

Asignación No.1

Descripción general

A continuación se definen los lineamientos de una actividad teórico-práctica para los estudiantes del curso sistemas operativos, en particular, se evalúa el dominio sobre algunos conceptos básicos, definiciones sobre la administración de CPU, procesos, planificación y la implementación de un simulador utilizando C o C++

Objetivos

Durante el desarrollo de las actividades se logra:

1. Enumerar los hitos de la evolución de los sistemas informáticos.
2. Identificar la noción de núcleo de un sistema operativo.

Antes de empezar

Realice la lectura de:

- Operating System Concepts 9th Silberschatz and Galvin
 - Capítulo 1, 2 y 3
- Sistemas Operativos Modernos de Tanenbaum
 - Capítulo 1 y 2
-

Actividad No. 2 Conceptualización [30%]

Después de realizar las [lectura recomendada](#) en este documento, responda con sus propias palabras, de manera concisa y puntual las siguientes preguntas.

1. Enumere los hitos de la evolución de los sistemas informáticos
2. Cuales son los cuatro componentes de un sistema de cómputo, describa cada uno de ellos
3. ¿Qué diferencia hay entre un núcleo monolítico y un micronúcleo?
4. Realice la definición de Sistema Operativo utilizando las dos perspectivas
5. ¿Para qué sirven las llamadas al sistema?
6. ¿Que es un sistema operativo multiprogramado?
7. ¿Que es un proceso?
8. ¿Cuáles son los estados de un proceso?
9. ¿Qué información se guarda en el PCB asociado a un proceso?
10. ¿Cuáles son las actividades principales de un sistema operativo en relación con la gestión de procesos?

Actividad No. 2 Sesiones prácticas [30%]

Incluya como parte de esta entrega las sesiones prácticas: *Basic Linux commands* y *Docker Commands*. Realice, responda y presente los puntos descritos en cada enunciado. Además, anexe evidencias que respalden el desarrollo de las actividades, como capturas de pantalla, códigos utilizados y cualquier otro material requerido.

Actividad No. 3 Simulador en C++ [40%]

Cree un programa que simule el ciclo básico de instrucción utilizando el lenguaje de programación C

1. El conjunto de instrucciones que admite el simulador está dado a continuación

SET - MEM - *SET D1 X NULL NULL*, Store X value in D1 memory address. where X is an immediate, direct or constant value. When SET instruction is read the X value is stored in Memory without processor execution.

LDR - LOAD - *LDR D3 NULL NULL* Load the value in D3 memory address and puts in accumulator register

ADD - ADDITION - There are three ways: *ADD D1 NULL NULL*, adds the value in D1 memory address to loaded value in accumulator register. *ADD D1 D3 NULL* Load the value in D1 memory address in the accumulator register and add to found value in D3 memory address. *ADD D1 D3 D4* same that *ADD D1 D3* but puts the result in D4

INC - INCREMENT - *INC D3 NULL NULL* Load the value in D3 memory address adds 1 and store in same address

DEC - DECREMENT - *DEC D3 NULL NULL* Load the value in D3 memory address adds 1 and store in same address

STR - STORE - *STR D3 NULL NULL* Read the value in accumulator register and puts in D3 memory address

SHW - SHOW - *SHW D2 NULL NULL* SHW D2 NULL NULL: Displays the value at memory address D2. SHW ACC: Displays the value in the accumulator register. SHW ICR: Displays the value in the Instruction Counter Register. SHW MAR: Displays the value in the Memory Address Register. SHW MDR: Displays the value in the Memory Data Register. SHW UC: Displays the state of the Control Unit.

PAUSE - PAUSE - *PAUSE NULL NULL NULL* Halts the instruction cycle, allowing inspection of register values and memory at any time.

END - FINISH READING INSTRUCTION

2. El simulador debe leer archivos de texto plano con instrucciones ejecutables. Ej. A continuación se presenta el contenido de un archivo de texto plano llamado "programa1.txt"

```
SET D5 12 NULL NULL
SET D2 23 NULL NULL
SET D8 3 NULL NULL
SET D3 5 NULL NULL
LDR D2 NULL NULL NULL
ADD D5 NULL NULL NULL
ADD D8 NULL NULL NULL
STR D3 NULL NULL NULL
LDR D3 NULL NULL NULL
ADD D2 NULL NULL NULL
STR D2 NULL NULL NULL
SHW D2 NULL NULL NULL
END NULL NULL NULL
```

La salida del simulador será:

61

3. Cree tres "programas" (archivos de texto plano) que el simulador pueda leer, cargar en "memoria", obtener, decodificar, ejecutar y presentar el resultado.
4. Para tener en cuenta:
 - ** Solo se ejecuta un programa a la vez
 - ** Implemente la memoria principal como un arreglo
 - ** Implemente cada uno de los registros de la CPU de manera independiente

Observaciones

- La entrega se debe realizar en equipos de (4)
- Si se incluyen desarrollos adicionales para la presentación de resultados, mejor ejecución e innovación añadirá puntuación hasta (6.0)
- Si no entiende el enunciado de alguna de las actividades no dude en escribir a jefferson.amado.pena@correounivalle.edu.co