Planificació de Processos

1.- Suponed que tenemos los siguientes procesos para ejecutar en el sistema:

proceso	tiempo total cpu	ráfaga cpu	prioridad
1	17	5	3
2	3	1	1
3	5	2	3
4	3	1	4
5	7	4	2

El orden de llegada de los procesos es :1, 2, 3, 4, 5. La diferencia de tiempos entre llegadas la despreciamos. La prioridad se considera más alta cuanto mayor es el número indicado en la columna prioridad. El tiempo que tarda una e/s es de 1 ciclo.

a) Realiza un diagrama de Gantt, utilizando como algoritmos de planificación: FCFS, Round Robin (quantum=2) y Prioridades no Apropiativas. Indica para cada uno de los casos las posibles apropiaciones.

b) ¿Cuál es el tiempo de retorno de cada uno de los procesos en cada caso? ¿Cuál es la planificación con la media de tiempo de retorno más bajo entre todos los procesos?

c) ¿Cuál es el tiempo de espera de cada uno de los procesos en cada caso? ¿Cuál es la planificación con la media de tiempo de espera más bajo entre todos los procesos?

d) ¿Existen diferencias de throughput entre los algoritmos?

2.- Examen Final Q2 02-03. Rellena la siguiente tabla sobre los algoritmos de planificación vistos en clase sin repetir ninguno:

Algoritmo	Apropiativo	Justo	Utiliza quantum?	Utiliza ráfagas?	Utiliza prioridades?
FCFS					
		Sí			
	No				No
		No	Sí		

3.- Teniendo en cuenta la siguiente tabla de procesos, y que conocemos a priori lo que dura cada ráfaga de cpu. Si utilizáramos como políticas de planificación FCFS, y Prioridades Apropiativas: ¿cuál es la que da el mejor, y cuál la que da el peor TROUGHPUT?

Cada e/s dura 2 unidades de tiempo.

Proceso	tiempo llegada	tiempo cpu total	ráfaga de cpu	prioridad
1	0	5	2	1 (-)
2	1	6	4	2
3	1	4	3	3
4	2	3	2	4 (+)

4.- Se propone el siguiente algoritmo apropiativo basado en el cambio dinámico de prioridades, donde mayor número equivale a prioridad mayor.

Cuando un proceso entra en el sistema recibe prioridad 0. Cuando un proceso se ejecuta, a su prioridad se le suma el valor β en cada interrupción de reloj. Si no se ejecuta (ready), a su prioridad se le suma el valor α en cada interrupción de reloj.

Los procesos no realizan e/s.

a) Asumiendo que $\alpha=-2$ y $\beta=-1$ dibuja el diagrama de Gantt para el siguiente conjunto de procesos:

proceso	tiempo llegada	tiempo total cpu
1	0	15
2	3	6
3	6	3
4	15	6

b) Para el algoritmo dado decir cuáles de las siguientes afirmaciones son verdaderas y cuáles falsas. Razonar las respuestas.

b.1) Una vez que a un proceso se le concede la CPU nunca es desbancado por otro que esté esperando.

b.2) La prioridad de un proceso es siempre menor o igual que 0.

b.3) El algoritmo garantiza que ningún proceso sufra inanición.

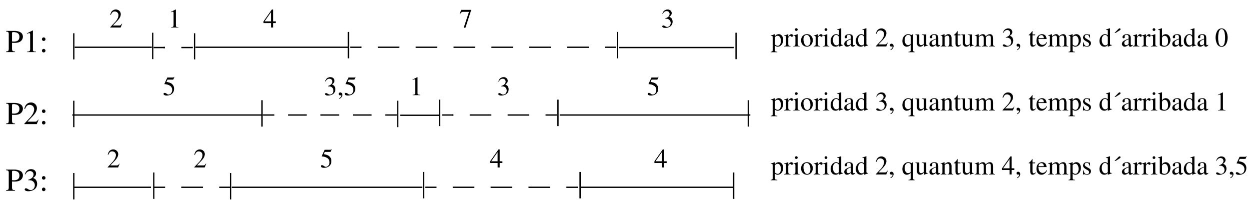
b.4) El proceso recién llegado es el primero que se pone en ejecución.

b.5) El comportamiento de la cola ready es LIFO.

