TECNOLÓGICO NACIONAL DE MÉXICO

INSTITUTO TECNOLÓGICO DE TLAXIACO

CARACTERIZACION DE EQUIPO DE COMPUTO

Subtema:

4.1. Estrategias de *scheduling*.

Presenta:

ANA KIMBERLY HERNANDEZ PEREZ 22620053

AMELI REYES HERNANDEZ 22620050

Asignatura:

ARQUITECTURA DE COMPUTADORAS

Carrera:

INGENIERIA EN SISTEMAS COMPUTACIONALES.

Docente:

EDWARD OSORIO SALINAS.

Tlaxiaco, Oax., 27 de octubre de 2024.

"educación, ciencia y tecnología, progresos día con día"

OBJETIVO.

El objetivo de esta práctica es que el alumno simule la administración de procesos/tareas en la CPU. Para ello, deberá investigar el funcionamiento de un sistema operativo y realizar un resumen de los pasos que se llevan a cabo en la administración de procesos/tareas en la CPU.

Funcionamiento de un Sistema Operativo en la Administración de Procesos/Tareas.

Un sistema operativo (SO) es el software principal que administra todos los recursos de hardware y software de una computadora, permitiendo que el usuario y los programas interactúen eficientemente con el sistema. Entre sus funciones principales está la administración de procesos, la cual se encarga de gestionar la ejecución de programas en la CPU. Esto implica coordinar y gestionar los procesos y recursos que cada aplicación necesita para funcionar.

Pasos en la Administración de Procesos/Tareas

1. Creación de procesos: Cuando el usuario o el sistema operativo inicia una nueva aplicación, se crea un proceso. Cada proceso tiene un identificador único (PID) y varios recursos asignados, como tiempo de CPU, memoria y archivos.
2. Cola de procesos: Una vez creado, el proceso se coloca en una cola de espera hasta que pueda ser ejecutado. Las colas de procesos pueden clasificarse según el estado en el que se encuentre el proceso:
    - Listo: El proceso está esperando a ser ejecutado.
    - Ejecutando: El proceso está en ejecución en la CPU.
    - Bloqueado: El proceso está esperando un recurso o evento externo.
3. Planificación de procesos (scheduling): Para decidir qué proceso debe ejecutarse en cada momento, el sistema operativo utiliza diferentes algoritmos de planificación (scheduling), como First Come First Served (FCFS), Round Robin (RR), entre otros.
4. Asignación de la CPU y cambio de contexto: La CPU se asigna a un proceso que esté en la cola de listos. Durante el cambio de contexto, se guarda el estado del

proceso anterior para poder retomar su ejecución posteriormente, y se carga el estado del proceso nuevo.

5. Ejecución y monitoreo: El proceso es ejecutado por la CPU. El sistema operativo monitorea los procesos y controla el tiempo de ejecución de cada uno para garantizar que los recursos sean compartidos equitativamente.
6. Terminación del proceso: Una vez que un proceso ha completado su tarea, el sistema operativo libera los recursos que estaban asignados, y el proceso se elimina de la cola.

Referencias en español

1. Sistemas Operativos: Administración de Procesos y Scheduling Este artículo proporciona una introducción sobre los diferentes tipos de administración de procesos en un sistema operativo, incluyendo estrategias de planificación.
2. Qué es la planificación de procesos en sistemas operativos Explica en detalle los conceptos de planificación y algoritmos como FCFS y Round Robin.
3. Introducción a la Administración de Procesos Material académico en PDF sobre la administración de procesos, el ciclo de vida de un proceso y estrategias de planificación. https://es.slideshare.net/ - Busca "Administración de procesos en sistemas operativos"
4. Administración de procesos en Sistemas Operativos - Monografías Documento detallado sobre la administración de procesos, los estados de los procesos y cómo el sistema operativo maneja los cambios de contexto.

