

Sistemas Operativos. Grado Ingeniería Informática. TGR Procesos.

1.-Considerar los siguientes procesos con los tiempos de llegadas, prioridades y ráfagas de cpu.

Proceso	Ráfaga de CPU	Prioridad	Tiempo llegada
A	50 ms	4	0 ms
B	20 ms	1	20 ms
C	100 ms	3	40 ms
D	40 ms	2	60 ms.

Mostrar cómo se planifican estos procesos con los algoritmos FCFS, SJF, SRTF, Prioridades Apropiativo y no Apropiativo (un valor más pequeño en la tabla indica más prioridad) y Round Robin con quantum de 30 ms. ¿Cuál es el tiempo de espera promedio para los distintos algoritmos y esta carga de trabajo?

SOLUCION:

llegada	A		B		C		D													
FCFS	A	A	A	A	A	B	B	C	C	C	C	C	C	C	C	C	D	D	D	D
SJF	A	A	A	A	A	B	B	D	D	D	D	C	C	C	C	C	C	C	C	C
SRTF	A	A	B	B	A	A	A	D	D	D	D	C	C	C	C	C	C	C	C	C
PRI.n	A	A	A	A	A	B	B	D	D	D	D	C	C	C	C	C	C	C	C	C
PRI.a	A	A	B	B	C	C	D	D	D	D	C	C	C	C	C	C	C	A	A	A
RR(30)	A	A	A	B	B	A	A	C	C	C	D	D	D	C	C	C	D	C	C	C

$$T_{FCFS} = \frac{0+30+30+110}{4}, T_{SJF} = \frac{0+30+10+70}{4}, T_{SRTF} = \frac{20+0+70+10}{4}$$

$$T_{pri.n} = \frac{0+30+10+70}{4}, T_{pri.a} = \frac{160+0+40+0}{4}, T_{RR} = \frac{20+10+70+70}{4}$$

2.- En el código que se muestra a continuación y que corresponde al ejecutable “./a.out” . ¿Cuántos procesos se crean? (no se cuenta la creación del primer proceso desde el shell?) ¿Qué salida produce por pantalla?

- a) ninguno
- b) uno
- c) 500
- d) 499
- e) 100
- f) 99
- g) infinitos

```
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>

#define MAX 500

main(int argc, char * argv[])
{
    int i=0;

    printf ("i vale %d \n",i);

    i++;

    if (i==MAX)
        exit(0);
    execl("./a.out", "./a.out", NULL);
    i=i+5;
}
```

SOLUCION:

a) Ninguno.

No se crea ningún proceso, el proceso es un bucle infinito que reemplaza su código por el mismo código una y otra vez

3.- En el código que se muestra a continuación y que corresponde al ejecutable “./a.out”. ¿Cuántos procesos se crean? (no se cuenta la creación del primer proceso desde el shell). Qué salida produce por pantalla?

a) ninguno

b) uno

c) 500

d) 499

e) 100

f) 99

g) infinitos

```
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>

#define MAX 500
```

```

main(int argc, char * argv[])
{
    int i=0;
    pid_t pid;

    i++;
    printf ("i vale %d \n",i);

    if ((pid=fork()) !=0)
        exit(0);
    sleep (1);
    if (i==MAX)
        exit(0);
    execl("./a.out", "./a.out", NULL);
    i=i+5;
}

```

SOLUCION:

g)Infinitos En este caso se crea un proceso hijo que es el que reemplaza su código por uno con el mismo código. Dicho proceso hijo hace lo mismo.

4.-Considérese un sistema con planificación por prioridades dinámicas y apropiativas. El funcionamiento del sistema es el siguiente:

1. El sistema tiene definidas prioridades de 1 a 5 (menor número indica menor prioridad)
2. Se ejecuta siempre el proceso de más prioridad. Cuando dos procesos tienen la misma prioridad
 1. si uno de ellos está en CPU se deja al que está en CPU.
 2. Si ninguno de ellos está en CPU obtiene la CPU el que lleve mas tiempo en la cola de listo.
3. Si un proceso es expulsado de la CPU, cuando la vuelve a obtener su cuánto vuelve a comenzar.
4. A cada proceso se le asigna un cuanto que depende de su prioridad y es igual $6-P$, siendo P la prioridad (p.e. a un proceso de prioridad 3 se le asigna un cuanto de 3).
5. Suponemos que varios procesos pueden realizar e/s concurrentemente.
6. El sistema recalcula la prioridad del proceso que abandona la CPU de la siguiente manera:
 1. Si el cuanto expira antes que la ráfaga, su prioridad disminuye en 1.