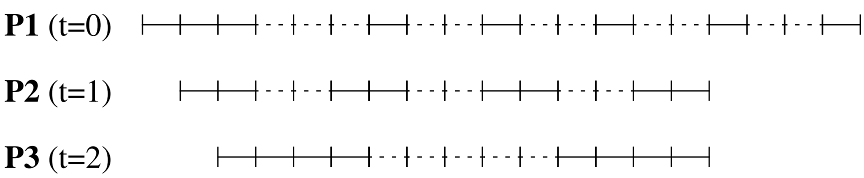
b.6) Intercambiando los valores de α y de β conseguimos un algoritmo Round Robin.

b.7) Si los valores α y β fuesen positivos, el algoritmo sería FIFO.

5.- Examen Final Q2 00-01. Dibuixeu el diagrama de Gantt resultat de l´execució concurrent dels següents processos si la política de planificació és Prioritats no Apropiatives (valor més gran de prioritat indica major prioritat):



6.- Control Aplicación Q2 98-99. Tenim tres processos amb el comportament que mostra la següent figura (el temps d'arribada al sistema és el que es mostra entre parèntesi):

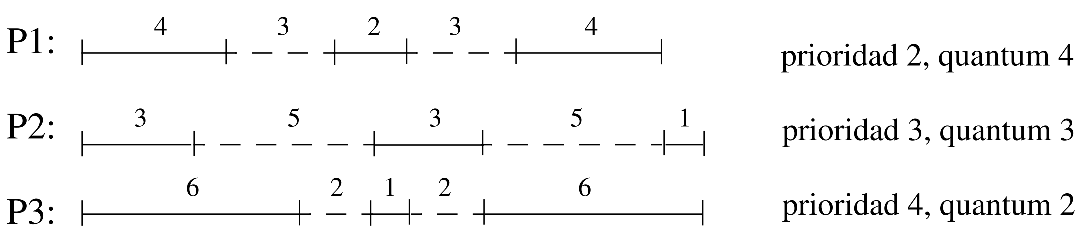


Es demana que ompliu la següent taula assumint les següents polítiques de planificació:
- algorisme per **Prioritats** Apropiatiu, assumint que:
    - la prioritat de P1 és 1, la de P2 és 2, i la de P3 és 3;
- algorisme **Round-Robin** amb 2 unitats de temps de quantum per cada procés.

En qualsevol política resoleu els empats amb FIFO com a segon criteri.

Política	**Throughput**	**Temps mig de Retorn**	**Temps mig d'Espera**	**Eficiencia**
Prioritats				
Round-Robin				

7.- Control Aplicación Q2 00-01. Dados tres procesos con el comportamiento igual al de la figura:



Dibujad el diagrama de Gantt y calcular el tiempo de retorno si utilizamos los siguientes algoritmos (considérese que todos los procesos llegan en t = 0)

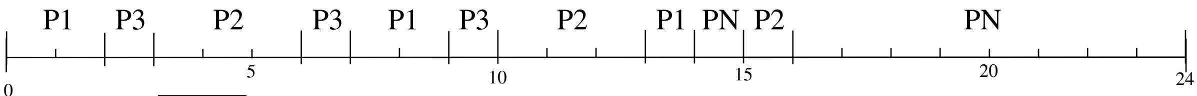
a) Round Robin

b) Prioridades no Apropiativas

c)¿Qué operaciones relacionadas con la planificación de procesos se realizan en la rutina de interrupción de reloj para cada uno de los casos anteriores?

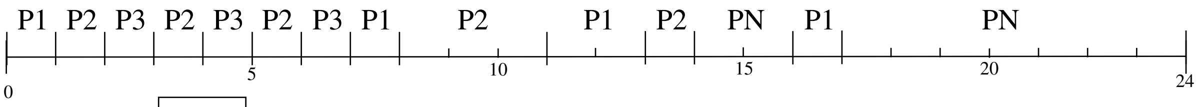
8.- Executem quatre processos (on PN indica el procés Nul) aplicant diferents algorismes de planificació. Els diagrames de Gantt resultants d'aplicar l' algorisme de SJF i el d'aplicar l'algorisme de Prioritats Apropiatiu es mostren tot seguit:

SJF



Throughput = ☐

Prioritats Apropiatiu



Throughput = ☐

Indicar als recuadres quin és el Throughput per cada algorisme (considerant els processos 1, 2 i 3).

