Proyecto Sistema Operativo Básico simulado en Python

Integrantes:

- Hernandez Garcia Jimena
- Ciriaco Vargas José Arturo
- Gomez Varela Daniel Imanol
- Mihal Brandon

Descripcion

El proyecto consiste en desarrollar un simulador de un sistema operativo simplificado, que gestione la planificación de procesos, la administración de memoria, y el sistema de archivos. Se tendrán que implementar varias funcionalidades en Python , simulando cómo un sistema operativo administra sus recursos.

Objetivos de Aprendizaje

- Comprender y aplicar los conceptos de planificación de procesos.
- Implementar un sistema básico de administración de memoria.
- Simular un sistema de archivos simple con comandos básicos de manipulación de archivos y directorios.

Componentes del Proyecto

1. **Planificación de Procesos**
   - Implementar varios algoritmos de planificación como **FIFO, Round Robin, SJF (Shortest Job First)**.
   - El sistema debe permitir al usuario crear "procesos" con distintos tiempos de ejecución y prioridades, y mostrar el orden de ejecución según el algoritmo elegido.
   - Incluir estadísticas como el **tiempo promedio de espera** y **tiempo de retorno** para analizar la eficiencia de cada algoritmo.
2. **Administración de Memoria**
   - Simular un esquema de **paginación** o **segmentación** en memoria, donde los procesos se carguen en "bloques" o "segmentos".
   - Implementar algoritmos de reemplazo de páginas, como **FIFO** o **LRU (Least Recently Used)**, para gestionar el reemplazo en caso de fallos de página.
   - Mostrar la tabla de páginas o el estado de la memoria tras cada carga de proceso.

3. **Sistema de Archivos**
    - Crear un sistema de archivos básico que permita crear, leer, y eliminar archivos y directorios.
    - Implementar comandos como mkdir, touch, rm, y ls, para que el usuario pueda manipular los archivos en el sistema.
    - Cada archivo puede contener texto, permitiendo operaciones de lectura y escritura en el mismo.

1. Módulo de Planificación de Procesos

Este modulo simula la ejecución de procesos usando tres algoritmos de planificación: FIFO (First In, First Out), Round Robin (RR) y SJF (Shortest Job First).

`FIFO_processes()`:

Descripción: Implementa el algoritmo de planificación FIFO. Los procesos se ejecutan en el orden en que llegaron.

Funcionamiento: Los procesos se colocan en una cola y se ejecutan de manera secuencial. El primer proceso en llegar es el primero en ser ejecutado hasta su terminación.

`RR_processes(quantum)`:

Descripción: Implementa el algoritmo Round Robin (RR). Cada proceso recibe un "quantum" de tiempo para ejecutarse, y si no termina en ese tiempo, se interrumpe y pasa al siguiente.

Funcionamiento: Se utiliza un ciclo para iterar sobre los procesos. Si un proceso no termina en su quantum, se devuelve al final de la cola para continuar su ejecución en el siguiente ciclo.

`SJF_processes()`:

Descripción: Implementa el algoritmo Shortest Job First (SJF). Los procesos con menor tiempo de ejecución se ejecutan primero.

Funcionamiento: Se ordenan los procesos en función del tiempo estimado de ejecución, y el proceso con menor duración es el que se ejecuta primero, minimizando el tiempo de espera promedio.

2. Módulo de Administración de Memoria

Este módulo simula el manejo de páginas en memoria utilizando dos algoritmos de reemplazo de páginas: FIFO y LRU (Least Recently Used).

FIFO (First In, First Out):

Descripción: En este algoritmo, las páginas más antiguas en la memoria son reemplazadas cuando se necesita espacio para cargar nuevas páginas.

Elección: Es simple de implementar y puede ser adecuado para sistemas donde los patrones de acceso no son muy complejos.

LRU (Least Recently Used):

Descripción: Este algoritmo reemplaza las páginas que no se han utilizado por más tiempo. Se basa en la idea de que las páginas que no han sido accedidas recientemente son menos propensas a ser usadas pronto.

Elección: Se elige porque tiende a ser más eficiente en entornos con patrones de acceso más predecibles y es más representativo de la gestión de memoria en sistemas reales.

3. Módulo de Sistema de Archivos

Este módulo simula operaciones básicas de un sistema de archivos, permitiendo al usuario interactuar con directorios y archivos mediante comandos como mkdir, ls, touch, rm, cd, pwd, entre otros.

El sistema de archivos permite la creación, eliminación y visualización de archivos y directorios. Los comandos simulan las acciones más comunes que realiza un usuario en un sistema operativo.

Funciones:

`mkdir(directory_name)`:

Descripción: Crea un nuevo directorio en el sistema de archivos.

Funcionamiento: Verifica si el directorio ya existe, y si no, lo crea en la estructura del sistema de archivos.

`ls()`:

Descripción: Muestra la lista de archivos y directorios en el directorio actual.

Funcionamiento: Accede a la estructura del sistema de archivos y devuelve una lista de los archivos y directorios presentes en la ubicación actual.

`touch(file_name)`:

Descripción: Crea un archivo vacío con el nombre especificado.

Funcionamiento: Si el archivo no existe, lo crea vacío en el sistema de archivos. Si ya existe, simplemente actualiza la fecha de modificación.

`rm(file_name)`:

Descripción: Elimina un archivo o directorio especificado.

Funcionamiento: Busca el archivo o directorio en el sistema de archivos y lo elimina de la estructura. Si se trata de un directorio, debe estar vacío para ser eliminado.

`cd(directory_name)`:

Descripción: Cambia el directorio actual de trabajo al especificado.

Funcionamiento: Cambia la ubicación actual dentro del sistema de archivos, actualizando la variable que mantiene la ruta del directorio activo.

`pwd()`:

Descripción: Muestra el directorio actual de trabajo.

Funcionamiento: Devuelve la ruta completa del directorio actual donde se encuentra el usuario en el sistema de archivos.

Aquí tienes una descripción de las funciones clave dentro de las clases del sistema operativo simulado. Cada función tiene su propia tarea dentro de su respectivo módulo, y su funcionamiento está diseñado para implementar los algoritmos seleccionados.

Funcionamiento General:

1. Planificación de Procesos: Los procesos se gestionan según los algoritmos de planificación seleccionados. En FIFO, los procesos se ejecutan en orden secuencial. En Round Robin, cada proceso tiene un tiempo limitado para ejecutarse y se rotan entre sí. En SJF, los procesos más cortos se ejecutan primero para optimizar el tiempo de espera.

2. Administración de Memoria: El sistema maneja el reemplazo de páginas utilizando FIFO y LRU. FIFO reemplaza las páginas más antiguas, mientras que LRU reemplaza las que no se han utilizado recientemente, intentando maximizar el rendimiento al predecir las páginas que probablemente se necesitarán más tarde.

3. Sistema de Archivos: El sistema permite al usuario interactuar con los archivos y directorios mediante comandos básicos. Los directorios se pueden crear, cambiar y eliminar, mientras que los archivos se pueden crear, ver y eliminar.

Este conjunto de funciones y clases implementa los conceptos básicos de un sistema operativo simulado y cubre los aspectos de planificación de procesos, administración de memoria y gestión de archivos, ofreciendo una visión general de cómo los sistemas operativos reales gestionan estos recursos.