llarg termini controla el *grau de multiprogramació* (el nombre de processos en memòria): El nombre mig de treballs que entren al sistema hauria de ser el nombre mig de treballs que surten. El planificador és cridat cada cop que un treball deixa el sistema.

Tipus de planificadors (IV)



- **MIG TERMINI**

- És el responsable de planificar els processos preparats que estan a la memòria secundària a conseqüència de l'intercanvi.
- Intenta reservar la memòria necessària per executar els processos de la seva cua i fa l'intercanvi per posar el processos a la cua de preparats.

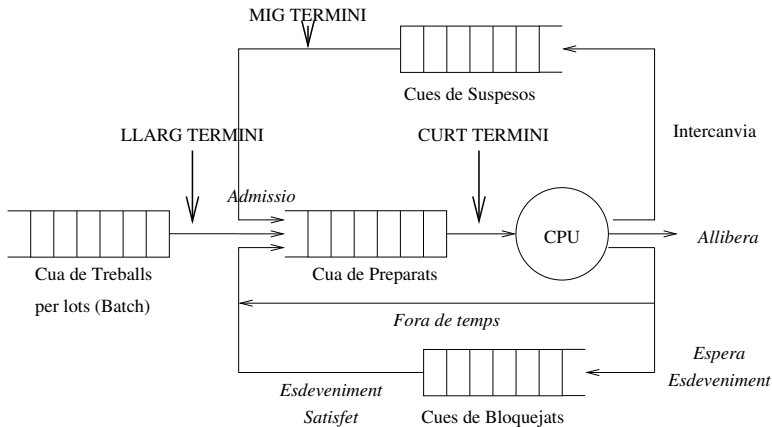
- **MIG TERMINI**

- És el responsable de planificar els processos preparats que estan a la memòria secundària a conseqüència de l'intercanvi.
- Intenta reservar la memòria necessària per executar els processos de la seva cua i fa l'intercanvi per posar el processos a la cua de preparats.

- MIG TERMINI

- És el responsable de planificar els processos preparats que estan a la memòria secundària a conseqüència de l'intercanvi.
- Intenta reservar la memòria necessària per executar els processos de la seva cua i fa l'intercanvi per posar el processos a la cua de preparats.

Tipus de planificadors (VI)



- Quins criteris aplica per decidir quin procés cal executar primer?
- Com i quan es farà efectiva la decisió de planificació?:
 - *No apropiatiu* El procés que té la CPU no la deixa fins que necessiti una operació d' E/S o faci una trucada al S.O. El procés seleccionat podrà llavors apropiar-se de la CPU.
 - *Apropiatiu* El procés que té la CPU pot ser expulsat de la CPU malgrat no haver acabat la seva tongada de càlcul. El procés seleccionat s'apropriarà de la CPU.

- Quins criteris aplica per decidir quin procés cal executar primer?
- Com i quan es farà efectiva la decisió de planificació?:
 - *No apropiatiu* El procés que té la CPU no la deixa fins que necessiti una operació d' E/S o faci una trucada al S.O. El procés seleccionat podrà llavors apropiar-se de la CPU.
 - *Apropiatiu* El procés que té la CPU pot ser expulsat de la CPU malgrat no haver acabat la seva tongada de càlcul. El procés seleccionat s'apropriarà de la CPU.

- Quins criteris aplica per decidir quin procés cal executar primer?
- Com i quan es farà efectiva la decisió de planificació?:
 - *No apropiatiu* El procés que té la CPU no la deixa fins que necessiti una operació d' E/S o faci una trucada al S.O. El procés seleccionat podrà llavors apropiar-se de la CPU.
 - *Apropiatiu* El procés que té la CPU pot ser expulsat de la CPU malgrat no haver acabat la seva tongada de càlcul. El procés seleccionat s'apropriarà de la CPU.

- Quins criteris aplica per decidir quin procés cal executar primer?
- Com i quan es farà efectiva la decisió de planificació?:
 - *No apropiatiu* El procés que té la CPU no la deixa fins que necessiti una operació d' E/S o faci una trucada al S.O. El procés seleccionat podrà llavors apropiar-se de la CPU.
 - *Apropiatiu* El procés que té la CPU pot ser expulsat de la CPU malgrat no haver acabat la seva tongada de càlcul. El procés seleccionat s'apropriarà de la CPU.

