山东大学 计算机科学与技术 学院

操作系统 课程实验报告

学号：202200101007	姓名：张祎乾	班级：22.3 班
实验题目：实验 2：Nachos 的 Makefiles		
实验学时：2	实验日期：2025/2/22	

实验目的：

该实验在目录 lab2 中完成。

（1）熟悉 Nachos 的 makefiles 的结构；

（2）熟悉如何在几个 lab 文件目录中构造相应的 Nachos 系统；

实验环境：WSL、Ubuntu

源程序清单：

无

编译及运行结果：

主要是跟着指南进行了“3.2.5 在其它目录中修改 Nachos 代码并生成修改后的 Nachos 系统”的实操，可能图没有截全，因为我在操作过程中还尝试了一些联想到的其他的操作，我是在学习完毕后补的截图。

```
zhang@zhang:~/OS/nachos-3.4/code/lab2$ ls
Makefile  Makefile.local  arch  scheduler.cc  scheduler.h
```

make 后：

```
../machine/sysdep.cc: In function 'void AssignNameToSocket(char*, int)':
../machine/sysdep.cc:331:16: warning: ISO C++ forbids converting a string constant to 'char*' [-Wwrite-strings]
  331 |     DEBUG('n', "Created socket %s\n", socketName);
      |                ^~~~~~~~~~~~~~~~~~~~~
>>> Compiling ../machine/stats.cc <<<
g++  -m32 -g -Wall -Wshadow -I../lab2 -I../threads -I../machine -DTHREADS -DHOST_i386 -DHOST_LINUX -DCHANGED -c
-o arch/unknown-i386-linux/objects/stats.o ../machine/stats.cc
>>> Compiling ../machine/timer.cc <<<
g++  -m32 -g -Wall -Wshadow -I../lab2 -I../threads -I../machine -DTHREADS -DHOST_i386 -DHOST_LINUX -DCHANGED -c
-o arch/unknown-i386-linux/objects/timer.o ../machine/timer.cc
>>> Assembling ../threads/switch-linux.s <<<
/lib/cpp  -D HOST_i386 -D HOST_LINUX ../threads/switch-linux.s > arch/unknown-i386-linux/objects/tmp.s
as  --32 -o arch/unknown-i386-linux/objects/switch-linux.o arch/unknown-i386-linux/objects/tmp.s
../threads/switch-linux.s: Assembler messages:
../threads/switch-linux.s:61: Warning: indirect call without `*'
../threads/switch-linux.s:62: Warning: indirect call without `*'
../threads/switch-linux.s:63: Warning: indirect call without `*'
rm arch/unknown-i386-linux/objects/tmp.s
>>> Linking arch/unknown-i386-linux/bin/nachos <<<
g++  -m32 arch/unknown-i386-linux/objects/main.o arch/unknown-i386-linux/objects/list.o arch/unknown-i386-linux/
objects/scheduler.o arch/unknown-i386-linux/objects/synch.o arch/unknown-i386-linux/objects/synchlist.o arch/unk
nown-i386-linux/objects/system.o arch/unknown-i386-linux/objects/thread.o arch/unknown-i386-linux/objects/utilit
y.o arch/unknown-i386-linux/objects/threadtest.o arch/unknown-i386-linux/objects/synchtest.o arch/unknown-i386-l
inux/objects/interrupt.o arch/unknown-i386-linux/objects/sysdep.o arch/unknown-i386-linux/objects/stats.o arch/u
nknown-i386-linux/objects/timer.o arch/unknown-i386-linux/objects/switch-linux.o  -o arch/unknown-i386-linux/bin
/nachos
ln -sf arch/unknown-i386-linux/bin/nachos nachos
```