DESARROLLO

Para el apartado de los procesos se ocupó la librería simpy en Python, por lo cual se realizó un código que nos muestra el nombre del proceso, el tiempo total de ejecución y el tiempo restante en que se actualiza. Simula la ejecución en la CPU usando quantum.
También utiliza scheduling round-Robin, lo cual toma el proceso de cola, lo ejecuta y si este tiene tiempo restante lo añade al final.

- En esta primera parte se puede observar la implementación de dichas librerías mencionadas anteriormente, se definió que tuviera solamente 5 números de procesos.

simulacion.py

```python
import simpy

QUANTUM = 2
NUM_PROCESOS = 5

class Proceso:
    def __init__(self, env, nombre, tiempo_ejecucion):
        self.env = env
        self.nombre = nombre
        self.tiempo_ejecucion = tiempo_ejecucion
        self.tiempo_restante = tiempo_ejecucion

    def ejecutar(self, cpu, quantum):
        with cpu.request() as req:
            yield req
            tiempo_proceso = min(self.tiempo_restante, quantum)
            print(f"Tiempo {env.now}: Proceso {self.nombre} ejecutando por {tiempo_proceso} unidades de tiempo.")
            yield env.timeout(tiempo_proceso)
            self.tiempo_restante -= tiempo_proceso

            if self.tiempo_restante > 0:
                print(f"Tiempo {env.now}: Proceso {self.nombre} vuelve a la cola con {self.tiempo_restante} unidades
            else:
                print(f"Tiempo {env.now}: Proceso {self.nombre} ha finalizado.")

```

```python
def planificador(env, cpu, procesos, quantum):
    while procesos:
        proceso_actual = procesos.pop(0)
        yield env.process(proceso_actual.ejecutar(cpu, quantum))

        if proceso_actual.tiempo_restante > 0:
            procesos.append(proceso_actual)


env = simpy.Environment()
cpu = simpy.Resource(env, capacity=1)

procesos = [Proceso(env, f'Proceso {i}', tiempo_ejecucion=(i+1)*2) for i in range(NUM_PROCESOS)]

env.process(planificador(env, cpu, procesos, QUANTUM))
env.run()

```

- Al ejecutar dicho código nos muestra el siguiente resultado en la terminal, donde nos da el tiempo del proceso, el numero de proceso, cuantos han terminado y cuantos procesos siguen en cola

```
Tiempo 22: Proceso Proceso 3 vuelve a la cola con 2 unidades restantes.
Tiempo 22: Proceso Proceso 4 ejecutando por 2 unidades de tiempo.
Tiempo 24: Proceso Proceso 4 vuelve a la cola con 4 unidades restantes.
Tiempo 24: Proceso Proceso 3 ejecutando por 2 unidades de tiempo.
Tiempo 26: Proceso Proceso 3 ha finalizado.
Tiempo 26: Proceso Proceso 4 ejecutando por 2 unidades de tiempo.
Tiempo 28: Proceso Proceso 4 vuelve a la cola con 2 unidades restantes.
Tiempo 28: Proceso Proceso 4 ejecutando por 2 unidades de tiempo.
Tiempo 30: Proceso Proceso 4 ha finalizado.
PS C:\Users\anaki\Documents\ARQ.COMPUTADORAS\python>
```

CONCLUSION.

Las estrategias de scheduling son fundamentales en la administración de recursos de un sistema operativo, ya que determinan el orden y la eficiencia con la que se ejecutan los procesos en la CPU. Cada algoritmo de scheduling tiene ventajas y desventajas dependiendo del contexto de uso y los requisitos del sistema. En conclusión, en sistemas modernos, frecuentemente se combinan múltiples algoritmos, o se emplean técnicas de scheduling dinámico, adaptándose a las cargas de trabajo para maximizar tanto la eficiencia como la experiencia del usuario, una buena elección de estrategia de scheduling puede mejorar significativamente el rendimiento y la respuesta del sistema, haciendo un uso óptimo de los recursos disponibles.

REFERENCIAS.

https://www.sistemasoperativos.com
https://conceptodefinicion.de/planificacion-en-sistemas-operativos
https://www.monografias.com
https://www.monografias.com