

SISTEMAS OPERATIVOS: AYUDANTÍA 4

Ayudante: Yerko Ortiz

Objetivo de la ayudantía: Repasar conceptos referentes a procesos, calendarización y threads.

1. CALENDARIZACIÓN

Llene la tabla según el algoritmo que corresponda

- Earliest Deadline First

Earliest Deadline First							
Process	Arrive Time	Burst	End Time	Turnaround	Execution Start	Response Time	Deadline
P1	0	300					500
P2	50	50					350
P3	75	100					450

- Round Robin ($Q = 20$)

Round Robin						
Process	Arrive Time	Burst Time	End Time	Turnaround	Execution Start	Response Time
P1	45	2400				
P2	15	60				
P3	25	90				
P4	0	45				

- Shortest Job First

Shortest Job First				
Process	Arrive Time	Burst Time	Turnaround	Execution Start
P1	15	60		
P2	45	2400		
P3	25	90		
P4	0	45		

2. THREADS

2.1. Implementación

Diseñe e implemente un programa en C que cree un thread, donde dicho thread imprima "Hola, mundo!" y luego realice un sleep durante 10 segundos.

2.2. Conceptos

- ¿Cuál es la diferencia entre concurrencia y paralelismo?
- ¿Cuál es la diferencia entre un proceso y un thread?
- ¿Qué es multithreading?

3. PROCESOS

Dibuje el árbol de procesos y escriba el output asociado a la ejecución del programa `exec.c`.

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

int main(void)
{
    int i, n;
    execl("./asdas", NULL);
    return 0;
}
```

asdas.c

```
#include <stdio.h>
#include <unistd.h>
int main(void)
{
    int i, n;
    for(i = 0, n = 0; i < 3; ++i, ++n) {
        printf("asdas %d\n", n);
        fork();
    }
    return 0;
}
```

4. SHELL

Ejecute el comando `ps -ef | awk '$3 == 1'`, ¿qué es lo que el comando muestra en la terminal?.

Modifique dicho comando para que muestre el nombre del proceso, el pid y el ppid.

Para la parte final investigue como hacer un comando para muestre la lista de los procesos que son daemons.

Gracias por su atención!

*“Program testing can be used to
show the presence of bugs, but
never to show their absence!”*

Edsger W. Dijkstra