- Primer en arribar, Primer en ser servit.
(First Come First Served, FCFS)
 - El primer procés que demana la CPU, serà el que agafarà primer la CPU. És el que porta més temps esperant
 - La implementació es fa mitjançant una cua (Primer en Entrar, Primer en Sortir) (First In First Out, FIFO). Quan la CPU està lliure el procés a la capçalera de la cua pren la CPU i deixa la cua (és eliminat de la cua).
 - El temps de retorn del sistema queda afectat segons l'ordre dels processos.

- Primer en arribar, Primer en ser servit.
(First Come First Served, FCFS)
 - El primer procés que demana la CPU, serà el que agafarà primer la CPU. És el que porta més temps esperant
 - La implementació es fa mitjançant una cua (Primer en Entrar, Primer en Sortir) (First In First Out, FIFO). Quan la CPU està lliure el procés a la capçalera de la cua pren la CPU i deixa la cua (és eliminat de la cua).
 - El temps de retorn del sistema queda afectat segons l'ordre dels processos.

- Primer en arribar, Primer en ser servit.
(First Come First Served, FCFS)
 - El primer procés que demana la CPU, serà el que agafarà primer la CPU. És el que porta més temps esperant
 - La implementació es fa mitjançant una cua (Primer en Entrar, Primer en Sortir) (First In First Out, FIFO). Quan la CPU està lliure el procés a la capçalera de la cua pren la CPU i deixa la cua (és eliminat de la cua).
 - El temps de retorn del sistema queda afectat segons l'ordre dels processos.

- Primer en arribar, Primer en ser servit.
(First Come First Served, FCFS)
 - El primer procés que demana la CPU, serà el que agafarà primer la CPU. És el que porta més temps esperant
 - La implementació es fa mitjançant una cua (Primer en Entrar, Primer en Sortir) (First In First Out, FIFO). Quan la CPU està lliure el procés a la capçalera de la cua pren la CPU i deixa la cua (és eliminat de la cua).
 - El temps de retorn del sistema queda afectat segons l'ordre dels processos.

Algorismes de planificació (II). Exemple

Exemple:

Treball	Temps de tongada
1	24
2	3
3	3

	1	2	3
0	24	27	30
	1	2	3

$$TempsRetornMig = \frac{24+27+30}{3} = 27$$

Si l'ordre de treballs és 2,3,1

	3	2	1
0	3	6	30

$$TempsRetornMig = \frac{3+6+30}{3} = 13$$

- Primer en arribar, Primer en ser servit.
 - *Efecte comboi* Tendeix a afavorir processos intensius de CPU enfront dels que són intensius en E/S. Suposem un procés molt intensiu de CPU i molts altres intensius de E/S. Quan el procés intensiu en CPU s'està executant, tots els altres hauran d'esperar. Quan aquest procés deixi la CPU, s'executaran els altres que ràpidament es quedaran bloquejats esperant una petició d'E/S. Quan les peticions estiguin satisfetes, encara hauran d'esperar que la tongada del procés intensiu en CPU acabi. Els dispositius d'E/S no són aprofitats al màxim. Si el procés intensiu en CPU també es bloqueja, la CPU pot estar desocupada.

- Primer en arribar, Primer en ser servit.
 - *Efecte comboi* Tendeix a afavorir processos intensius de CPU enfront dels que són intensius en E/S. Suposem un procés molt intensiu de CPU i molts altres intensius de E/S. Quan el procés intensiu en CPU s'està executant, tots els altres hauran d'esperar. Quan aquest procés deixi la CPU, s'executaran els altres que ràpidament es quedaran bloquejats esperant una petició d'E/S. Quan les peticions estiguin satisfetes, encara hauran d'esperar que la tongada del procés intensiu en CPU acabi. Els dispositius d'E/S no són aprofitats al màxim. Si el procés intensiu en CPU també es bloqueja, la CPU pot estar desocupada.

- Primer el procés més curt. (Shortest Job First, SJF)

- Es tracta de reduir l'afavoriment del procés més llarg del FCFS.
- Es selecciona el procés amb el temps més curt d'execució.
- Exemple:

Treball	Temps de tongada
1	6
2	3
3	8
4	7

2	1	4	3
---	---	---	---

$$\text{TempsRetornMig} = \frac{3+9+16+24}{4} = 13$$

- És optim ja que dona el temps mig mínim d'espera per un conjunt de processos. Seleccionar un treball curt abans que un treball llarg disminueix molt més el temps d'espera del treball curt que l'increment del temps d'espera del treball llarg.

