CS 342302 Operating Systems

Nachos Project 1

Start-up and System call

Motivation

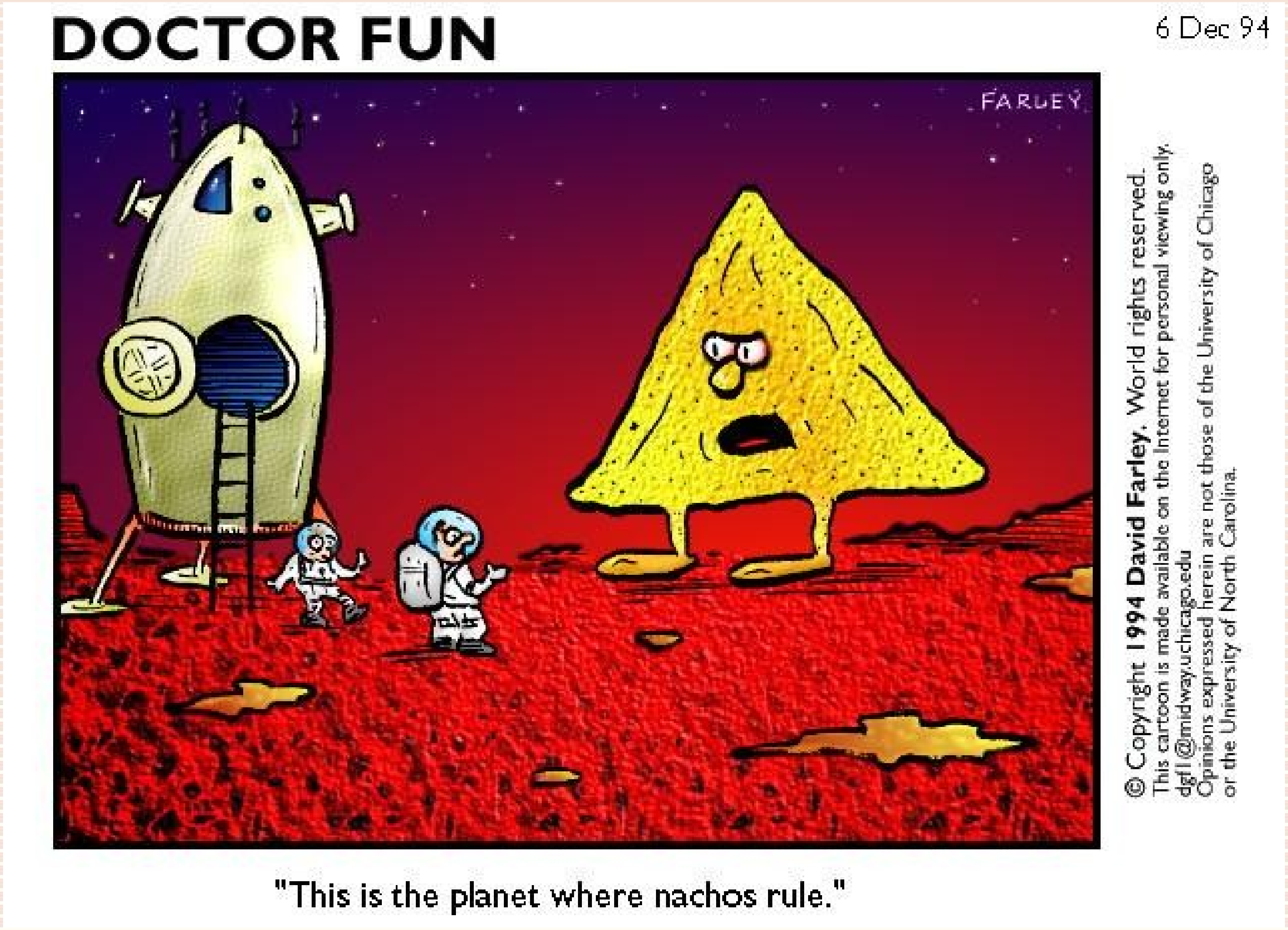
- System calls are the interfaces that user programs can ask services from kernel.
- How to add a new system call is very essential.

Objective

- Be familiar with the Nachos environment.
- Design and implement new system calls in this project.

