

Advanced Programming COEN 11

Intro

1

Overview

- ❑ Administrative Stuff
- ❑ What this course is all about
- ❑ Environment - Unix and C
- ❑ C - overview
- ❑ Project 1

2

Administrative Stuff

- ❑ Classes: M/W/F
- ❑ Lab: 10 projects
- ❑ TAs: Daniel, Jesse, Tingting
- ❑ Office Hours: T/Th, 10:30-11:30am
- ❑ <http://www.cse.scu.edu/~sfigueira>
 - lecture notes
 - lab projects
 - bulletin board

3

Administrative Stuff

- ❑ Grading
 - Exams
 - Quizzes (20%)
 - 2 Midterms (30%)
 - 1 Final (30%)
 - Labs (20%)

4

Administrative Stuff

Grading

➤ **Labs**

▪ **Grading:**

- On time, works with all the requirements: full credit
- Late and/or not complete: partial credit
- Not returned or does not work at all: no credit

5

Administrative Stuff

Lab projects

➤ mandatory

➤ individual

➤ require demo

Environment

➤ Unix/Mac OSX

➤ C

6

Administrative Stuff

Homeworks

➤ For practice

➤ Not for grade

7

Administrative Stuff

Text book

➤ C primer plus

Stephen Prata, 6th Edition, Sams.

8

Administrative Stuff

- ❑ **C books**
 - The C Programming Language
Kernighan and Ritchie, 2nd Edition,
Prentice Hall, 1989
 - Practical C Programming
Steve Oualline, 3rd Edition,
O'Reilly, 1997
- ❑ **Unix**
 - The UNIX Programming Environment
Brian W Kernighan and Rob Pike, Prentice Hall,
1984.

9

Administrative Stuff

- ❑ **For the Summer**
 - The Practice of Programming
by Kernighan and Pike
Addison-Wesley, 1999

10

Introduction

- ❑ **What is this course all about?**
 - Lots of programming!
 - Usage of programming techniques for solving problems
 - Advanced programming techniques
 - Pointers
 - Strings
 - Dynamic Memory Allocation
 - Recursion
 - File I/O
 - Threads

11

Environment

- ❑ **System → Unix**
 - Dominant operating system for high-end workstations and servers
 - Runs on a variety of computers
 - from phones to supercomputers
 - It was carefully designed to be simple
 - In spite of its age, it is still modern and elegant

12

Environment

❑ System → Unix

- Many OS design principles came with, or from Unix
 - Many of these design principles have been copied by other systems
- Many versions
 - Linux, BSD
 - Mac OSX, IBM AIX, HPUX, Sun Solaris, ...

13

Environment

❑ Unix

- Designed by programmers for programmers
- Interactive system
- Designed to handle multiple processes and multiple users at the same time
- System is expected to be simple, elegant, and consistent
 - Files should be just a collection of bytes
 - Commands should accept parameters in the same way
 - The commands' interface should be simple

14

Environment

❑ Unix → important feature

- Good (and fast) programmers type, don't click!!
 - Terminal mode
 - Editor: vi or vim
 - On your computer
 - Mac: terminal
 - » Remote login: ssh
 - Windows: PuTTY or cygwin

15

Environment

❑ Unix → command format

- cmd [-options] arguments

❑ To verify a command's format

- man cmd_name

16

Environment

❑ Unix → very useful commands

➤ Directories

- To create -- mkdir name_dir
- To remove -- rmdir name_dir
- To list the contents -- ls [name_dir]
- To traverse
 - To go down -- cd name_dir
 - To go up -- cd ..
 - To go up and/or down - cd path_name

17

Environment

❑ Unix → very useful commands

➤ To edit

- vi name_file

➤ To see the contents of files

- cat file_names
- more file_names

18

Environment

❑ Unix → very useful commands

➤ To manipulate files

- Remove - rm file_name
- Copy -- cp file_name_from file_name_to
- Compare -- cmp file_name_1 file_name_2
- Remote transfer
sftp uname@linux.scudc.scu.edu

19

Environment

❑ Language → C

➤ General purpose, machine-independent

➤ Developed at Bell Labs in 1972 by Dennis Ritchie

➤ Main language used for development,
especially for lower level programming

20

Environment

- C programs should have a .c extension
- To compile and execute at the terminal
 - Default
 - gcc name.c
 - ./a.out
 - or
 - gcc -o name name.c
 - ./name

21