Computer Science 2505
Computer Organization I

Modern Times
Chaplin
Instructors

Henry Monti
  Email: hmonti@cs.vt.edu
  Office: 122 McBryde
  Office Hours: see course website
Course Description

An introduction to high-level computer architecture and systems concepts, system software and programming in C.

Covers information representation, basic computer organization, fundamentals of the x86 architecture, the relationship between the C programming language and x86 assembly language, the use of debugging and other system tools, and the development of small- to medium-sized software solutions in C.

The course will focus on the x86-64 architecture.
Course Objectives

Having successfully completed this course, the student will be able to:

- understand the representation of information at the hardware level,
- understand the different layers of abstraction in a computing system, i.e., logic design, computer architecture, machine language, assembly language, high level language,
- write simple to moderately complex programs in C,
- read and understand simple programs in x86 assembly language.
- be able to apply common UNIX tools, such as gdb and objdump, to analyze, debug and correct C programs,
- be able to operate efficiently in a UNIX command-line environment.
<table>
<thead>
<tr>
<th>TAs</th>
<th>Email</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abhishek Chauhan</td>
<td>zxcve AT vt DOT edu</td>
</tr>
<tr>
<td>Nischel Kandru</td>
<td>nischelk AT vt DOT edu</td>
</tr>
<tr>
<td>Chun Wang</td>
<td>wchun AT vt DOT edu</td>
</tr>
<tr>
<td>Nick Bolin</td>
<td>nbolin AT vt DOT edu</td>
</tr>
<tr>
<td>Daniel Gil</td>
<td>dpgil AT vt DOT edu</td>
</tr>
<tr>
<td>Ethan Mattice</td>
<td>ephan AT vt DOT edu</td>
</tr>
</tbody>
</table>
Pre- and Co-requisites

CS 2114  Software Design and Data Structures
or
ECE 2574  Data Structures and Algorithms
The prerequisite must have been completed with a grade of C or higher (C- is not acceptable) or AP credit.

Math 2534  Discrete Mathematics
or
Math 3034  Intro to Proofs
Must be taken before or concurrently with CS 2505.

We will not grant any exceptions to the stated prerequisites, including the minimum grade requirements.
Textbooks

Required:

*Computer Systems: a Programmer’s Perspective, 3rd Edition*
Randal E Bryant & David R O’Hallaron
Addison Wesley ©2011

Recommended:

Mark G Sobell
Prentice Hall ©2010
(available via the Safari Database in the VT Library)
Recommended C Resources

*C Primer Plus, 6th Edition*
Stephen Prata
Developer's Library ©2014
978-0-321-92842-9

*Programming in C, 4th Edition*
Stephen G Kochan
Developer's Library ©2014
978-0-321-77641-9
Recommended References

(available ONLY at the course website)

Available via the Safari Database in the VT Library:

The Art of Debugging with GDB, DDD, and Eclipse

Write Great Code, Volume 1

Write Great Code, Volume 2
Final grades will be based on the average achieved over the following:

<table>
<thead>
<tr>
<th>Item</th>
<th>Weight</th>
<th>Dates</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assignments</td>
<td>60%</td>
<td>See course website</td>
</tr>
<tr>
<td>Tests</td>
<td>10% each</td>
<td>Tentatively:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Midterm 1: September 22</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Midterm 2: November 10</td>
</tr>
<tr>
<td>Final Exam*</td>
<td>20%</td>
<td>CTE Exam Date: Dec 14, 7:00PM - 9:00PM</td>
</tr>
</tbody>
</table>

**Grade Scale**

The usual 10-point scale will apply (subject to any curve). A final average of 90% will guarantee an A-, 80% will guarantee a B-, and so forth.

**Curve**

A grade curve may or may not be employed in this course. The application of a curve is dependent upon class performance on tests, projects and homework. The decision to utilize a curve rests entirely with the course instructors.

* Exam score will replace the lower test score, if it is higher.
In order to receive a grade of C or higher for the course, you are required to meet some minimum requirements:

**Binary bomb** You must defuse at least the first four phases

Understand something clearly: if you fail to meet the stated requirements, you will receive a grade of C- or lower for the course, regardless of your overall average.
The Meaning of Pairs Work

You will probably be allowed to work together on some assignments. If you are allowed to work in pairs, or groups, it is important you understand what we expect.

Acceptable pairs work requires:

• Each partner contributes to the analysis of the assignment, and to the derivation of a solution. This does not mean the contributions will always be equal, but both partners must be actively involved. When the solution is complete, each partner should understand the entire solution.

• The partners do not "divide" different parts of a multi-part assignment, with each working independently on his/her parts and having little or no involvement in the other parts.

• No partner "runs away" with the problem and solves it independently, not giving the other partner(s) the opportunity to contribute.

If you are allowed to work in pairs, you will choose your partner. Choose wisely.
Test Environments

- When relevant, a test environment will be specified for homework assignments.

- The C-language assignments will be compiled with gcc 4.8 (as installed on the rlogin cluster).

- Solutions will only be tested under the specified environment.

- It is the YOUR responsibility to ensure that YOUR solutions execute correctly in the appropriate environment; solutions that do not will receive substantial deductions.
The rlogin cluster is a collection of virtual machines running CentOS Linux, provided for the use of students in certain CS courses.