- Primer el procés més curt. (Shortest Job First, SJF)
 - Es tracta de reduir l'afavoriment del procés més llarg del FCFS.
 - Es selecciona el procés amb el temps més curt d'execució.
 - Exemple:

Treball	Temps de tongada
1	6
2	3
3	8
4	7

2	1	4	3
---	---	---	---

$$\text{TempsRetornMig} = \frac{3+9+16+24}{4} = 13$$

- És optim ja que dona el temps mig mínim d'espera per un conjunt de processos. Seleccionar un treball curt abans que un treball llarg disminueix molt més el temps d'espera del treball curt que l'increment del temps d'espera del treball llarg.

- Primer el procés més curt. (Shortest Job First, SJF)
 - Es tracta de reduir l'afavoriment del procés més llarg del FCFS.
 - Es selecciona el procés amb el temps més curt d'execució.
 - Exemple:

Treball	Temps de tongada
1	6
2	3
3	8
4	7

2	1	4	3
---	---	---	---

$$\text{TempsRetornMig} = \frac{3+9+16+24}{4} = 13$$

- És optim ja que dona el temps mig mínim d'espera per un conjunt de processos. Seleccionar un treball curt abans que un treball llarg disminueix molt més el temps d'espera del treball curt que l'increment del temps d'espera del treball llarg.

- Primer el procés més curt. (Shortest Job First, SJF)
 - Es tracta de reduir l'afavoriment del procés més llarg del FCFS.
 - Es selecciona el procés amb el temps més curt d'execució.
 - Exemple:

Treball	Temps de tongada
1	6
2	3
3	8
4	7

2	1	4	3
---	---	---	---

$$\text{TempsRetornMig} = \frac{3+9+16+24}{4} = 13$$

- És òptim ja que dona el temps mig mínim d'espera per un conjunt de processos. Seleccionar un treball curt abans que un treball llarg disminueix molt més el temps d'espera del treball curt que l'increment del temps d'espera del treball llarg.

- Primer el procés més curt. (Shortest Job First, SJF)
 - Es tracta de reduir l'afavoriment del procés més llarg del FCFS.
 - Es selecciona el procés amb el temps més curt d'execució.
 - Exemple:

Treball	Temps de tongada
1	6
2	3
3	8
4	7

2	1	4	3
---	---	---	---

$$\text{TempsRetornMig} = \frac{3+9+16+24}{4} = 13$$

- És optim ja que dona el temps mig mínim d'espera per un conjunt de processos. Seleccionar un treball curt abans que un treball llarg disminueix molt més el temps d'espera del treball curt que l'increment del temps d'espera del treball llarg.

- Primer el procés més curt.

- La dificultat d'aquest algorisme està en saber el temps que durarà la següent petició de la CPU.

Un mètode consisteix en preveure el temps (τ_i) de la nova tongada a partir de la mitja dels temps de les anteriors tongades (t_i):

$$\tau_{n+1} = \frac{1}{n} \sum_{i=0}^n t_i$$

Que es pot reescriure com

$$\tau_{n+1} = \frac{1}{n} t_n + \frac{n-1}{n} \tau_n$$

Tots els casos tenen el mateix pes.

- Sovint per predir valors futurs a partir de valors passats s'utilitza la tècnica de promig exponencial:

$$\tau_{n+1} = \alpha t_n + (1 - \alpha) \tau_n, \quad 0 \leq \alpha \leq 1$$

Per aquests valors d' α els termes més distants es fan petits

$$\tau_{n+1} = \alpha t_n + (1 - \alpha) \alpha t_{n-1} + \dots + (1 - \alpha)^i \alpha t_{n-i} + \dots + (1 - \alpha)^n \tau_0$$

És a dir, els valors del passat influeixen menys en la previsió del valor futur.

- Primer el procés més curt.
- La dificultat d'aquest algorisme està en saber el temps que durarà la següent petició de la CPU.