Contestar a les següents preguntes:

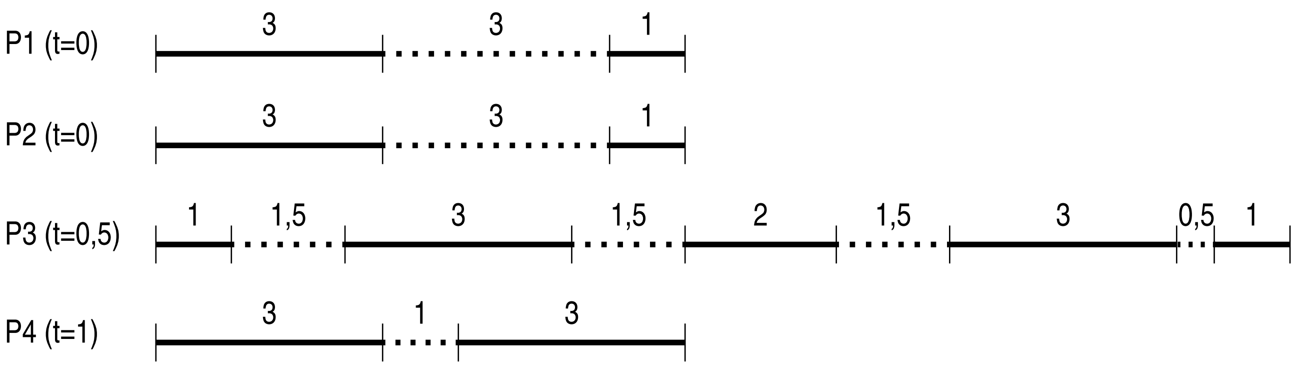
a) Algorisme que dóna un millor Throughput

b) Algorisme que dóna un pitjor Throughput

c) Omplir la següent taula sabent que un procés té prioritat 1, un altre 2 i un altre 3, on un nombre més gran indica més prioritat.

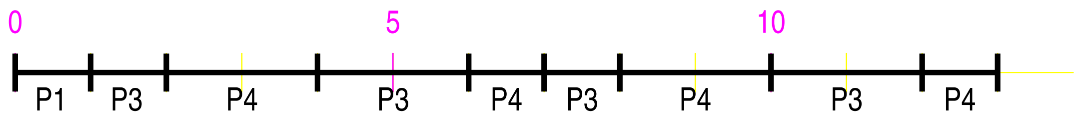
Procés	Arribada al sistema (en unitats de temps)	Temps total de CPU	Durada de les ràfagues de cpu	Temps de cada E/S	Prioritat
1					
2					
3					

9.- Examen final Q1 00-01. Tenim quatre processos P1, P2, P3 i P4 executant-se en un sistema. El comportament de cada un dels processos en termes de càlcul i entrada/sortida és el següent (el nombre entre parèntesis indica el temps d'arribada):



a on les unitats representen tics de rellotge, i els nombres entre parèntesis indiquen els temps d'arribada dels processos al sistema.

El diagrama de Gantt després dels 13 primers cicles de rellotge és el que es mostra a continuació:



Es demana que respongueu a les següents qüestions:

a) Quin algorisme de planificació s'està executant en el sistema?

b) Indiqueu quines apropiacions hi ha hagut en aquest interval i raoneu el motiu de la apropiació.

c) Quants traps s'han executat com a mínim en aquest interval de temps (incloent el tic 13 de rellotge)? Indiqueu de quin tipus és cada trap.

d) Dibuixeu la continuació del diagrama de Gantt a partir del tic de rellotge 13, assumint que en aquest moment s'executa un algorisme de planificació **Round Robin amb prioritat no apropiativa**. Els processos P1 i P2 tenen quantum 2 i prioritat 2, i els processos P3 i P4 tenen quantum 2 i prioritat 3.