Nachos

- http://inst.eecs.berkeley.edu/~cs162/sp08/

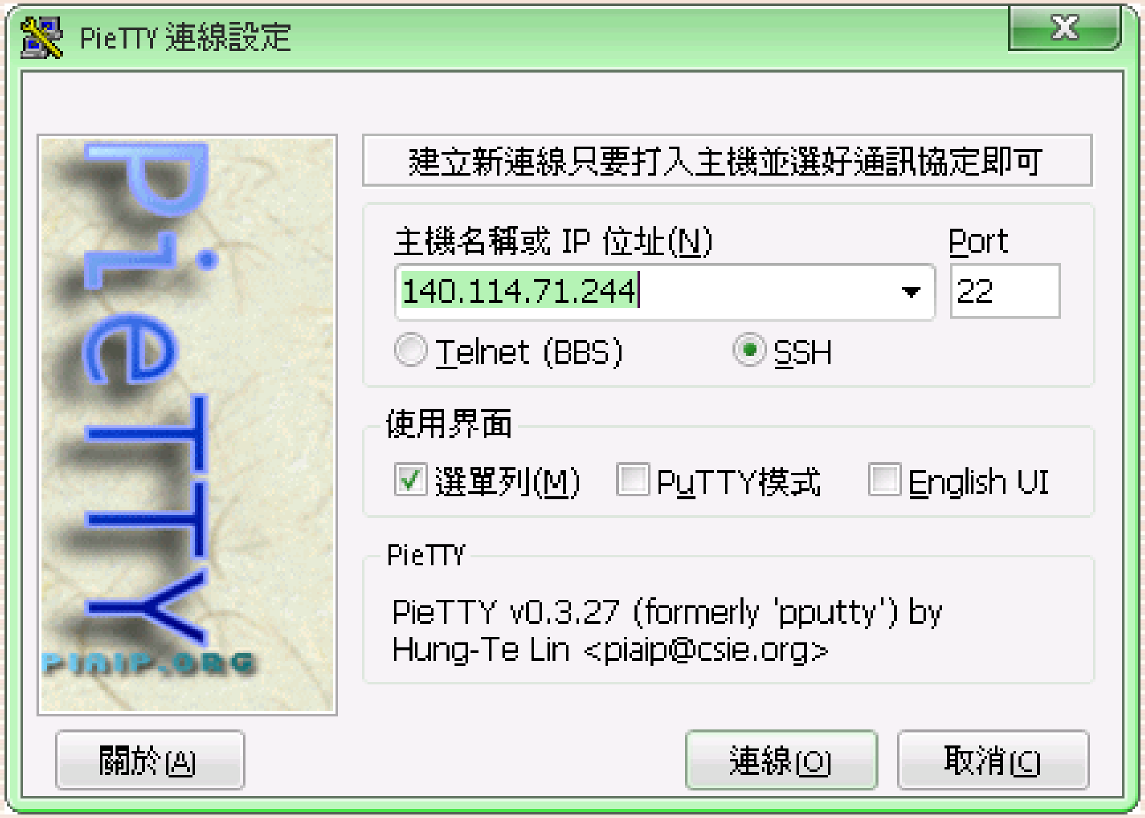

DOCTOR FUN
6 Dec 94
FARLEY
© Copyright 1994 David Farley. World rights reserved.
This cartoon is made available on the Internet for personal viewing only.
dgf1@midway.uchicago.edu
Opinions expressed herein are not those of the University of Chicago
or the University of North Carolina.
"This is the planet where nachos rule."

Platform

- You could build up your Linux environment by yourself.
- Use Linux on our server.

Login

- Download pietty (http://ntu.csie.org/~piaip/pietty/)
- Server IP : 140.114.71.233


PieTTY 連線設定

建立新連線只要打入主機並選好通訊協定即可

主機名稱或 IP 位址(N): 140.114.71.244

Port: 22

( ) Telnet (BBS) [✓ SSH]

使用界面

[✓ 選單列(M)] [ ] PuTTY模式 [ ] English UI

PieTTY

PieTTY v0.3.27 (formerly 'pputty') by
Hung-Te Lin <piaip@csie.org>

- Both the account name and the default password are your group number. (ex. os01)
- After you login, you had better change your password. (use ***passwd*** command)

