

Short Term Plan

- Today go over expectations and course plan
- This Week
 - » Administrative, Expectations, Objectives
 - » UNIX systems overview.
 - » UNIX programming history
- Next week (lecture oriented)
 - » Introduction to the UNIX environment
 - » Create a simple C program. [theme of course: simple/basic tutorials to provide fundamentals for projects]

CSCI 1730 Systems Programming

Course Overview



Maria Hybinette, UGA

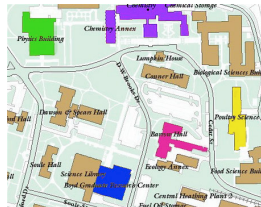
1

Maria Hybinette, UGA

2

Administration / Logistics

- Who am I?
 - » Office: Boyd 219C
- Class:
 - » Check Web Page
- maria@cs.uga.edu
- Office Hours: Thursday After Class
 - » And by e-mail appointment
- TA: multiple 1 per 30 students- check class web page for updates... office hours



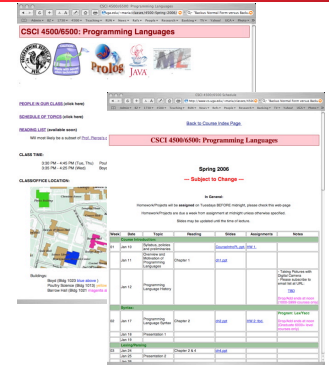
Maria Hybinette, UGA

3



The 3 Communication Links

1. Web Page (different from image on right), navigate via: www.cs.uga.edu/~maria/
2. Your responsibility
 - » Understand policies, honor code
 - » Work independently on projects & homework
 - » Check page often for updates "refresh" to get latest copy
3. Email list (tentative name)
 - TBD@listserv.uga.edu



Maria Hybinette, UGA

4

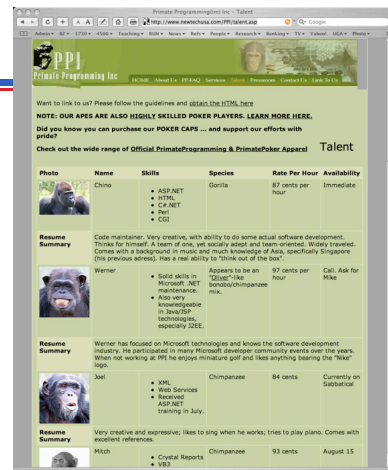
Course Objectives: Unix Systems Programming

- UNIX System Programming - So you understand or learn the strengths and limitation of the operating system and what it can do for you.
 - » Why UNIX?
 - most common OS outside the PC world, it is simple, elegant and been around for a long time (but not longer than the instructor)
- C/C++ Skills - Beefs up your resume
 - "The" language for systems programs
 - Flexible and powerful gives a lot of control left to the programmer
 - Food for thought: Why learn programming when you can get a gorilla do it for you? ..



Maria Hybinette, UGA

5



Maria Hybinette, UGA

6

The Prime Objective

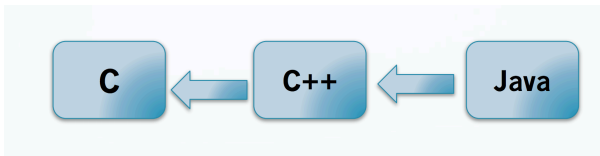
- How do we command the Operating System?
 - » File Input/ Output
 - » Processes (the programs that run on the computer)
 - » How do Processes Communicate
 - Messages
 - Files
 - Signals
 - Socket Programming
- And more...

Not the Prime Objective, but we will do this too.

- We will create systems programs in C, plain C.
- We will later learn some C++.

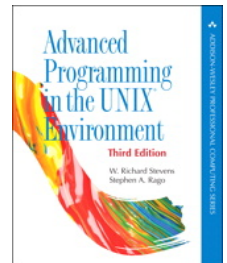
Why C?

- It is closer to the hardware?