Un mètode consisteix en preveure el temps (τ_i) de la nova tongada a partir de la mitja dels temps de les anteriors tongades (t_i):

$$\tau_{n+1} = \frac{1}{n} \sum_{i=0}^n t_i$$

Que es pot reescriure com

$$\tau_{n+1} = \frac{1}{n} t_n + \frac{n-1}{n} \tau_n$$

Tots els casos tenen el mateix pes.

- Sovint per predir valors futurs a partir de valors passats s'utilitza la tècnica de promig exponencial:

$$\tau_{n+1} = \alpha t_n + (1 - \alpha) \tau_n, \quad 0 \leq \alpha \leq 1$$

Per aquests valors d' α els termes més distants es fan petits

$$\tau_{n+1} = \alpha t_n + (1 - \alpha) \alpha t_{n-1} + \dots + (1 - \alpha)^i \alpha t_{n-i} + \dots + (1 - \alpha)^n \tau_0$$

És a dir, els valors del passat influeixen menys en la previsió del valor futur.

- Primer el procés més curt.
 - La dificultat d'aquest algorisme està en saber el temps que durarà la següent petició de la CPU.

Un mètode consisteix en preveure el temps (τ_i) de la nova tongada a partir de la mitja dels temps de les anteriors tongades (t_i):

$$\tau_{n+1} = \frac{1}{n} \sum_{i=0}^n t_i$$

Que es pot reescriure com

$$\tau_{n+1} = \frac{1}{n} t_n + \frac{n-1}{n} \tau_n$$

Tots els casos tenen el mateix pes.

- Sovint per predir valors futurs a partir de valors passats s'utilitza la tècnica de promig exponencial:

$$\tau_{n+1} = \alpha t_n + (1 - \alpha) \tau_n, \quad 0 \leq \alpha \leq 1$$

Per aquests valors d' α els termes més distants es fan petits

$$\tau_{n+1} = \alpha t_n + (1 - \alpha) \alpha t_{n-1} + \dots + (1 - \alpha)^i \alpha t_{n-i} + \dots + (1 - \alpha)^n \tau_0$$

És a dir, els valors del passat influeixen menys en la previsió del valor futur.

- **Prioritat.**

- El SJF és un algorisme que usa prioritat. La prioritat és la inversa del temps previst de tongada.
- Es poden utilitzar diferents criteris interns per fixar les prioritats dels processos (límits de temps, quantitat de memòria, nombre de fitxers oberts, la raó de tongades de E/S enfront tongades de càlcul, etc)
- El problema de l'ús de la prioritat és que pot deixar alguns processos en estat de *bloqueig indefinit o inanició*. Una solució consisteix en *envellir o madurar* el bloqueig indefinit dels processos. S'incrementa la prioritat dels processos que estan en el sistema des de fa temps.

- Prioritat.

- El SJF és un algorisme que usa prioritat. La prioritat és la inversa del temps previst de tongada.
- Es poden utilitzar diferents criteris interns per fixar les prioritats dels processos (límits de temps, quantitat de memòria, nombre de fitxers oberts, la raó de tongades de E/S enfront tongades de càlcul, etc)
- El problema de l'ús de la prioritat és que pot deixar alguns processos en estat de *bloqueig indefinit o inanició*. Una solució consisteix en *envellir o madurar* el bloqueig indefinit dels processos. S'incrementa la prioritat dels processos que estan en el sistema des de fa temps.

- Prioritat.
 - El SJF és un algorisme que usa prioritat. La prioritat és la inversa del temps previst de tongada.
 - Es poden utilitzar diferents criteris interns per fixar les prioritats dels processos (límits de temps, quantitat de memòria, nombre de fitxers oberts, la raó de tongades de E/S enfront tongades de càlcul, etc)
 - El problema de l'ús de la prioritat és que pot deixar alguns processos en estat de *bloqueig indefinit o inanició*. Una solució consisteix en *envellir o madurar* el bloqueig indefinit dels processos. S'incrementa la prioritat dels processos que estan en el sistema des de fa temps.

- Prioritat.

- El SJF és un algorisme que usa prioritat. La prioritat és la inversa del temps previst de tongada.
- Es poden utilitzar diferents criteris interns per fixar les prioritats dels processos (límits de temps, quantitat de memòria, nombre de fitxers oberts, la raó de tongades de E/S enfront tongades de càlcul, etc)
- El problema de l'ús de la prioritat és que pot deixar alguns processos en estat de *bloqueig indefinit o inanició*. Una solució consisteix en *envellir o madurar* el bloqueig indefinit dels processos. S'incrementa la prioritat dels processos que estan en el sistema des de fa temps.