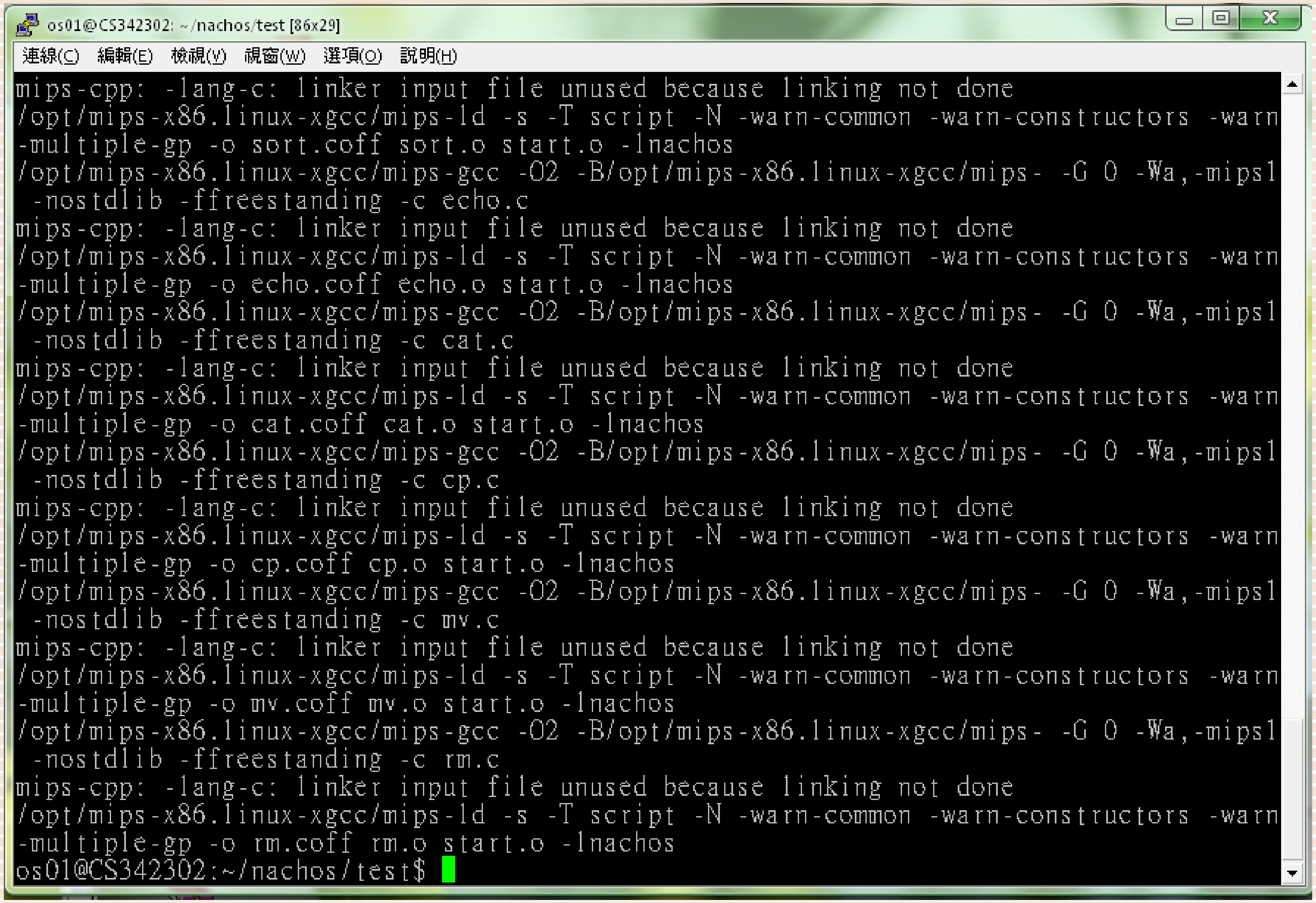
- NOTE : Some commands you have to know:

cd <directory>	: enter directory
cd ..	: go back to upper directory
cd ~	: enter home directory

Build up your Nachos environment

- Enter your home directory
  - ***cd ~***
- Download the source code
  - ***wget http://hscc.cs.nthu.edu.tw/sheujp/cs342302/nachos.tar.gz***
- Decompress it!
  - ***tar zxvf nachos.tar.gz***