10.- Supongamos que hemos implementado la rutina de atención a la interrupción de teclado de Onion de la siguiente forma:

```
rut_atencion_int_tec (){
 int * ptero;
 ptero = salvar ();
         :
         :
 restaurar (ptero);
}
```

a) Si implementáramos el scheduling de procesos de Onion (prioridades y Round-Robin) con apropiación inmediata, describid los problemas que pudieran producirse.

b) ¿Y si fuera con apropiación diferida ?

11.- Examen Final Q2 01-02. ¿Qué estructuras de datos y rutinas habría que modificar/ añadir para ofrecer la siguiente llamada al sistema: *quiespare*? Esta llamada devuelve el PID del padre del proceso que la invoca.

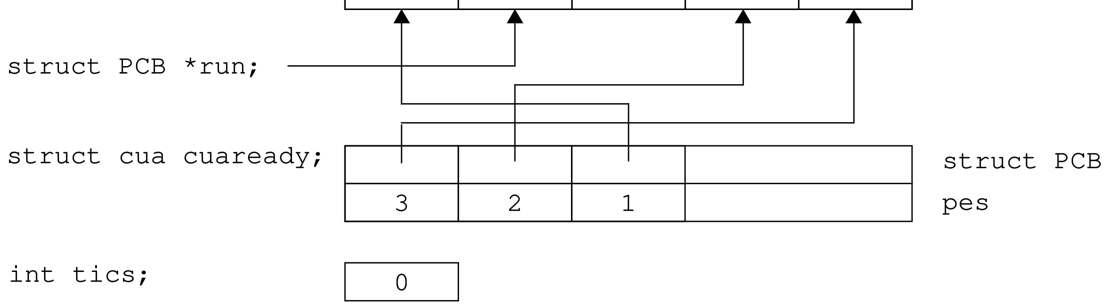
12.- Control Aplicación Q2 99-00. Tenim un sistema operatiu amb un algorisme de planificació **Round-Robin amb prioritat no apropiativa**. Es demana:

a) Escriviu el pseudo-codi relacionat amb la planificació que hi ha a la rutina d´atenció a la interrupció de rellotge. Assumiu que el comptador `tics` **s´incrementa a cada interrupció de rellotge** fins arribar al quantum (per comptes de decrementar-se fins arriba a zero).

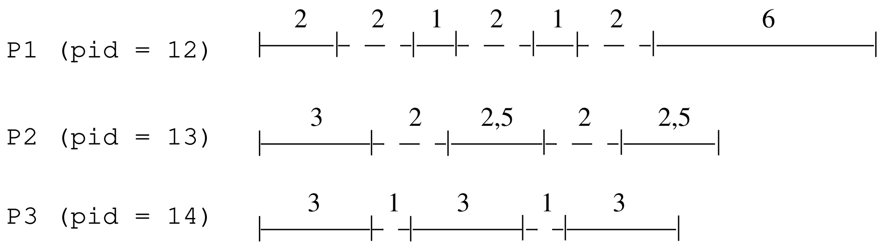
Just a l´acabar d´executar el codi de la rutina d´atenció a la interrupció de rellotge tenim a ONION unes estructures de dades amb els següents valors:

`struct PCB pcbs[5];`

0	1	2	3	4	index
1	12	-1	13	14	pid
1	2	-1	2	3	prioritat
1	3	-1	2	2	quantum
...	...	...	...	...	...



El que li queda a cada procés per executar-se, representat en unitats de temps(tics de rellotge) mitjançant intervals de CPU i E/S és el que es motra a continuació:

