

Informe del proyecto

Sistemas Operativos / Sistema Eragileak

Profesor / Irakaslea: Ander Soraluze Irureta

Alumnos / Ikasleak: Victor Moreno Arribas



INFORMATIKA
FAKULTATEA
FACULTAD
DE INFORMÁTICA

Grado en Ingeniería Informática / Informatika Ingeniaritzako Gradua

Facultad de Informática / Informatika Fakultatea

Universidad del País Vasco / Euskal Herriko Unibertsitatea



Tabla de contenidos

Memoria del proyecto.....3

Introducción 3

Diseño del sistema..... 3

Implementación del sistema 4

Política de planificación 4

Conclusiones..... 4

Proiektuaren Txostena4

Sarrera 5

Sistema Diseinua 5

Sistema Inplementazioa 5

Planifikazio Politika 6

Ondorioak..... 6

Memoria del proyecto

Introducción

El sistema simula la ejecución de procesos en una máquina con múltiples CPUs, cores e hilos hardware. En este documento, se detallará el diseño y la implementación del sistema, así como las políticas de planificación adoptadas.

Diseño del sistema

El sistema está compuesto por cuatro componentes principales: Clock, Timer, Process Generator y Scheduler/Dispatcher. Cada uno de ellos desempeña un papel fundamental en la simulación de la planificación de procesos.

- **Clock:** El Clock es el motor del simulador y se encarga de avanzar la estructura de CPUs, cores e hilos hardware. Su función principal es simular el reloj de los procesadores, donde cada ciclo de ejecución de cada hilo hardware se representa como una ejecución del Clock.
- **Timer:** El Timer genera una señal periódica mediante pulsos del Clock. Esta señal se utiliza para interrumpir periódicamente al Scheduler/Dispatcher y desencadenar la planificación de procesos.
- **Process Generator:** El Process Generator genera procesos (PCBs) de manera aleatoria con una frecuencia variable. Los procesos generados tienen un tiempo de vida asignado, que determina su duración en la simulación.
- **Scheduler/Dispatcher:** El Scheduler es responsable de planificar y realizar los cambios de contexto de los procesos. En este sistema, se ha implementado una política de planificación por prioridad. El Scheduler utiliza una cola de prioridad para organizar los procesos en orden ascendente de prioridad y asigna un tiempo de ejecución (timeSlice) a cada proceso. El Dispatcher, por el momento, no realiza ninguna tarea específica y se activa con cada interrupción del Timer.

Además de estos componentes principales, se han definido tres estructuras de datos adicionales: **PCB** (Process Control Block), **Process Queue** y **Machine**. El PCB representa la información de cada proceso y se va actualizando a medida que avanza la simulación. La Process Queue almacena el conjunto de procesos existentes en la simulación. La Machine representa la estructura de CPUs, cores e hilos hardware del sistema, y es configurable en el sistema.

Implementación del sistema

El sistema se ha implementado utilizando el lenguaje de programación C. Se han creado archivos separados para cada componente del sistema, lo que facilita la organización y modularidad del código.

1. **clock.c**: Contiene la implementación del Clock, que se encarga de avanzar la estructura de CPUs, cores e hilos hardware en cada ejecución.
2. **timer.c**: Implementa el Timer, que genera la señal periódica utilizada para interrumpir al Scheduler/Dispatcher.
3. **process_generator.c**: Contiene el Process Generator, encargado de generar procesos aleatorios con tiempos de vida variables.
4. **scheduler.c** y **dispatcher.c**: Implementan el Scheduler y el Dispatcher, respectivamente. El Scheduler realiza la planificación de procesos por prioridad, utilizando una cola de prioridad para almacenar los procesos. El Dispatcher, por el momento, no tiene funcionalidad específica.
5. **priority_queue.c**: Contiene la implementación de la cola de prioridad utilizada por el Scheduler para organizar los procesos.
6. **pcb.c**, **process_queue.c** y **machine.c**: Contienen las implementaciones de las estructuras de datos PCB, Process Queue y Machine, respectivamente.

Además de estos archivos, se ha creado un archivo `main.c` que funciona como punto de entrada del programa. En este archivo se crean los procesos, se inicializan los componentes del sistema y se inicia la simulación.