2. Si la ráfaga acaba antes que el cuanto, su prioridad aumenta en 1.
3. Si el proceso es expulsado de la CPU por otro de más prioridad su prioridad no varía
4. Dado que esto es un ejemplo y ponemos ráfagas representadas por números enteros (cosa que no ocurre en un sistema real) supondremos que cuando el instante de terminación de la ráfaga coincide con el de terminación del cuánto, es el cuánto lo que termina antes y, por tanto, la prioridad del proceso disminuye

Mostrar cómo sería la planificación para los procesos que se muestran en el cuadro. Las ráfagas e/s se muestran entre paréntesis, así 4(3)4 representa 4 cpu + 3 e/s + 4cpu

Proceso	Ráfagas	Prioridad inicial	Instante de llegada
P1	3(4)3	2	0
P2	1(2)1(1)3	4	1
P3	4(2)1	3	3

SOLUCION:

CPU	P1	P2	P1	P3	P2	P3	P2	P2	P2	P3	P3	P1	-	P3	-	-	P1	P1	P1		
E/S			P2	P2		P2						P3	P3 P1	P1	P1	P1					
Pri P1	2											3								*	
Pri P2		4	5			4			3	*											
Pri P3				3								4			*						

Instante 0: P1 (pri=2) es el único proceso listo, obtiene la cpu

Instante 1: Llega P2 (pri=4) con mas prioridad: expulsa a P1. Prioridad de P1 no varia (r6.3)

Instante 2: P2 abandona CPU antes de termina cuánto: su prioridad aumenta. P1 obtiene CPU

Instante 3: P3 llega y obtiene CPU

Instante 4: P2 termina e/s y obtiene la CPU

Instante 5: P2 termina ráfaga pasa a e/s. Su prioridad disminuye (r6.1 y 6.4). P3 obtiene CPU

Instante 6: P2 termina e/s y obtiene CPU

Instante 8: P2 agota el cuanto (2), disminuye su prioridad a 3, pero como tiene ña misma que P3, y está en CPU prosigue

Instante 9: P2 termina. P3 obtiene CPU

Instante 11: P3 termina rafaga antes que cuánto y su prirodad aumenta. P1 obtiene n CPU

Instante 12: P1 acaba ráfaga antes que cuánto: su prioridad aumenta

Instante 13: P3 termina e/s obtiene CPU

Instanta 14: P3 termina, P1 sigue en e/s

Instante 16: P1 termina e/s y obtiene CPU

Instante 19: P1 termina

5.-Considere los siguientes procesos con tiempos de llegada y ráfagas de CPU

Proceso	Tiempo de Llegada	Tiempo de CPU
A	0	3
B	1	5
C	3	2
D	10	6

a)Se presenta una posible planificación. Dígase de qué planificación se trata, y calcúlese el tiempo de retorno promedio para esta planificación

???	A	A	A	B	B	B	B	C	C	B	D	D	D	D	D	D	
-----	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	--

b)Con estos procesos llegando en estos instantes. ¿Cuál es el mejor tiempo de espera promedio que puede obtenerse?

SOLUCION:

a)

$$Tr \text{ promedio} = \frac{3+9+6+6}{4} = 6 \quad \text{Es un RR de quanto}=4$$

b) El que proporciona SRTF, que en este caso coincide con SJF

SRTF	A	A	A	C	C	B	B	B	B	B	D	D	D	D	D	D
------	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

$$T_{\text{EsperaPromedio}} = \frac{0+4+0+0}{4} = 1$$

6.-Dos procesos A y B tienen una ráfaga de CPU de 50 ms y un proceso C tiene cuatro ráfagas de CPU de 10 ms. intercaladas con tres ráfagas de E/S a disco de 30 ms. Los procesos tienen el mismo tiempo de llegada con el orden C, A, B. Muestre el diagrama de planificación y calcule el tiempo de retorno promedio y el porcentaje de utilización de la CPU y del disco bajo las políticas de planificación de la CPU RR con quantum 50 ms y SRTF (Shortest Remaining Time First).

SOLUCION:

RR	CPU	C	A	A	A	A	A	B	B	B	B	B	C	-	-	-	C				C
	E/S		C	C	C									C	C	C		C	C	C	
SRTF	CPU	C	A	A	A	C	A	A	B	C	B	B	B	B	C						
	E/S		C	C	C		C	C	C		C	C	C								

$$RR \quad Tr = \frac{60+110+200}{3} = 123.3$$

%CPU=70 %

$$SRTF \quad Tr = \frac{70+130+140}{3} = 113.3$$

%CPU=100%

7.-Propóngase el código de un programa que liste el directorio que se le pasa como parámetro pero que no liste durante más de 10 segundos. Ayudas: puede usarse una de la llamadas “exec” para ejecutar el programa externo “ls”; la función “sleep” deja un proceso en espera los segundos que se le pasan como parámetro y “kill (pid,SIGTERM)” termina el proceso pid.

SOLUCION:

El proceso crea un hijo que es el que lista el directorio, espera 10 segundos y luego termina el proceso hijo por medio de una señal

```
main (int argc, char * argv[])
{
    pid_t p;

    if (argv[1]==NULL) {
        printf ("Nada que listar\n");
        return;
    }
    if ((p=fork())==0) { /*proceso hijo*/
        execlp ("ls"."ls", "-l", argv[1],NULL);
        exit (255); /*por si falla execlp */
    }
    sleep (10);
    kill (p, SIGKILL);
}
```