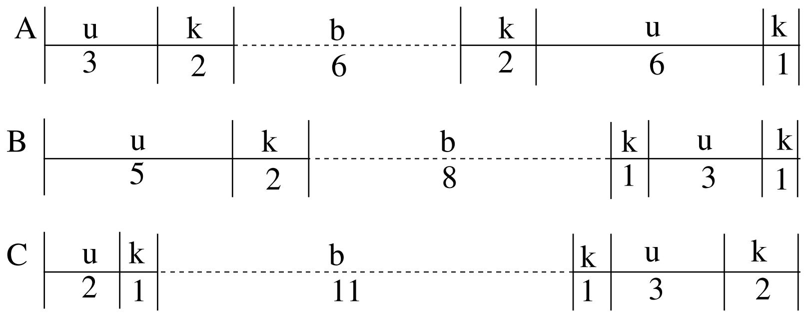


b) Dibuixeu el diagrama de Gantt assumint un algorisme de planificació **Round-Robin amb prioritat no apropiativa**, i considerant el temps actual com a t=0.

c) Quantes **apropiacions** hi ha hagut per **exhauriment de quantum**? Indiqueu en quins instants de temps s´han produit.

d) Quantes **apropiacions** hi ha hagut per **prioritat**? Indiqueu en quins instants de temps s´han produit.

13.- Dibuja el diagrama de Gantt y calcula:
- throughput (productividad) de los siguientes flujos:

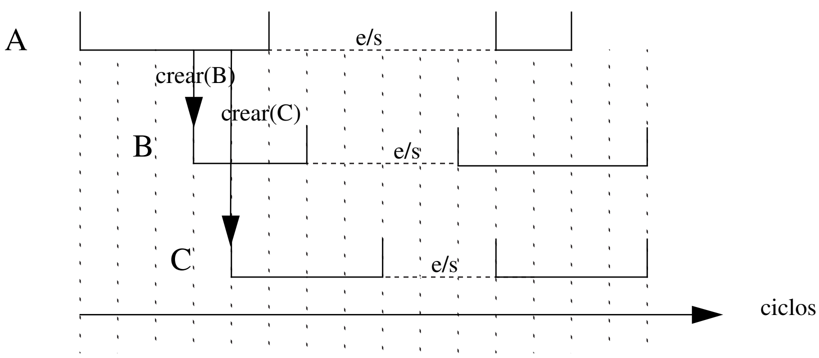


donde u, k y b representan los estados user-running, kernel-running y blocked respectivamente.

El algoritmo de scheduling es por **prioridades apropiativo**. La prioridad de los flujos mientras se ejecutan en modo kernel es 10 y mientras se ejecutan en modo user son 2, 3 y 4 respectivamente para A, B, y C. (Valores más altos indican prioridades más altas).

Los tres procesos llegan en la unidad de tiempo 0.

14.- Dibujar el diagrama de Gantt correspondiente a los 16 primeros ciclos de la ejecución en ONION (con preemption inmediata) de los procesos A,B,C, sabiendo que las prioridades son 2, 3 y 3 respectivamente, el quantum de todos ellos es 2 y su comportamiento sería el siguiente si se ejecutasen en paralelo



15.- Es mostren a sota algunes de les rutines que formen part de la nostra implementació d'ONION. Els valors numèrics que hi ha a la vora d'alguns troços del pseudocodi de les subrutines indica el temps, en unitats de temps (u.t.), que triguen en fer-se aquelles parts del codi. Considerarem que el temps necessari per a executar la resta de codi que es pugui necessitar, es mostri a la figura o no, és despreciable, nul, zero, 0!!, amb l'exepció de:

Rutina de Servei a la Interrupció del teclat: 0.1 u.t.
llegir_teclat_nuc: depén de si bloqueja o no
  si s'executa **sense bloquejar** al procés: **0.4 u.t.**
  si s'executa **bloquejant** al procés que l'ha cridat i fent un canvi de contexte: **0.2 u.t.**