ALGORISMES APROPIATIUS ROUND ROBIN

- Temps compartit.
- La cua de preparats es tracta com una cua circular
- Cada procés executa un quantum (10ms, per ex.)
 - si acaba abans (espera E/S o ha acabat) passem al següent treball.
 - si el temps de tongada es superior al quantum l'encuem al final de la cua i passem al següent treball.

ALGORISMES APROPIATIUS ROUND ROBIN

- Temps compartit.
- La cua de preparats es tracta com una cua circular
- Cada procés executa un quantum (10ms, per ex.)
 - si acaba abans (espera E/S o ha acabat) passem al següent treball.
 - si el temps de tongada es superior al quantum l'encuem al final de la cua i passem al següent treball.

ALGORISMES APROPIATIUS ROUND ROBIN

- Temps compartit.
- La cua de preparats es tracta com una cua circular
- Cada procés executa un quantum (10ms, per ex.)
 - si acaba abans (espera E/S o ha acabat) passem al següent treball.
 - si el temps de tongada es superior al quantum l'encuem al final de la cua i passem al següent treball.

ALGORISMES APROPIATIUS ROUND ROBIN

- Temps compartit.
- La cua de preparats es tracta com una cua circular
- Cada procés executa un quantum (10ms, per ex.)
 - si acaba abans (espera E/S o ha acabat) passem al següent treball.
 - si el temps de tongada es superior al quantum l'encuem al final de la cua i passem al següent treball.

ALGORISMES APROPIATIUS ROUND ROBIN

- Temps compartit.
- La cua de preparats es tracta com una cua circular
- Cada procés executa un quantum (10ms, per ex.)
 - si acaba abans (espera E/S o ha acabat) passem al següent treball.
 - si el temps de tongada es superior al quantum l'encuem al final de la cua i passem al següent treball.

ROUND ROBIN

- Exemple:

Treball	Temps de tongada
1	24
2	3
3	3

1	2	3	1	1	1	1	1
---	---	---	---	---	---	---	---

$$TempsRetornMig = \frac{47}{3} = 6$$

- El rendiment depén del temps de quantum.
Si $t_{quantum} \rightarrow \infty \Rightarrow FCFS$ Si $t_{quantum} \rightarrow 0 \Rightarrow$ Compartició processador.
- Context Switch.

ROUND ROBIN

- Exemple:

Treball	Temps de tongada
1	24
2	3
3	3

1	2	3	1	1	1	1	1
---	---	---	---	---	---	---	---

$$\text{TempsRetornMig} = \frac{47}{3} = 6$$

- El rendiment depén del temps de quantum.
Si $t_{\text{quantum}} \rightarrow \infty \Rightarrow FCFS$ Si $t_{\text{quantum}} \rightarrow 0 \Rightarrow$ Compartició
processador.
- Context Switch.

ROUND ROBIN

- Exemple:

Treball	Temps de tongada
1	24
2	3
3	3

1	2	3	1	1	1	1	1
---	---	---	---	---	---	---	---

$$\text{TempsRetornMig} = \frac{47}{3} = 6$$

- El rendiment depén del temps de quantum.
Si $t_{\text{quantum}} \rightarrow \infty \Rightarrow FCFS$ Si $t_{\text{quantum}} \rightarrow 0 \Rightarrow$ Compartició processador.
- Context Switch.

- Fixed-Priority Scheduling (FPS): Prioritat fixa calculada abans de córrer.
 - Rate Monotonic Scheduling (RMS): Si les tasques són periòdiques, la prioritat s'estableix fent que la prioritat més alta correspongui al període més curt.

Inconvenients: Ús baix del processador. Fam i Bloquejos (starvation and deadlock)

- Deadline-based Dynamic Scheduling (DMS): Prioritats d'acord al seu termini
 - Earliest Deadline First (EDF): Alta prioritat amb el termini més curt (més aviat). No té en compte el temps d'execució.
 - Basats en Laxitud (Laxity): Alta prioritat a menor laxitud (Maximum Laxity First (MLF), Least Slack-timeFirst, (LSF), Least Laxity First (LLF)
Laxitud=termini-temps restant en executar-se (estimació)