

Spring 2009

## **CSC 2405 – Computing Systems II**

### **Syllabus**

<b>Instructor</b>	Mirela Damian, MSC 167A mirela.damian@villanova.edu Phone: (610)519-7414
<b>Office Hours</b>	M 1:30 pm – 2:30 pm Th 4:30 pm – 5:30 pm
<b>Course Meets</b>	MW 9:00 pm - 10:20 pm in Mendel G86
<b>Prerequisites</b>	CSC 1300 – Discrete Structures

### **Course Description**

This course is the second part of a two semester sequence – Computer Systems I and II. The goal of this course is to bridge the gap between high-level programming and actual computer components, such as processors, caches, operating systems, compilers, linkers, and so on. This course will teach the way computer systems work from the low level device interrupts up to processes, threads, memory management, file systems and security. Labs will be interspersed with lectures to enhance students' experience with the C language and the Unix system.

### **Student Learning Outcomes**

1. Students will demonstrate working knowledge of thread and process management and concurrency.
2. Students will demonstrate a thorough understanding of virtual memory, physical memory and cache management.
3. Students will use basic operating system services to implement medium-to-large application programs.
4. Students will be able to identify and analyze the needs and concerns of real-time systems.
5. Students will demonstrate an understanding of security risks faced by computer systems and security mechanisms designed to protect computer systems.
6. Students will enhance their knowledge of C and Unix through the development of two major projects, and will become adepts at writing simple scripts.

### **Resources**

The textbook for this class is

R. Bryant and D. O'Hallaron  
Computer Systems - A Programmer's Perspective  
Prentice Hall, ISBN: 0-13-034074-X

Other course-related materials will be posted online at

<http://www.csc.villanova.edu/~mdamian/csc2405/>

Please make sure you check the class page regularly.

## Course Requirements

1. **Assignments:** weekly, some of which will involve C programming in Unix. All assignments must be completed individually, unless otherwise noted.
2. **Quizzes:** weekly, 5-10 minutes each. The intent is to help students learn the material incrementally and know where they stand.
3. **Presentations:** teams of one or two students will present a topic of choice on system security in the final weeks of the semester. Presentations will be 15 mins each.
4. **Tests:** one midterm and one comprehensive final exam. Tests will be closed books, closed notes. However, you are allowed to bring one sheet of paper (letter size) with any information you think will help you during the exam. *Please note that notes may not be shared during the exam.*
5. **Class Attendance.** Regular attendance is expected of all students. Each student is responsible for all material, announcements, and assignments covered during any class missed.
6. **Late Policy.** All assignments are due at the beginning of the class on the due date. No credit will be given to late assignments. Exceptions to this policy will be granted only in extraordinary circumstances.

## Tentative Grading Procedure

The following allocation of points is tentative and may change during the semester:

Midterm:	20%
Final exam:	30%
Assignments:	40%
Presentation:	10%

## Academic Integrity

I encourage you to collaborate on assignments and learn from your fellow students. However, there is a fine line between collaboration and cheating. Collaboration means discussing problems and solution approaches with other students and independently writing your own answers; cheating means copying solutions from someone else or giving someone else your solutions. If you have questions about what is acceptable, please bring them to me *before* submitting your work.

Cheating, plagiarism and helping others commit these acts are all forms of academic dishonesty, and will not be tolerated. Academic misconduct could result in disciplinary action that may include, but is not limited to, suspension or dismissal. To read the entire Code on Academic Integrity, consult

<http://www.academics.villanova.edu/AcademicIntegrity.html>

## Special Arrangements

If anyone has a disability or other problems that warrant the need for special accommodation to complete the course work, please contact me at your earliest convenience.

## Tentative Course Schedule

The course schedule below is approximate and subject to change as the semester progresses. It is the responsibility of the student to learn and adjust to changes. Please **read chapter material to be covered prior to each session.**

<b>Date</b>	<b>Topic</b>	<b>References</b>
Wk 1: Jan. 12, 14	Overview. Review of relevant 2400 material.	Ch. 8.1-8.4
Wk 2: Jan. 19, 21	Signals.	Ch. 8.5
Wk 3: Jan. 26, 28	Tiny Shell. Hands-on Programming.	Class work.
Wk 4: Feb. 2, 4	Process Scheduling. Review of Threads.	Ch. 13.3-13.5
Wk 5: Feb. 9, 11	Network Programming. Web Services.	Ch. 12
Wk 6: Feb. 16, 18	System-level I/O. Robust I/O.	Ch. 11
Wk 7: Feb. 23, 25	Catching up. <b>Midterm.</b>	Class work.
<b>Mar. 2 – 9</b>	<b>Spring Break – ENJOY !</b>	
Wk 8: Mar. 9, 11	Web Proxy Project.	Class work.
Wk 9: Mar. 16, 18	Virtual Addressing. Relocation.	Ch. 7.1-7.7
Wk 10: Mar. 23, 25	Virtual to Physical Address Translation.	Ch. 10.1-10.6
Wk 11: Mar. 30, Apr. 1	Page Replacement.	Notes.
Wk 12: Apr. 6, 8	Integrating Caches and Virtual Memories	Ch. 6.3-6.4, 10.6
<b>Apr. 9 – 13</b>	<b>Easter Recess – ENJOY !</b>	
Wk 13: Apr. 15	Student Presentations on OS Security.	Web resources.
Wk 14: Apr. 20, 22		
Wk 15: Apr. 27, 29	Unix Scripting.	Class work.
Thursday, May 7	<b>FINAL EXAM 10:45 am – 1:15 pm</b>	