

OS-Project1

Design concept

Scheduler

The scheduler and each process are executed on different cores.

The Scheduler manages policy and process in a unified manner. Because no matter what kind of scheduling method, the process must be executed in ready state

1. Sort the read process according to ready time
2. Assign a core to the scheduler
3. Let other processes share another core to ensure the scheduler can continue to run
4. Decide the scheduling method according to which policy is RR FIFO SJF PSJF
5. Design a timer to calculate the running time of each process, has decided whether to interrupt

Process

Regardless of the scheduling method, there will be the same process operation: `set_priority`. Therefore, the API is also provided in the program to provide the scheduler to control each process to ensure that the status of each process can be accurately controlled, and the scheduling method can comply with the policy.

First in first out (FIFO)

The way FIFO selects the next process is based on the ready time of each process, first run first. Since FIFO is non-preemptive, if a process is running today, wait for that process to finish. When no process is running, you can run a for loop to determine whether it is the i-th process run. The basis for the judgment is that from the processes that have not been run and are ready, find the smallest ready time, as The next one is to run of process.

Shortest Job First (SJF)

The way SJF chooses the next process is based on the execution time of each process, with the execution time smaller and running first. Since this SJF is non-preemptive, if a process is running today, wait for that process to finish. When no process is running, you can run a for loop to determine whether it is the i-th process run. The basis for the

judgment is that from the processes that have not been run and are ready, find the smallest one with the examination time as A process to run.

Preemptive Shortest Job First (PSJF)

PSJF scheduling method is very similar to SJF method, the difference is that PSJF is preemptive. Every time from the process that is ready and not yet run, find the process with the smallest execution time to run. When a process is running but another process has a smaller execution time, the process is suspended first. From the processes that have not been run and are ready, find the process with the smallest execution time to run.

Round-Robin Scheduling (RR)

The RR scheduling method is very similar to the FIFO method. The difference is that RR is preemptive, and a timeslice is defined. Each process can only run for a time within a timeslice at a time. When no process is running, it is directly from the Among the processes that are ready and not yet run, find the first process that meets the criteria to run. When a process is running but the timeslice interval arrives, the process is suspended first. From the processes that have not been run and are ready, find the next process to run.

Core version

Platform: ubuntu 16.04

Kernel version: 4.14.25

Two CPU

Comparison and discussion

Comparing actual time with theoretical time, we find that the actual time of all scheduling methods is greater than the theoretical time. The possible reasons are as follows:

In order to reduce the delay when switching processes, we use dual cores, but even with multiple cores, the virtual machine system itself will still occupy the performance of the two cores, so our actual time in each scheduling is longer than the theoretical time.

In addition, in round robin or OS systems based on priority queues, busy waiting may cause the process to repeatedly check the conditions while waiting, causing waste of resources and affecting performance.