National Tsing Hua University

Operating Systems

Introduction to Nachos

Dr. Nen-Fu Huang
Department of Computer Science, NTHU
E-mail: nfhuang@cs.nthu.edu.tw

Outline

- **Nachos**
  - **Introduction**
  - **Directory**
  - **DEBUG**
- **Basic Linux Command**

Nachos- Introduction

- Not Another Completely Heuristic Operating System, or **Nachos**
- Developed at the University of California, Berkeley.
- Written in C++ for MIPS code machine simulator.
- Runs as a process on a host OS. In our case, host OS is Ubuntu 14.04 32-bit.

Nachos- Directory

- **lib/**
  - some utilities used by Nachos code
  - Ex. sysdep.cc
- **machine/**
  - machine code simulation
  - Ex. mipssim.cc
- **threads/**
  - multi-threaded support code

Nachos- Directory

- **network/**
  - some networking functions implementations, like post office and network communication
- **filesys/**
  - codes about file system
  - There are 2 files systems in Nachos. The real one use simulated disk to hold files. While the stub one translates those system calls of file to the host OS. You can modify Makefile to switch between the 2 modes.
  - Ex. filesys.h

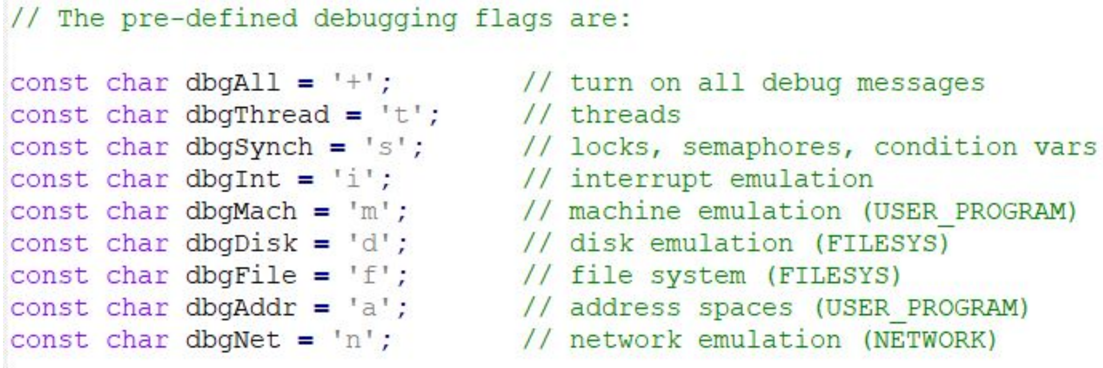
Nachos- Directory

- **userprog/**
  - OS codes to support address spaces, exception handling and so on
  - Ex. exception.cc

- **test/**
  - test programs to run on simulated machine
  - Ex. test1.c, start.s

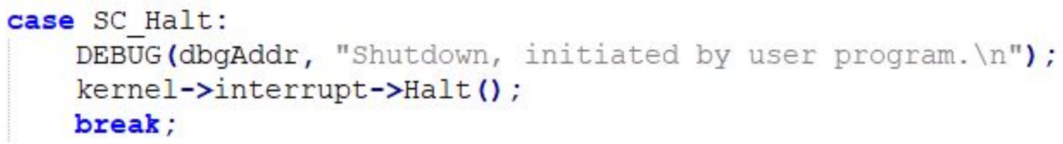
Nachos- DEBUG

- DEBUG(flag, expr)
- A special debug message implemented in Nachos.
- -flag: used to filter the type of debug message
  - All usable flags are defined in lib/debug.h.
  - You can also define your own flags!
- -expr: the message to be printed
- nachos -e <test file> -d <flag>
  - Ex. nachos -e halt -d a



```
// The pre-defined debugging flags are:

const char dbgAll = '+';         // turn on all debug messages
const char dbgThread = 't';      // threads
const char dbgSynch = 's';       // locks, semaphores, condition vars
const char dbgInt = 'i';         // interrupt emulation
const char dbgMach = 'm';        // machine emulation (USER_PROGRAM)
const char dbgDisk = 'd';        // disk emulation (FILESYS)
const char dbgFile = 'f';        // file system (FILESYS)
const char dbgAddr = 'a';        // address spaces (USER_PROGRAM)
const char dbgNet = 'n';         // network emulation (NETWORK)
```



```
case SC_Halt:
    DEBUG(dbgAddr, "Shutdown, initiated by user program.\n");
    kernel->interrupt->Halt();
    break;
```

Basic Linux Command

- **> cd <path>** #change directory
  - "..": parent directory
  - "~": home directory
  - "/": root directory
- **> ls** #list files in current directory
- **> mv <source> <destination>** #move(rename) file or directory
- **> cp [-r] <source> <destination>** #copy file
  - recursive: copy a directory, including all its files and subdirectories
- **> rm [-r] [-f] <file name>** #remove file
  - recursive: remove a directory, including all its files and subdirectories
  - force: force to remove

Basic Linux Command

- **> mkdir <directory name>** #make directory
- **> pwd** #print work directory
- **> vi / vim <file name>** #edit file with vi / vim
- **> sudo <cmd>** #super user do

Nachos- Directory

Nachos

https://homes.cs.washington.edu/~tom/nachos/

https://en.wikipedia.org/wiki/Not_Another_Completely_Heuristic_Operating_System

Linux command

http://linux.vbird.org/linux_basic/0220filemanager.php

The End