Some assignments may require you to work on the rlogin cluster.

The rlogin cluster has exhibited signs of strain due to the increased enrollments in the courses that make use of it, and that is unlikely to improve in the near term.

Therefore you are required to run CentOS Linux on your own computer, so that you have a stable development environment for your personal use in this course.

Notes explaining how to set up a CentOS Linux installation will be supplied on the course website.
Late Work

Due dates
Each programming project and homework assignment will have a due date and time and will include instructions for submission.

Homework (TA graded)
Usually, no late submissions will be allowed for homework assignments.

Programming Assignments (Auto-graded)
Except in the very rare case that an extension is granted, late submissions will incur a penalty per diem late penalty that will be included in the project specification. This is typically 10% per day, up to 5 days late.

Extensions
Any request for an extension must be made, preferably by email, at least 24 hours prior to the due date.

Late submissions will not be given any credit if submitted after graded assignments or solutions have been released.

Statute of Limitations
Any questions about the grading of an assignment must be raised with your instructor within one week after the graded assignment has been made available to you.
Sources of Help for This Course

General Issues
- CS 2505 classmates
- CS 2505 Forum board (forum.cs.vt.edu)
- CS 2505 TAs
- CS 2505 Instructor

Programming Language Help
- B&O text and other resources from the course website
- C language references
- CS 2505 Forum board

Lecture Instruction
Lectures will consist of presentations, applications, problems and solutions interspersed with classroom discussion.
Backups

Students are responsible for making backup copies of all their work in this (and all) courses.

Loss of work due to hard drive failure is NOT an acceptable excuse. Backup copies of files on the same hard drive are not backup copies. Backup copies of files on second hard drives are also risky. Backup copies should be maintained on two separate distinct storage mediums, (e.g., hard drives and Zip disks).

Backup copies should be maintained until after the end of the term and students have received their course grade. (The Army lives by triplicate for a reason.)

Remember: Computer systems are mechanical devices.

Network connections break.
Plan for it. It is inevitable!
The Undergraduate Honor Code pledge that each member of the university community agrees to abide by states:

“As a Hokie, I will conduct myself with honor and integrity at all times. I will not lie, cheat, or steal, nor will I accept the actions of those who do.”

Students enrolled in this course are responsible for abiding by the Honor Code. A student who has doubts about how the Honor Code applies to any assignment is responsible for obtaining specific guidance from the course instructor before submitting the assignment for evaluation. Ignorance of the rules does not exclude any member of the University community from the requirements and expectations of the Honor Code.
An exhaustive list of Honor Code violations would be impossible to present here, but among other things, each of the following is a flagrant violation of the Virginia Tech Honor Code, and violations will be dealt with severely (Honor Court):

- Working with another student to derive a common program or solution to a problem. Unless explicitly stated otherwise, there are no group assignments in this course.
- Discussing the details required to solve an assignment. You may not share solutions, or collaborate in the creation of a solution.
- Copying source code (programs) in whole or in part from someone else.
- Copying files from another student's disk or lab account even though they might be unprotected.
- Editing (computer generated) output to achieve apparently correct results.

It is acceptable to discuss an assignment with classmates in a general way, i.e., to discuss the nature of the assignment. In other words, you may discuss with your classmates what your solution is required to accomplish but not how to achieve that goal using C, x86 assembly, or other relevant tools. In no way should the individual statements of a program or the steps leading to the solution of the problem be discussed with or shown to anyone except those people cited in the following statement.
Honor Code

Feel free to discuss the homework assignments and your program source code with the teaching assistants assigned to CS 2505, the instructor, or the free tutors provided by UPE. The discussion of your program source code must be limited to these people. Note that this specifically excludes discussions of your program source code with other students (even if they are not enrolled in CS 2505), or with tutors except for those named above. Privately hired tutors are not an exception to this requirement, nor are athletic or other tutors provided by the University.

Copies of all submitted work are retained indefinitely by the Department. Submitted programs are subjected to automated analysis for detection of cheating.

If you have any question as to how the Honor Code applies to this class, remember that:

- Any work done in this class must be done on an individual basis.
- Credit will be given only for work done entirely on an individual basis.
- Do not make any assumptions as to who can provide help on a programming assignment.
- All submitted work is archived. All submitted programs will be subjected to automated cheat analysis.

Evidence indicating the violation of the policies stated above will be submitted to the Honor Court.

It is much easier to explain a poor grade to parents or a potential employer than to explain an Honor Court conviction.
In recent terms we have observed a new behavior regarding cheating.

Some of the projects we use each semester may have been used in previous offerings of the course.

A small number of students have submitted solutions that were based (in whole or in part) on solutions submitted by other students in previous offerings. Be advised:

• That is cheating.
• When we do the cheating analysis for a project, we include submissions from previous offerings as well as the current term. (We save everything.)
• If we detect this form of cheating, the students from the previous terms will also be charged with violating the Honor Code.
• The University does make provisions for cases in which a charged student has already graduated. You do not what to discover how this works, so safeguard your code for the long term!
Posting Code Online

At some point you will very likely want to make samples of the code you write available to potential employers. There is absolutely nothing wrong with that!

But if you do that, you need to be sure you're not making it easy for other students to access your code.

GitHub is an interesting service, and the free student accounts allow you to protect your posted code.

And there are alternatives...