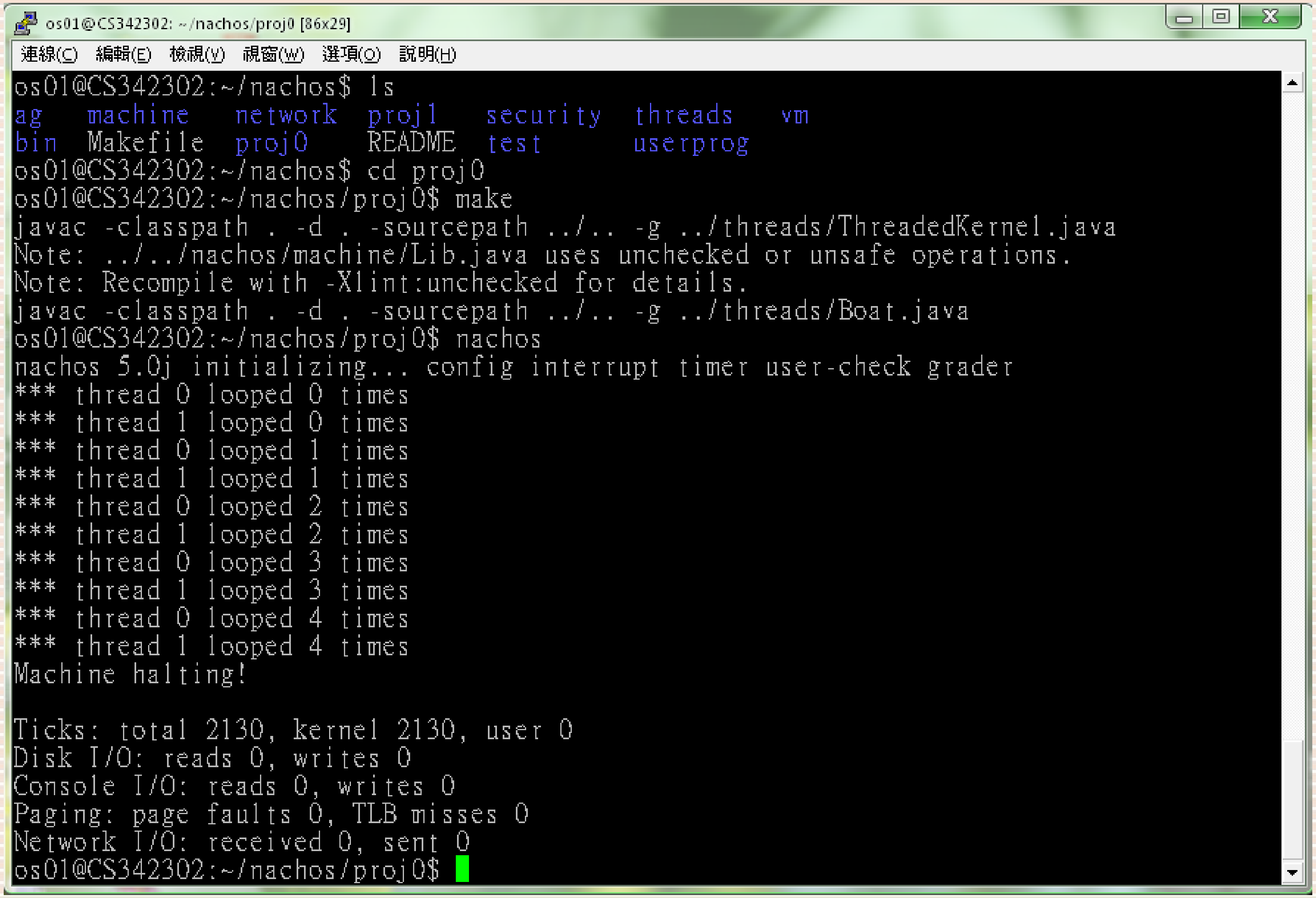
- Enter the nachos/test directory and execute **sudo make**. If there is no errors, your MIPS cross compiler environment is set correctly.
  - ***cd /nachos/test***
  - ***sudo make***
  - ***(enter your password)***


os01@CS342302: ~/nachos/test [86x29]
連線(C) 編輯(E) 檢視(V) 視窗(W) 選項(O) 說明(H)

```
mips-cpp: -lang-c: linker input file unused because linking not done
/opt/mips-x86.linux-xgcc/mips-ld -s -T script -N -warn-common -warn-constructors -warn
-multiple-gp -o sort.coff sort.o start.o -lnachos
/opt/mips-x86.linux-xgcc/mips-gcc -O2 -B/opt/mips-x86.linux-xgcc/mips- -G 0 -Wa,-mips1
 -nostdlib -ffreestanding -c echo.c
mips-cpp: -lang-c: linker input file unused because linking not done
/opt/mips-x86.linux-xgcc/mips-ld -s -T script -N -warn-common -warn-constructors -warn
-multiple-gp -o echo.coff echo.o start.o -lnachos
/opt/mips-x86.linux-xgcc/mips-gcc -O2 -B/opt/mips-x86.linux-xgcc/mips- -G 0 -Wa,-mips1
 -nostdlib -ffreestanding -c cat.c
mips-cpp: -lang-c: linker input file unused because linking not done
/opt/mips-x86.linux-xgcc/mips-ld -s -T script -N -warn-common -warn-constructors -warn
-multiple-gp -o cat.coff cat.o start.o -lnachos
/opt/mips-x86.linux-xgcc/mips-gcc -O2 -B/opt/mips-x86.linux-xgcc/mips- -G 0 -Wa,-mips1
 -nostdlib -ffreestanding -c cp.c
mips-cpp: -lang-c: linker input file unused because linking not done
/opt/mips-x86.linux-xgcc/mips-ld -s -T script -N -warn-common -warn-constructors -warn
-multiple-gp -o cp.coff cp.o start.o -lnachos
/opt/mips-x86.linux-xgcc/mips-gcc -O2 -B/opt/mips-x86.linux-xgcc/mips- -G 0 -Wa,-mips1
 -nostdlib -ffreestanding -c mv.c
mips-cpp: -lang-c: linker input file unused because linking not done
/opt/mips-x86.linux-xgcc/mips-ld -s -T script -N -warn-common -warn-constructors -warn
-multiple-gp -o mv.coff mv.o start.o -lnachos
/opt/mips-x86.linux-xgcc/mips-gcc -O2 -B/opt/mips-x86.linux-xgcc/mips- -G 0 -Wa,-mips1
 -nostdlib -ffreestanding -c rm.c
mips-cpp: -lang-c: linker input file unused because linking not done
/opt/mips-x86.linux-xgcc/mips-ld -s -T script -N -warn-common -warn-constructors -warn
-multiple-gp -o rm.coff rm.o start.o -lnachos
os01@CS342302:~/nachos/test$
```