Sabent que l'algorisme de **planificació** és el d'**Onion** amb la seva variant d'**apropiació immediata**, dibuixar el diagrama de Gantt a l'interval comprés entre la u.t. 0 i la u.t. 9, on es considera com u.t. 0 l'instant en que es fica el procés menu en execució. Arriben **interrupcions del teclat a les u.t. 0.7, 3.8, i 4.1**. Els caràcters llegits són **"1", "1", "2"**. Hi ha buffer de teclat.

```
main (....)
{
  ...
  omplir_pcb_proc_nul();
  desinhibir(0x200);
  crear_pcb(menu,4,10);
  for (;;);
}
```

```
menu ()
{
     ...
0.3  inicialitzacions();
  while(1) {
    c = llegir_teclat_nuc();
    switch (c) {
      case 1: crear_pcb_nuc(pantalla,2,2); break;
      case 2: crear_pcb_nuc(teclat,3,5); break;
      default: escr_msg_error("Opció Invàlida");
    }
  }
}
```

```
pantalla ()
{
  ...
  while(1) {
      for (i=21; i<80; i++) {
0.3     escriure_pantalla_nuc(...);
      }
  }
}
```

```
crear_pcb_nuc (...)
{
       ...
       inhibir();
0.1    obtenir_i_marcar_pcb();
       desinhibir(0x200);
0.3 -> omplir_pcb_i_pila();
       inhibir();
0.1    planificació();
       desinhibir(0x200);
       ...
}
```

```
teclat ()
{
  ...
  while(1) {
      c = llegir_teclat_nuc();
      for (i=21; i<80; i++) {
0.2     escriure_pantalla_nuc(...);
      }
  }
}
```

```
nul ()
{
  ...
  for(;;);
}
```

NOTA: **Indicar clarament al peu del diagrama de Gantt les u.t.** en que hi ha *canvis de procés en execució*, el procés entra a executar la rutina *crear_pcb_nuc*, *llegir_teclat_nuc*, o s'executa la *RSI del teclat*. Si hi han diversos processos que executen una mateixa rutina reflexeu-ho al diagrama ficant *nom_rutina*$_i$ on la *i* indica si és el 1r. 2n., ... procés d'aquell tipus.

...	exemple$_1$	exemple$_1$ crear_pcb_nuc	exemple$_1$ llegir_tec_nuc	exemple$_2$	RSI Tec	. . .
0	0.4	1.2	1.5	1.9	2.3	2.4

16.- Examen final Q2 98-99. Es mostren a sota algunes de les rutines que formen part de la nostra implementació d'Onion, a on els valors numèrics que hi ha al costat del pseudocodi indiquen les unitats de temps (en tics de rellotge) que triguen en fer-se aquelles parts de codi. Considereu que el temps necessari per a executar la resta del codi és zero.

```
void main() {
      ...
      omplir_pcb_proc_null();
      desinhibir(0x200);
      crear_pcb(menu,4,10);
      for(;;);
}

void menu() {
0.2 [ inicialitz();
    [ i=0;j=0;
      for(;;){
         c=llegir_teclat_nuc();
         switch(c){
            case 1: crear_pcb_nuc(p1,3,2);
            case 2: crear_pcb_nuc(p2,3,2);
            case 3: crear_pcb_nuc(p3,2,3);
         }
      }
}
void crear_pcb_nuc(...) {
    [ inhibir();
0.2 [ obtenir_i_marcar_pcb(...);
    [ desinhibir();
      omplir_pcb_i_pila(...); ] 0.3
    [ inhibir();
0.1 [ planificacio();
    [ desinhibir();
}
```

```
void RSI_teclat() {
   0.1 [ ...
}

void destruir_pcb_nuc(...) {
   0.5 [ buscar_pcb(...);
       [ alliberar_pcb(...);
}

void p1() {
   3.0 [ rut3();
         retardar_nuc(3);
   3.0 [ rut3();
         retardar_nuc(3);
   3.0 [ rut3();
         destruir_pcb_nuc(quisoc());
}

void p2() {
   3.0 [ rut3();
         retardar_nuc(1);
   3.0 [ rut3();
         destruir_pcb_nuc(quisoc());
}

void p3() {
   2.0 [ rut2();
         retardar_nuc(2);
   3.0 [ rut3();
         destruir_pcb_nuc(quisoc());
}
```

Dibuixeu el diagrama de Gantt en els intervals de temps especificats, assumint els algorismes de planificació **Round-Robin amb prioritat apropiativa i preemció immediata** (apartats a i b) i **Round-Robin amb prioritat no apropiativa** (apartat c). L'instant de temps zero es considera una vegada s´ha creat el procés menu i ha començat a executar-se.