touch scheduler.h

make

```
zhang@zhang:~/OS/nachos-3.4/code/lab2$ make
>>> Building dependency file for  scheduler.cc <<<
>>> Compiling scheduler.cc <<<
g++  -m32 -g -Wall -Wshadow -I../lab2 -I../threads -I../machine -DTHREADS -DHOST_i386 -DHOST_LINUX -DCHANGED -c
-o arch/unknown-i386-linux/objects/scheduler.o scheduler.cc
scheduler.cc: In member function 'void Scheduler::ReadyToRun(Thread*)':
scheduler.cc:56:16: warning: ISO C++ forbids converting a string constant to 'char*' [-Wwrite-strings]
   56 |     DEBUG('t', "Putting thread %s on ready list.\n", thread->getName());
      |                ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
scheduler.cc: In member function 'void Scheduler::Run(Thread*)':
scheduler.cc:108:16: warning: ISO C++ forbids converting a string constant to 'char*' [-Wwrite-strings]
  108 |     DEBUG('t', "Switching from thread \"%s\" to thread \"%s\"\n",
      |                ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
scheduler.cc:119:16: warning: ISO C++ forbids converting a string constant to 'char*' [-Wwrite-strings]
  119 |     DEBUG('t', "Now in thread \"%s\"\n", currentThread->getName());
      |                ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
>>> Linking arch/unknown-i386-linux/bin/nachos <<<
g++  -m32 arch/unknown-i386-linux/objects/main.o arch/unknown-i386-linux/objects/list.o arch/unknown-i386-linux/
objects/scheduler.o arch/unknown-i386-linux/objects/synch.o arch/unknown-i386-linux/objects/synchlist.o arch/unk
nown-i386-linux/objects/system.o arch/unknown-i386-linux/objects/thread.o arch/unknown-i386-linux/objects/utilit
y.o arch/unknown-i386-linux/objects/threadtest.o arch/unknown-i386-linux/objects/synchtest.o arch/unknown-i386-l
inux/objects/interrupt.o arch/unknown-i386-linux/objects/sysdep.o arch/unknown-i386-linux/objects/stats.o arch/u
nknown-i386-linux/objects/timer.o arch/unknown-i386-linux/objects/switch-linux.o  -o arch/unknown-i386-linux/bin
/nachos
ln -sf arch/unknown-i386-linux/bin/nachos nachos
```

只有目录 lab2 中的 scheduler.cc 被重新编 译,其它与 scheduler.h 有关联的文件（如 threads 目录中的 main.cc， sysch.cc，syschtest.cc，system.cc，thread.cc 及 threadtest.cc 等）没有被重新 处理。

touch ../threads/scheduler.h

make

```
     |                    ^~~~~~~~~~~~~~~~~~~~
../machine/interrupt.cc: In member function ‘void Interrupt::Idle()’:
../machine/interrupt.cc:213:16: warning: ISO C++ forbids converting a string constant to ‘char*’ [-Wwrite-string
s]
  213 |     DEBUG('i', "Machine idling; checking for interrupts.\n");
      |                ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
../machine/interrupt.cc:230:16: warning: ISO C++ forbids converting a string constant to ‘char*’ [-Wwrite-string
s]
  230 |     DEBUG('i', "Machine idle.  No interrupts to do.\n");
      |                ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
../machine/interrupt.cc: In member function ‘void Interrupt::Schedule(VoidFunctionPtr, int, int, IntType)’:
../machine/interrupt.cc:270:16: warning: ISO C++ forbids converting a string constant to ‘char*’ [-Wwrite-string
s]
  270 |     DEBUG('i', "Scheduling interrupt handler the %s at time = %d\n",
      |                ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
../machine/interrupt.cc: In member function ‘bool Interrupt::CheckIfDue(bool)’:
../machine/interrupt.cc:321:16: warning: ISO C++ forbids converting a string constant to ‘char*’ [-Wwrite-string
s]
  321 |     DEBUG('i', "Invoking interrupt handler for the %s at time %d\n",
      |                ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
>>> Compiling ../machine/sysdep.cc <<<
g++  -m32 -g -Wall -Wshadow -I../lab2 -I../threads -I../machine -DTHREADS -DHOST_i386 -DHOST_LINUX -DCHANGED -c
-o arch/unknown-i386-linux/objects/sysdep.o ../machine/sysdep.cc
../machine/sysdep.cc: In function ‘void AssignNameToSocket(char*, int)’:
../machine/sysdep.cc:331:16: warning: ISO C++ forbids converting a string constant to ‘char*’ [-Wwrite-strings]
  331 |     DEBUG('n', "Created socket %s\n", socketName);
      |                ^~~~~~~~~~~~~~~~~~~~~
```