- Note: If there are errors when you make, contact TA please. Otherwise, you couldn’t complete this project.

- Go back and enter the proj0 directory and execute **make**. It will create the directory named nachos. Finally, execute **sudo nachos** to run Nachos, you will see some lines of messages when running.
  - ***cd ..***
  - ***cd proj0***
  - ***make***
  - ***sudo nachos***

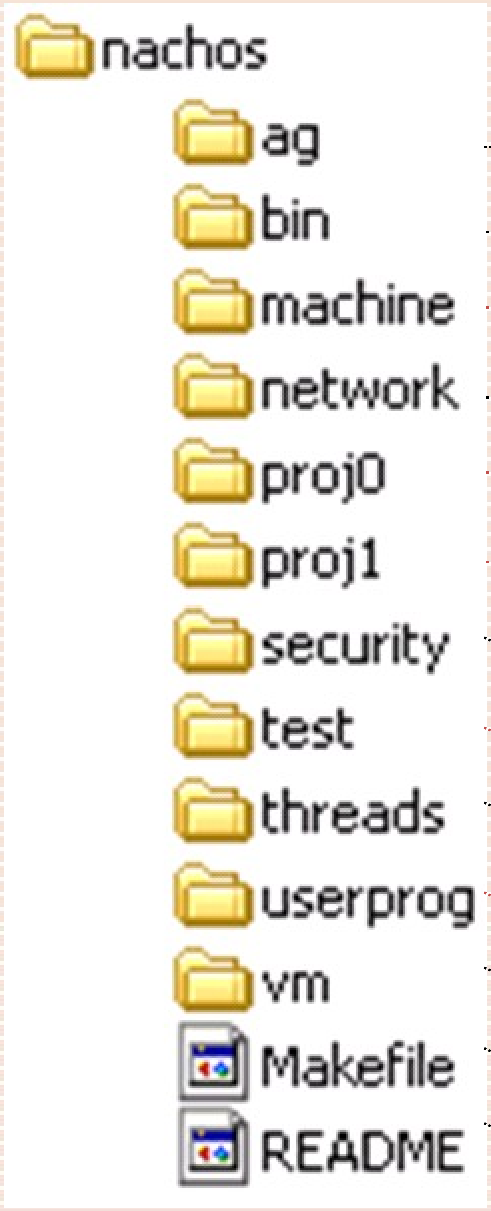

os01@CS342302: ~/nachos/proj0 [86x29]
連線(C) 編輯(E) 檢視(V) 視窗(W) 選項(O) 說明(H)

```
os01@CS342302:~/nachos$ ls
ag   machine   network  proj1   security  threads   vm
bin  Makefile  proj0    README  test      userprog
os01@CS342302:~/nachos$ cd proj0
os01@CS342302:~/nachos/proj0$ make
javac -classpath . -d . -sourcepath ../.. -g ../threads/ThreadedKernel.java
Note: ../../nachos/machine/Lib.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
javac -classpath . -d . -sourcepath ../.. -g ../threads/Boat.java
os01@CS342302:~/nachos/proj0$ nachos
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
Network I/O: received 0, sent 0
os01@CS342302:~/nachos/proj0$ 
```

- If there is no error so far, your Nachos environment is set successfully.