Considereu que podem tenir més d'un procés bloquejat en el temporitzador. Les interrupcions de teclat arriben a les unitats de temps 0.9, 1.4 i 1.8, i els caracters llegits són el 2, 1 i 3. Hi ha buffer de teclat.

a) Apropiació inmediata, des de la unitat de temps 0 a la 4:

b) Apropiació inmediata, des de la unitat de temps 4 a la 28:

c) No apropiatiu, des de la unitat de temps 4 a la 28:

17.- Implementem un sistema de planificació de processos a Onion basat en cues multinivell. El planificador és apropiatiu immediat i disposa de dues cues: una amb prioritat alta i Round Robin com a política de planificació, i una altra amb prioritat baixa i FCFS. La primera s'usarà per a processos interactius i la segona per a processos batch.

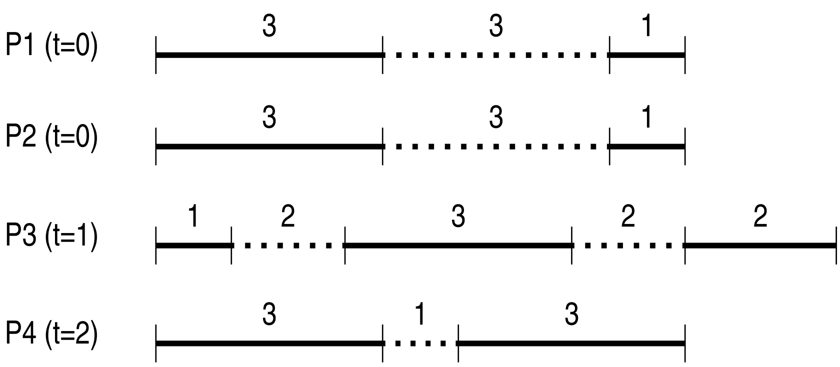
	Prioritat	Temps arribada	Quantum	Cicles totals de CPU
P1	Alta	0	4	13
P2	Alta	1	3	9
P3	Baixa	2	3	12
P4	Baixa	3	6	23
P5	Alta	35	2	8

Cada procés alterna 5 cicles de CPU amb 1 cicle d'E/S.

a) Dibuixeu el diagrama de Gantt de l'execució dels processos durant els primers 45 cicles de temps.
b) Calcula el throughput del sistema en aquests 45 cicles

18.- Tenim un sistema que utilitza com algorisme de planificació cues multinivell realimentadas no apropiatives. Hi han tres cues. De menys a més prioritaries, a la primera s'aplica Round Robin amb un quantum de 2 cicles, a la segona s'aplica Round Robin amb un quantum de 4 cicles i a la última s'aplica prioritats apropiatives immediates. Tots els processos comencen a la cua menys prioritària. Cada cop que un procés s'executa durant un quantum sencer s'encua a la cua de prioritat inmediatament superior.

Si tenim els següents processos:



a) Dibuixa el diagrama de Gantt de l'execucio de aquests processos.
b) Indica al diagrama amb un cercle els instants en que el procés canvia de cua.
c) Calcula el temps mig d'espera d'aquest sistema.

19.- Tenim tres processos amb el comportament descrit a la seguent taula de processos:

Procés	Arribada al sistema (en unitats de temps)	Temps total de CPU	Durada de les ràfagues de cpu	Temps de cada E/S	Prioritat inicial
1	0	15	6	1	3
2	0	7	3	2	1
3	3	7	4	3	2

a) Dibuxeu el diagrama de Gantt si s'utilitza un algorisme de Prioritats Apropiatives, tenint en compte que tambe s'aplica envelliment de prioritats (Aging). Un proces veura incrementada la seva prioritat en una unitat per cada 4 cicles consecutius en que es trobi a la cua de Ready. Quan el proces s'executi, en el moment de deixar la CPU recuperara la seva prioritat inicial.

b) Marqueu totes les apropiacions que s'hagin produit

c) Suposeu que ara l'envelliment de prioritats s'aplica als procesos que es troben bloquejats per E/S en comptes dels procesos que es troben a la cua de Ready. Per cada cicle d'E/S que fa un procés la seva prioritat s'incrementa temporalment en 1.

d) Quina diferencia de comportament hi ha entre les dues propostes?