

PAUTA DE CORRECCIÓN  
CERTAMEN 1 - 2010  
SISTEMAS DE COMPUTACIÓN

Wenceslao Palma M. <wenceslao.palma@ucv.cl>

1. (a) Un proceso se puede ejecutar en modo usuario y/o modo kernel. Por ejemplo, un proceso pasa de modo usuario a modo kernel cuando se ejecuta una llamada al sistema la cual incluye instrucciones privilegiadas. En cambio un cambio de contexto ocurre cuando un proceso que se está ejecutando en modo usuario o kernel abandona el procesador y otro proceso ocupa su lugar.
- (b) La importancia radica en que una llamada al sistema permite a un proceso el acceso a un servicio proporcionado por el kernel. Esta situación provoca que un proceso pase de modo usuario a modo kernel. Cuando la llamada al sistema finaliza el proceso pasa a modo usuario.
- (c) La principal desventaja de este algoritmo es la inanición que pueden sufrir procesos largos si llegan continuamente procesos más cortos. Una forma de resolver este problema es mover el proceso que podría sufrir inanición hacia una cola de mayor prioridad.
- (d) Un problema de exclusión mutua y sincronización siempre se puede resolver ya sea con semáforos binarios o con semáforos generales. Sin embargo en ciertas problemas como el problema del productor/consumidor una solución con semáforos binarios requiere más esfuerzo de codificación y está mas propensa a errores. Por ejemplo, con un semáforo binario es más difícil representar la cantidad de procesos que se pueden ejecutar sin bloquearse.

2. Proceso	Llegada	Tiempo de Servicio
A	0	3
B	1	5
C	3	2
D	9	5
E	12	5

FCFS (9 ptos.)

	0	3	8	10	15	20
A	XXX					
B		XXXX				
C			XX			
D				XXXXX		
E					XXXXX	

	A	B	C	D	E	Media
Tfinalizacion	3	8	10	15	20	
Tretorno	3	7	7	6	8	6,2
Tr/Ts	1	7/5	3,5	6/5	8/5	1,74

RR (q=1) (9 ptos.)

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
A	X	X			X																	
B		X	X			X	X		X													
C				X			X															
D								X		X		X		X		X		X				
E											X		X		X		X		X	X		

	A	B	C	D	E	Media
Tfinalizacion	6	11	8	18	20	
Tretorno	6	10	5	9	8	7,6
Tr/Ts	2	2	2,5	9/5	8/5	1,98

Comentario (2 ptos.)

3. (a)

```
int girar(int n){
    wait(s);
    if (saldo-n>=0){
        saldo = saldo-n;
        ok = 1;
    }else
        ok = 0;
    signal(s);
    return ok;
}
```

```
void depositar(int n){
    wait(s);
    saldo +=n;
    signal(s);
}
```

```
semaforo s=1;
void main{
    parbegin(depositar(n),girar(n));
}
```

(b) #!/bin/bash

```
size='wc -c datos.txt|awk '{print $1}''
echo $size
while [ 1 ];do
Newsize='wc -c datos.txt|awk '{print $1}''

while [ "$size" -eq "$Newsize" ];do
    size=$Newsize;
    Newsize='wc -c datos.txt|awk '{print $1}''
done
echo "tienes un email!"
size=$Newsize;

done
```