Trace the source codes



nachos

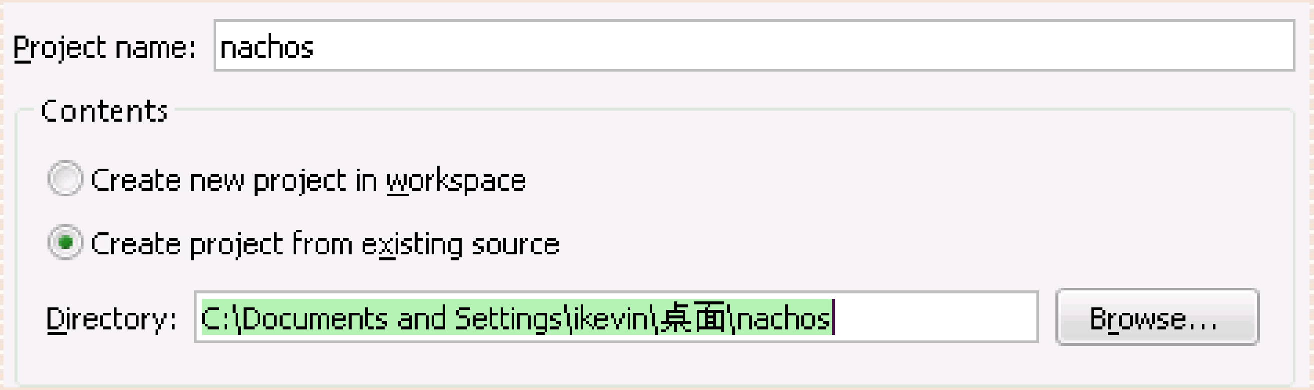
ag	auto grader
bin	Include the binary file to execute Nachos
machine	most of the action
network	I / O message queue
proj0	the sample project for you testing
proj1	The configure file of this project. You have to make Nachos in this directory.
security	tracks privilege
test	user programs
threads	threads kernel
userprog	user program kernel
vm	virtual memory
Makefile	the configure file when making Nachos
README	readme file

Modify the source codes

- You could modify the codes under Linux.
- Use **vim** to edit your files.
  - ***vim <filename>***

- Reference：大家來學 VIM （一個歷久彌新的編輯器）
  http://www.study-area.org/tips/vim/

- You could also trace and edit codes more easily with **Eclipse** under Windows.
- Download Eclipse Classic 3.4.1 (http://www.eclipse.org/downloads/)
- File -> New -> Java project



