Nachos Assignment
Project 2 - System Call & CPU Scheduling

Qi Zheng, Lin.
Advisor: Farn, Wang.

Outline

- System call
  - sleep()
- CPU scheduling
  - FCFS
  - SJF
  - Priority
- Report
- Policy

System Call

System Call - Sleep()

- Implement a system call - *sleep()*
  - userprog/syscall.h
    - Define a system call number of Sleep
  - test/start.s
    - Prepare registers for Sleep
  - userprog/exception.cc
    - Add a new case for Sleep in ExceptionHandler
    - Note the use of kernel->alarm->WaitUntil()

System Call Cont.

- /code/threads/alarm.h
- /code/threads/alarm.cc
- The WaitUntil() will be called when a thread going to sleep.
- Call the CallBack() to check which thread should wake up.
- See the comments in the file, they are helpful.
- Don’t forget the useful data structure like list in the lib folder.

System Call Cont.

- Write your own test code like test1 and test2
  - in the same way as other test code
  - Create test.c in code/test/
  - Modify the Makefile in code/test/
  - In the same way as test1 and test2
  - Type “make” in code/test/

CPU Scheduling

FCFS, SJF, Priority

CPU Scheduling

- Choose at least **ONE** of the following to implement:
  - First-Come-First-Service (FCFS)
  - Shortest-Job-First (SJF)
  - Priority • Otherwise
- Design your own test code :
  - You can find Class::SelfTest() in many classes
  - Implement some test code, and call it in SelfTest()

CPU Scheduling Cont.

- Only FCFS, SJF, priority available?
  - You can choose any scheduling algorithm from lecture
  - Specify your algorithm in the report.
- Design at least 2 test case to proof your result
  - Specify the test case setting and plot that screenshot in your report.

Hint File !!!

- To make your own SelfTest() function:
  - threads/thread.h
  - threads/thread.cc
- To call your test code in ThreadedKernel:
  - threads/kernel.cc

Hint File !!! Cont.

- Where are the schedulers?
    - threads/scheduler.h
    - threads/scheduler.cc
- Useful data structure
    - E.g. lib/list.h for SortedList.

Report & Policy

Report contents, grading policy

Report

- Report
  - Motivation and the problem analysis
  - What’s your plan to deal with the problem (high-level)
  - You can including some important code segments and comments
  - Experiment result and some discussion
  - Tell me **why and how** it happened
  - Remember there are two parts in project2
- Please saved as [Student ID]_report.pdf
  - E.g. r04921119_report.pdf

Code and Report

- Upload to CEIBA
  - Do not mail me the homework please
- Source code and report BOTH
  - create a folder and follow the structure below

```
/r04921119_Nachos2
    |___ /nachos-4.0
    |___ r04921119_report.pdf
```

  - tar zcvf r04921119_Nachos2.tar.gz ./r04921119_Nachos2

Policy

- Nachos source code: (40%)
  - Part 1: (20%)
  - Part 2: (20%)
- Report: (60%)
  - Important !!! Tell as detail as you can
  - **why you choose, what you do, and why it works**