从上面的信息可以看出，目录 threads 中的文件 scheduler.h 被修改后，目录 machine 与 threads 中涉及 scheduler.h 的源程序全部被重新编译。

grep scheduler.h *

```
zhang@zhang:~/OS/nachos-3.4/code/threads$ grep scheduler.h *
grep: arch: Is a directory
Binary file nachos matches
scheduler.cc:#include "scheduler.h"
scheduler.h:// scheduler.h
system.h:#include "scheduler.h"
```

grep system.h *

```
zhang@zhang:~/OS/nachos-3.4/code/threads$ grep system.h *
grep: arch: Is a directory
main.cc:#include "system.h"
Binary file nachos matches
scheduler.cc:#include "system.h"
synch.cc:#include "system.h"
synchtest.cc:#include "system.h"
system.cc:#include "system.h"
system.h:// system.h
thread.cc:#include "system.h"
threadtest.cc:#include "system.h"
```

将文件都复制过来后

ls

```
zhang@zhang:~/OS/nachos-3.4/code/threads$ ls
Makefile        copyright.h  main.cc       switch-linux.s  synch.h       system.cc   threadtest.cc
Makefile.local  dump         nachos        switch.h        synchlist.cc  system.h    utility.cc
arch            list.cc      scheduler.cc  switch.s        synchlist.h   thread.cc   utility.h
bool.h          list.h       scheduler.h   synch.cc        synchtest.cc  thread.h
```

touch scheduler.h

make clean

make

touch ../threads/scheduler.h

make

```
zhang@zhang:~/OS/nachos-3.4/code/lab2$ make
make: 'arch/unknown-i386-linux/bin/nachos' is up to date.
zhang@zhang:~/OS/nachos-3.4/code/lab2$ touch ../threads/scheduler.h
zhang@zhang:~/OS/nachos-3.4/code/lab2$ make
make: 'arch/unknown-i386-linux/bin/nachos' is up to date.
zhang@zhang:~/OS/nachos-3.4/code/lab2$
```

问题及收获：

收获：

1、学会了如何在 makefile.local 中控制编译的顺序，我觉得这一章实验主要让我们学会了这样一件事：

在尽可能不改动原系统的情况下，对所需改动的文件进行操作

我觉得这很重要，这能让我们在尽可能保全原项目的情况下进行改动。

2、跟着指南一边看一边操作 3.2.5，我大致明白了，就是当文件出现头文件间接包含的时候，会导致部分文件，应该被重新编译，但是并没有被重新编译的情况，然后使用 grep 命令尽可能找到全部需要重新 make 的文件，而不复制多余没必要的冗余文件。

3、Touch：touch 一遍文件，可以在不改动文件的情况下让系统误以为我们更新过文件，进而重新 make

4、Grep：命令 grep scheduler.h * 列出当前目录中包含字 符串 scheduler.h 的所有文件。

5、Makefile.common 与 Makefile.dep 的作用：包含 编译、链接 Nachos 系统所需工程文件的公共设置，被其它下子目录中的 Makefile 与 Makefile.local 所共享，我对于这两个文件的了解，大概止步于此了，虽然后文中有对这两个文件的内容的讲解，但是在是没有看太懂，不过我大概能感觉到，我们不太需要编辑这两个文件，所以主要学习 Makefile.local 去了

6、Makefile.local 一般包括下述三个变量：

CCFILES：涉及到的 C++源文件

INCPATH：根据 INCPATH 中的路径顺序依次查找要编译的头文件

DEFINES：传递给 g++的一些标号或者宏