How we're going to do it

- Read & Listen
 - » Text book.
 - » Web resources.
 - » Tutorials in class and on the web.
- Practice
 - » 9-10 Weekly short programming assignments
 - » 3-4 project (more substantial than the weekly assignments)
- Test
 - » 2 Midterms, 1 Final, Quizzes (unannounced)
- Talk and think in class, and outside!



How to get an A? B? C?... F?

- Theory 40%
 - » 2 Exams (10% each) + Final 15% + Quizzes 05% = 40%
- Practice 50%
 - » Short Assignments
 - » Projects,
- 100% attendance will raise your final grade by 2%
 - » Constructive participation on class list may raise your grade by 1%

How to get an A? B? C?... F?

100	A	79	B-
99	A	78	C+
98	A	77	C+
97	A	76	C+
96	A	75	C
95	A	74	C
94	A	73	C
93	A	72	C
92	A	71	C-
91	A-	70	C-
90	A-	69	C-
89	A-	68	D+
88	B+	67	D+
87	B+	66	D
86	B+	65	D
85	B	64	D
84	B	63	D
83	B	62	D
82	B	61	D-
81	B-	60	D-
80	B-	59	D-
		58	F

Percentage	Grade
92-100%	A
89-91%	A-
86-88%	B+
82-85%	B
79-81%	B-
76-78%	C+
72-75%	C
69-71%	C-
59-68%	D
Below 59	F

Policy on Collaboration

- Assignments/projects/summaries:
 - » Purpose: familiarization of concepts and details of game programming
 - » Work on project independently:
 - No direct sharing of code
 - No line-by-line assistant
 - No exchange of code snippets
 - » You are encouraged to ask questions of one another, and to respond to other student's questions (and especially on the email list)
- Exams:
 - » (Laptops required – Check UGA rentals)
 - » **No make-up tests** unless absence is due to **serious** illness. Doctor's diagnostic note is required. The final grade will be scaled accordingly.

Maria Hybinette, UGA

13

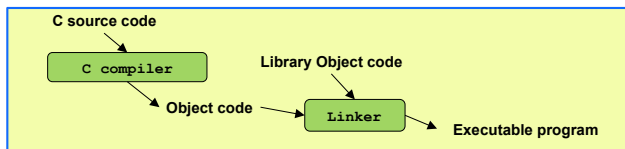
Demo

- Guess number
 - » Navigate to a directory using UNIX command lines
 - cd, ls
 - edit a file
 - compile
 - run

Maria Hybinette, UGA

14

Compilation and Linking



- You write C source code
 - » Source code is (in principle) human readable
- The compiler translates what you wrote into object code (sometimes called machine code)
 - » Object code is simple enough for a computer to “understand”
- The linker links your code to system code needed to execute
 - » E.g. input/output libraries, operating system code, and windowing code
- The result is an executable program
 - » E.g. a .exe file on windows or an a.out file on Unix

Maria Hybinette, UGA

15

What to do now ...

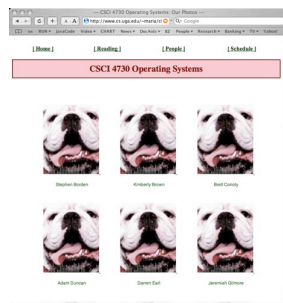
- Find class page
- www.cs.uga.edu/~maria/
 - » Go to course list, find 2014 1730 listing
- Get the Steven's book
- Read Chapter 1 & 2
- Write / Compile/ Write a simple program, e. g., hello world, or the warm-up assignment, multiple.c

Maria Hybinette, UGA

16

Homework 1

- See schedule for details...
- Digital Image -- How to get out of the dog pound (and improve your grade).
- Introductory Program:
multiple.c



Maria Hybinette, UGA

17

Introductions: Also Turn in

- Name, major, year?
- What are you hoping to learn from the class?
- What is your background?
- What type of computer platforms do you own,
 - » Model/brand, memory, processor (be specific)
- What type of projects are you interested in?
- What do you want to do when you graduate?

Maria Hybinette, UGA

18