Project 0: Introduction
CSE-150-01 Points: 15

Overview

This is an individual assignment. Each student must **demonstrate** their session's output to the TA before the deadline.

Nachos is a Java application written to perform the functions of a real operating system. The details of how the operating system implements concepts such as threading, virtual memory, and processes are exposed to you. All of the source code implementing the Nachos operating system is included with the distribution. However, this baseline implementation does not implement some very useful features. During the semester, you will complete the missing portions of the operating system and use the functionality in your projects.

In this pre-project you will familiarize yourself with installing and compiling the Nachos distribution. All instructions assume you are working on a Linux/Unix-like environment using the Bash shell. The default terminal in Mac OS should be Bash. If you are using windows, you will need to download and install Windows Subsystem for Linux. Instructions are available at: https://docs.microsoft.com/en-us/windows/wsl/install-win10.

Tasks

1. Download the Nachos distribution from the project assignment page in CatCourses and then unpack it using the following command:

   ```
   % tar xvfz nachos-java.tar.gz
   ```

   Note that this is a customized Nachos distribution for CSE 150, so you should only use the version of Nachos that we make available (i.e., do not download from other sites on the Web).
2. As mentioned earlier, Nachos is an instructional operating system for undergraduate OS classes. It is a challenging system to learn but is also as rewarding as it is challenging.

   Nachos may seem complex (it is) and confusing at first, but after using the system for a while you will become familiar and comfortable with it. There are also some advantages with using such a well-established project. First, it has been thoroughly debugged and is therefore relatively stable. Second, there is extensive documentation describing the Nachos system. Both these aspects should make it easier for you to learn and use Nachos.

   Use the resources provided in the project assignment page to start learning about Nachos and how to use it.
3. After unpacking the Nachos distribution, go into the `proj0` directory and compile Nachos for this project using the following commands:

   ```
   % cd nachos/proj0
   % make
   ```

   You can safely ignore any compiler warnings. Then run the Nachos program using the following command:

   ```
   % nachos
   ```

   The `nachos` command is a script that invokes the Java VM to run Nachos.

   IMPORTANT: You must make sure `/bin/nachos` is in your PATH for the last command to work (otherwise, you will get a `nachos: command not found` error when you execute the `nachos` script). Assuming you unpacked Nachos in the directory `/home/cse150/`, you may add the script to your PATH using the command: `export PATH=$PATH:/home/cse150/nachos/bin`. Note that the addition you made to PATH is only temporary and will be lost when you reboot, logout, or open a new terminal. To permanently change PATH, add the same line to your `~/.bashrc` file. More information can be found at: https://opensource.com/article/17/6/set-path-linux.

   Running the program causes the methods of `nachos.threads.ThreadedKernel` to be called in the order listed in `threads/ThreadedKernel.java`:

a. The `ThreadedKernel` constructor is invoked to create the Nachos kernel.
b. This kernel is initialized with `initialize()`.
c. This kernel is tested with `selfTest()`.
d. This kernel is finally “run” with `run()`. For now, `run()` does nothing, since our kernel is not yet able to run user programs.

Your session should resemble the following:

```
% cd /nachos/proj0
% make
% nachos
Picked up _JAVA_OPTIONS: -Xms128m -Xmx512m
nachos 5.0j initializing... config interrupt timer user-check grader
*** thread 0 looped 0 times
*** thread 1 looped 0 times
*** thread 0 looped 1 times
*** thread 1 looped 1 times
*** thread 0 looped 2 times
*** thread 1 looped 2 times
*** thread 0 looped 3 times
*** thread 1 looped 3 times
*** thread 0 looped 4 times
*** thread 1 looped 4 times
Machine halting!

Ticks: total 2130, kernel 2130, user 0
Disk I/O: reads 0, writes 0
Console I/O: reads 0, writes 0
Paging: page faults 0, TLB misses 0
Swap: COFF reads 0, swap reads 0, swap writes 0
Network I/O: received 0, sent 0
```

4. Trace the execution path by hand to find where the output is coming from (i.e., which classes are generating those output statements). Your job will be simple for this initial project, simply modifying a print statement. Specifically, change the Nachos output print statement:

```
*** thread n looped m times
```

to

```
*** the thread n looped m times
```

And that is it. Be sure to test your changes before you demonstrate your code.

NOTE: If you wish to use Eclipse for compiling and running Nachos, please see the **Eclipse Guide for Nachos** that is made available to you in the project assignment page.

Code Demonstration

When you are done with this assignment, demonstrate your changes to your TA. You must demonstrate before the deadline to get any credit for this project.