**Chương trình mô phỏng quản lý tiến trình và cấp phát bộ nhớ**

1. **Purposes:**

* Implement process control (create, block, wakeup, cancel) through simulation
* Further familiarize with the basic strategies of process scheduling and memory allocation. Further understanding of the complete architecture of the OS with the process as the core
* Improve the ability of analysis, design and programming.

1. **Knowledge:**

* Process control block (PCB): The process control block PCB contains description information, control information, and resource information about the process, and is a concentrated reflection of the dynamic characteristics of the process. The system perceives the existence of the process according to the PCB and through the changes of various variables contained in the PCB, grasps the state of the process to achieve the purpose of controlling the process activities.
* The basic state of the process: create, cancel, ready, blocked, running.
* The main primitives of process control: the process creation primitive create(); the process termination primitive kill(); the process blocking primitive block(); the process wake-up primitive wakeup(); the process scheduling primitive schedule();
* Process scheduling algorithm: FIFO; SJF; RR; high response ratio priority; priority scheduling
* Memory allocation strategy and corresponding data structure: variable partition (free partition table); page allocation (free partition table, bit map, page table)

1. **Contents:**

* Design an OS process management simulation program to simulate the functions of OS creating process primitives, blocking process primitives, waking up process primitives, terminating process primitives, and scheduling process primitives; each process is represented by a PCB, and its content can be determined according to the specific situation is set. The process scheduling algorithm can choose any one of FCFS, RR, SJF or priority. The memory allocation can adopt any one of the variable partition strategy or the page memory allocation scheme. When the process is created, it needs to allocate memory space for the process, and when the process terminates, it needs to reclaim the memory space of the process.
* During the running process, the program can provide a front-end interactive interface for submitting basic job information and displaying or printing the status of each process and the changes of related parameters (similar to the ps command in Linux), so as to observe the running status of various processes.

1. **Requirements:**

* Carry out the overall design of the program architecture and reasonable function division according to the experimental contents.
* Complete the tasks required by the experiment contents, programming implementation, development environment and tools are not limited.
* Write an experiment report, including experiment purpose, experiment contents, experiment requirements, experiment design, experiment code and running screenshots, experiment experience, etc.

1. **Report template:**

* Name
* Purposes
* Contents
* Design
* Code and running results
* Summary