Política de planificación

En esta primera parte del proyecto, se ha implementado una política de planificación por prioridad. Los procesos se organizan en una cola de prioridad según su prioridad asignada, y se ejecutan de acuerdo con un tiempo de ejecución predefinido (`timeSlice`). Los procesos de mayor prioridad se ejecutan primero, y se realizan cambios de contexto cuando corresponde. La política de planificación por prioridad se ha elegido por su simplicidad y facilidad de implementación.

Conclusiones

En este proyecto se ha desarrollado un sistema de simulación de planificación de procesos que permite simular la ejecución de procesos en una máquina con múltiples CPUs, cores e hilos hardware. El sistema se ha implementado utilizando el lenguaje de programación C y se ha organizado en varios archivos para una mejor modularidad y mantenibilidad.

Proiektuaren Txostena

Sarrera

Sistemak prozesuen exekuzioa simulatzen du makina batean, PUZ, kore eta hardware hari anitzekin. Dokumentu horretan, sistemaren diseinua eta inplementazioa zehaztuko dira, baita hartutako plangintza-politikak ere.

Sistema Diseinua

Sistema lau osagaiz osatuta dago: Clock, Timer, Prozesu Sortzailea eta Planifikatzailea/Despacher. Osagaio nagusien bakoitzak funtzio garrantzitsua du prozesu-planifikazioa simulatzen.

- **Clock:** Clock da simulatzailearen motorra eta CPU, kor eta hardware-ilaroen egituraren aurreratzea arduratzen da. Bere funtzio nagusia prozesadoreen erlojua simulatzea da, non hardware-ilaro bakoitzaren exekuzio zikloa Clock-en exekuzio baten moduan errepresentatzen den.
- **Timer:** Timer-ak isilikoa jartzeko senala sortzen du Clock-eko kolpeak erabiliz. Seinua Planifikatzailea/Despacher-a maiztasunez ikusi eta prozesu-planifikazioa abiarazteko erabiltzen da.
- **Prozesu Sortzailea:** Prozesuak (PCB) ausaz sortzen ditu eta frekuentzia aldakorrarekin. Sortutako prozesuek bizitzako denbora izango dute, simulazioan iraupena adierazten duena.
- **Planifikatzailea/Despacher:** Planifikatzaileak prozesuak planifikatzea eta kontekstuko aldaketa egitea arduratzen da. Sistema honetan, lehentasun-planifikazio politika bat inplementatu da. Planifikatzaileak prioritatearen arabera prozesuak antolatzen ditu lehentasunaren erdia hartuta. TimeSlice izeneko exekuzio denbora bat esleitzen zaio bakoitzari. Prioritate handiagoa duten prozesuak lehenengo exekutatzen dira eta kontekstuko aldaketa egin egiten da. Despacher-a, une honetan, ez du funtzio espezifikorik betetzen eta Timer-aren interruptzio bakoitzean aktibatzen da.

Oro har, osagaio nagusien gainean, hiru datu-egitura gehigarri definitu dira: PCB (Prozesu Kontrol Blokea), Prozesu Sorta eta Makina. PCB-k prozesu bakoitzaren informazioa adierazten du eta simulazioa aurrera eramatean eguneratzen da. Prozesu Sortak simulazioan dauden prozesu multzoa gordetzen du. Makina-k CPU, kor eta hardware-ilaroak adierazten ditu sistema-an.

Sistema Inplementazioa

Sistema, main.c, clock.c, timer.c, process_generator.c eta scheduler.c izeneko fitxategi desberdinetan banatuta dago. pcb.c, process_queue.c eta machine.c fitxategiak ere erabili dira datu egiturak inplementatzeko.

Planifikazio Politika

Proiektuaren lehenengo zatian, lehentasun-planifikazio politika bat inplementatu da. Prozesuak lehentasun-ilaran antolatzen dira beren aurreiritzaren arabera eta exekuzio-denbora predefinitu baten arabera exekutatzen dira (timeSlice). Lehentasun handiagoa duten prozesuak lehenengo exekutatzen dira eta kontekstuko aldaketa egiten da. Lehentasunaren arabera prozesuak antolatzea erabaki da, inplementazioa erraza izan dadin.

Ondorioak

Proiektu honetan, prozesu-planifikazioaren simulazio-sistema garatu da, eta CPU, kor eta hardware-ilaro multzo batean prozesuak exekutatzea simulatzen du. Sistema C programaziolengoaia erabiliz inplementatu da eta eragile gehienak banatuta daude fitxategi anitzetan modularitate eta mantentze hobe izateko.