EECS 211
Advanced System Software
Winter 2009

Assignment 1

Posted: January 12, 2009 (Week 2)
Due: January 21, 2009 (Week 3), 12pm

Topic: Introduction to the Nachos system

Instructions:

The goal of this first assignment is to get familiar with the Nachos instructional operating system that we will be using for this and the following assignments.

Task 1: Read the overview chapter about Nachos

To make you familiar with Nachos, read Appendix D of the text book (7th edition). Note that this chapter is provided online at the author's web site: http://cs-www.cs.yale.edu/homes/avi/os-book/os7/online-dir/Nachos.pdf.
For your convenience, a local copy is available on the course web site in the resource section: http://eee.uci.edu/09w/18410/resources/appd.pdf

Hint: Read the chapter so that you get an *overview* about the Nachos system and the possible assignments (thus, you don't need to understand every single detail at this time!). We will structure the homework assignments for this course along the lines of the assignments outlined in the chapter.

Task 2: Install the Nachos system

To get started, you need to setup the Nachos system in a working directory for EECS 211, as follows:

- We will use the EECS department servers as work platform, namely **`vivian.eecs.uci.edu`** (or, as alternatives, **`malibu.eecs.uci.edu`** and **`newport.eecs.uci.edu`**). If you do not have an account on these machines, send an email to the course instructor and an account will be created for you.
- To work on the server, you will need to connect to one of the machines through a remote shell client. The secure-shell (SSH) protocol is necessary for this. Secure shell clients for all major operating systems exist for free, please refer to the resources page on the course web pages for details: http://eee.uci.edu/09w/18410/resources.html.

- Working with the Nachos operating system requires the use of two compilers, one for the native operating system (Sun Solaris 5.9 on SPARC), and one cross-compiler for the Nachos system which is simulated to run on a MIPS processor architecture. We will use the GNU C/C++ compiler version 2.95.3 in both cases.
- To ensure that you use the correct compilers, please add the directory **`/users/faculty/doemer/eecs211/bin`** to your execution path, i.e. **`setenv PATH /users/faculty/doemer/eecs211/bin:$PATH`** This is probably best done in your setup files, i.e. file **`.cshrc`**, otherwise you will have to repeat this command every time you login.
- For your convenience, a setup script is also provided. You may simply use **`source /users/faculty/doemer/eecs211/bin/setup.csh`** (for **`csh`**) or **`. /users/faculty/doemer/eecs211/bin/setup.sh`** (for **`sh`** or **`bash`**).
- Next, you obtain a copy of the Nachos tar-archive and install it. The following commands will create a new directory for EECS211 in your home directory, copy the tar-archive for this course over from the class directory, and extract the sources.
  - **`mkdir eecs211`**
  - **`cd eecs211`**
  - **`cp /users/faculty/doemer/eecs211/nachos-3.4-patched4EECS211.tar.gz .`**
  - **`gtar xvzf nachos-3.4-patched4EECS211.tar.gz`**
  - **`cd nachos-3.4`**
  - **`more README`**
- The file **`README`** describes the details of the usual setup process for Nachos. Note that most of the steps have already been applied to the patched tar-ball, so you don't need to worry about the system setup. However, it will be helpful to read some of the documentation files listed in this file (i.e. steps 4 and 5). In particular, the files **`nachos.ps`** and **`doc/overview.ps`** will be helpful at this time.
- The next step is to compile the basic Nachos system.
  - **`cd code`**
  - **`gmake`**
- This will compile the basic system and all the assignment sub-directories. If everything goes well (there should be no errors or warnings), you should have Nachos installed now.
- Test the installation by going into thread assignment sub-directory and start the system:
  - **`cd threads`**
  - **`./nachos`**
- This final step should show logging information about two threads running alternatively 5 times. After the threads exit, the system will halt the simulated machine, show some statistics, and then terminate.

- Submit the logging output printed in the previous step as deliverable (see section Deliverables at the end of this text for details)!

Task 3: Understand the Nachos system

Before you now dig into modifying and writing code, it is very important that you *understand* the Nachos software. There is a lot of documentation provided and a lot of source code that you should actually read and understand, before you start coding yourself. So, take the time to read through it, understand it, and if anything is unclear, go back and read again. Keep in mind that this preparation will pay off multiple times later in reduced coding and debugging time!

The following is a suggestion on the items to read, and on the order in which to read them:

- **`README`** (see above)
- **`nachos.ps`** (overview, similar to the book chapter)
- **`doc/overview.ps`** (follow the suggestions at the bottom)
- **`doc/thread.ps`** (main document for the next assignment!)
- **`code/threads/*.h`** (header files, well-commented, very important)
- **`code/threads/*.cc`** (implementation sources, to be extended)

For reading the source code files, it is useful to follow the order suggested in the **`doc/thread.ps`** document.

A key to understanding the Nachos system is the given multi-threading mechanism. When you run the original code in the **`threads`** directory, you will already run two simple threads in Nachos.

Use the debugger **`gdb`** or **`ddd`** to run the program step by step (or function by function). Trace the execution path by reading through the appropriate source files. Make sure you understand what is going on especially in the function **`SWITCH`** (the debugger may show you confusing results here!). Also, run the Nachos program with different options, e.g. with the debug option **`-d`**, and notice what changes. Finally, try the option **`-rs <seed>`** and observe its effects.
(Note: additional options to Nachos are available, see the comments in file **`main.cc`**, but most do apply to later assignments only).

As deliverable, briefly describe the execution path of the unmodified Nachos program. In particular, explain the functionality of the **`Thread::Yield()`** method and its underlying **`SWITCH`** function. What is the purpose of **`SWITCH`**, and why is this function not implemented in C/C++ language?

Deliverables:

- Submit the logging output printed when running the plain Nachos installation (see end of Task 2 above) as proof that your installation works correctly (just copy/paste the log text into your email body).
- Describe the Nachos threading mechanism (briefly, about one paragraph!), including the purpose of **`Thread::Yield()`** and **`SWITCH`** (see above) in the body of your email.

Submission instructions:

To submit your homework, send an email with subject "EECS211 HW1" to the course instructor at doemer@uci.edu.

To ensure proper credit, be sure to send your email before the deadline: Wednesday, January 21, 2009, at 12:00pm (noon).

--
Rainer Doemer (ET 444C, x4-9007, doemer@uci.edu)