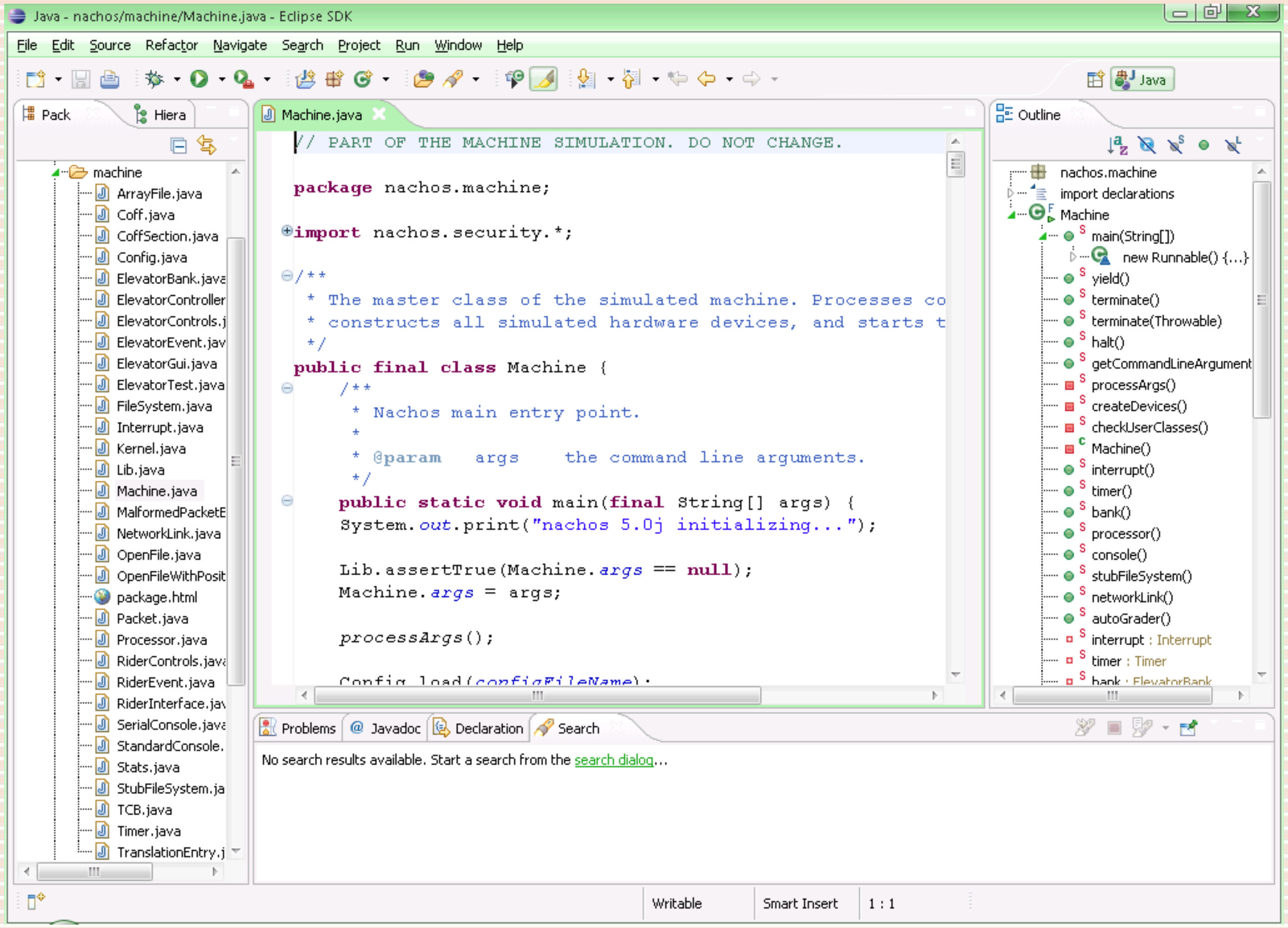
Project name: nachos

Contents

Create new project in workspace [✓ Create project from existing source]

Directory: C:\Documents and Settings\jkevin\桌面\nachos

- Give the project whatever name you want and select the location of nachos directory. Then click *Finish*.



Java - nachos/machine/Machine.java - Eclipse SDK

```
// PART OF THE MACHINE SIMULATION. DO NOT CHANGE.

package nachos.machine;

import nachos.security.*;

/**
 * The master class of the simulated machine. Processes co
 * constructs all simulated hardware devices, and starts t
 */
public final class Machine {
    /**
     * Nachos main entry point.
     *
     * @param   args    the command line arguments.
     */
    public static void main(final String[] args) {
    System.out.print("nachos 5.0j initializing...");

    Lib.assertTrue(Machine.args == null);
    Machine.args = args;

    processArgs();

    Config.load(configFileName);
```

No search results available. Start a search from the search dialog...

- After you modified the files, you should upload them to the server , then *make* and execute your Nachos.
- You could use the FTP software (ex. FileZilla) with SSH (port:22) to upload.

Related Code for User Processes

- test/start.S
  - Startup assembly code for every user program of Nachos.
- test/syscall.h
  - Definitions of the system call prototypes
- userprog/UserProcess.java
  - Encapsulates the state of a user process, including the handler for system calls and other exceptions.

How to add your System Call

- Declare a new system call in test/syscall.h
  - Define a new system call ID

    E.g., #define syscallAdd 13
  - Declare the interface for Nachos system calls, which will be called by the user program.

    E.g., int Add(int op1, int op2);
- Add the low level operation to support the new declared system call in test/start.S

