Project 0: Jump Start

Spring 2018

Due: Monday, April 9, at 11:59pm

Note: This is an individual assignment. Each student must submit a copy of their session's output.

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 quarter, your job will be to 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. The tutorial below should not take that much time; if you seem to be having serious problems, check Piazza and post a note if necessary.

Tasks

1. Whether you are working on a desktop in the basement lab or your laptop, login to ieng6 using an ssh client. Be sure to `prep cs120s` when you login. (If you have a `cs120s**` account, it should automatically prep.)

   ```
   % ssh ieng6.ucsd.edu
   % prep cs120s
   ```

2. Download the Nachos distribution into your home directory on ieng6, and then unpack it:

   ```
   % tar xvfz nachos-cse120-sp18.tar.gz
   ```

   Also take some time to read the instructions for all projects on the main projects page.

3. After unpacking the Nachos distribution, go into the `proj0` directory and compile Nachos for this project:

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

   You can safely ignore the compiler warnings. Then run the nachos program:

```
% nachos
```

The `nachos` command is a script that invokes the Java VM to run Nachos. It should already be in your path, and you can also find the script in `bin/nachos`. Running the program causes the methods of `nachos.threads.ThreadedKernel` to be called in the order listed in `threads/ThreadedKernel.java`:

1. The `ThreadedKernel` constructor is invoked to create the Nachos kernel.
2. This kernel is initialized with `initialize()`.
3. This kernel is tested with `selfTest()`.
4. 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 (equivalent to ../bin/nachos)
nachos 5.0j initializing... config interrupt timer user-che
*** 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
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

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

And that's it. Be sure to test your changes before you submit them.

Code Submission

For this project, each student must submit their project using a script we provide. Go to your nachos directory (and not the proj0 subdirectory) and run the `submit-proj0` script:

```
% cd nachos
% submit-proj0
```

You will have until the posted due date to submit this project. You can submit as many times as you like. The last version submitted before the due date will be the one accepted.

---

voelker@cs.ucsd.edu