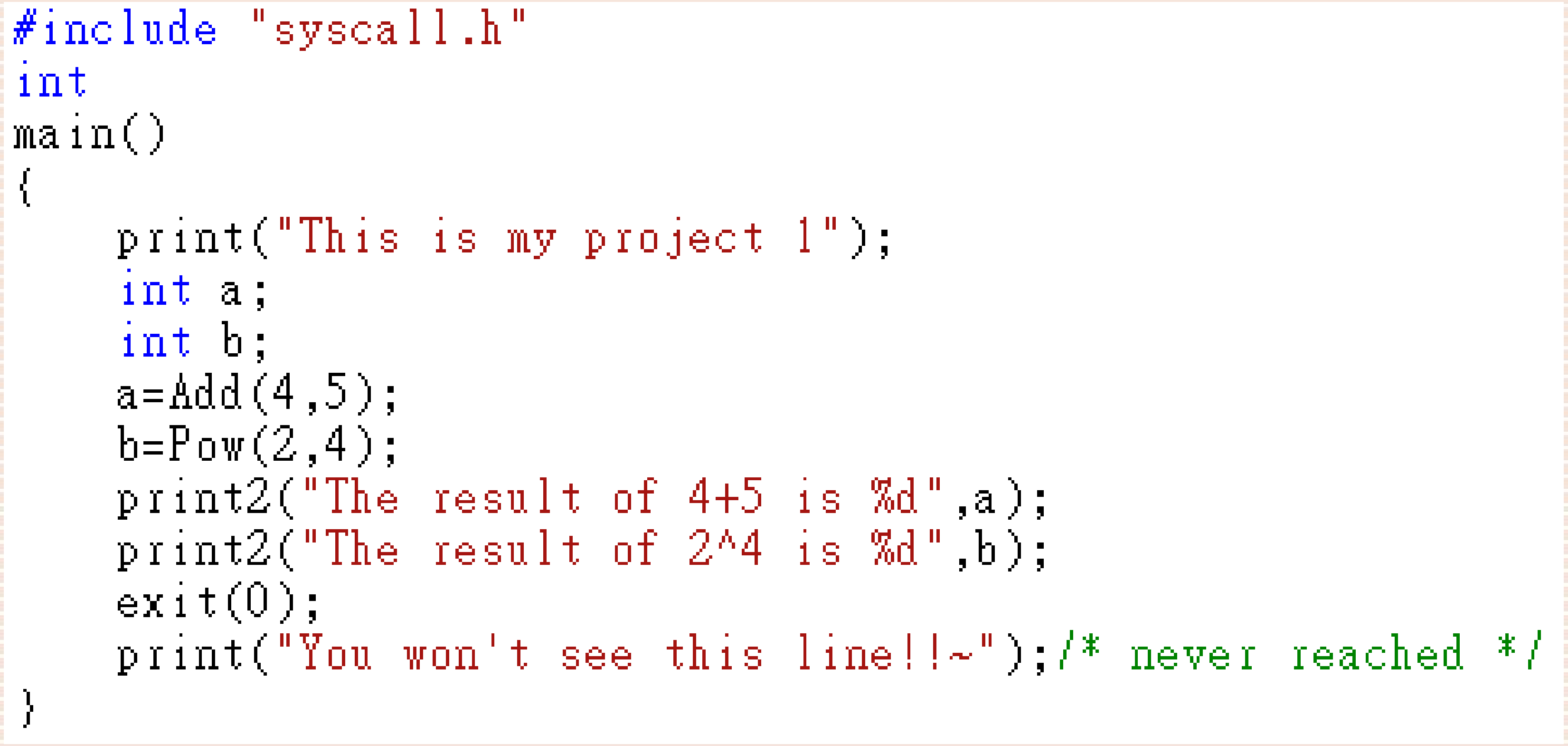
  E.g., SYSCALLSTUB(add, syscallAdd)

In handleSyscall in userprog/UserProcess.java

```
public int handleSyscall (int syscall, int a0, int a1, int a2, int a3)
 {
    switch (syscall) {
          case syscallAdd:
                    return a0+a1;
    …..
```

Basic requirement

- You have to run the following test program on your Nachos. I Assume you call it “proj1.c”.



```
#include "syscall.h"
int
main()
{
    print("This is my project 1");
    int a;
    int b;
    a=Add(4,5);
    b=Pow(2,4);
    print2("The result of 4+5 is %d",a);
    print2("The result of 2^4 is %d",b);
    exit(0);
    print("You won't see this line!!~");/* never reached */
}
```

Basic requirement (cont.)

- If you call the test file “proj1.c”.
- First, put the file under /test, and modify the file “Makefile”
  - TARGETS = halt sh matmult sort echo cat cp mv rm #chat chatserver

    ->

    TARGETS = halt sh matmult sort echo cat cp mv rm proj1 #chat chatserver
- and “sudo make” again under the /test directory.

Basic requirement (cont.)

- Then modify the line 12 of /proj1/nachos.conf
  - Kernel.shellProgram = halt.coff #sh.coff

    to

    Kernel.shellProgram = proj1.coff #sh.coff
- "make" under /proj1 one more time.
- Enter "sudo nachos" to run it.

- Implement the following system call:
  - int Add(int op1, int op2);
  - int Pow(int op1, int op2);
  - void Print(char *msg);
  - void Print2(char *msg, int value);

    You only need to parse the parameter %d.
  - void Exit(int status);

    In order to terminate user programs properly.

Bonus

- Implement other system call which is about file system I/O.
  - Creat, Open, Read, Write, Close, Unlink
- Documents about those system calls are in syscall.h
- Other system calls you think which can be bonus are exactly OK.

Hint

- In system call “print” and “print2” may use the functions below:
  - readVirtualMemoryString();
  - System.out.println();
- Instead, if you use the functions which nachos provides to print messages, you will get some bonus grade.

Submission & Grading Policy

- Code correctness 60%
  - Every system call is 15%
- Report 30%
- Bonus 20%
  - Every bonus system call is 5%

Deadline

- 10/22 PM11:59