CSCI 513/CSCI 462 – Fall 2017
Systems Programming and Operating Systems Internals / OS Device Drivers

**Time:** MTWF 2pm

**Place:** CF 225

**Instructor:** Phil Nelson

**Office/Phone:** CF471, 650-3035

**Office Hours:** MTW 3 - 4pm, others by appointment.

**Web:** [http://facultyweb.cs.wwu.edu/~phil](http://facultyweb.cs.wwu.edu/~phil)

**E-mail:** phil.nelson@wwu.edu When you send me e-mail, please use plain text (no HTML) messages and include “CSCI 513” or “CSCI 462” in the subject of your e-mail to help your e-mail not be classified as spam.

**Web Access:** This quarter I have information for this class on the web. As they are assigned, I will put a copy of each assignment on the web. Also, other useful information will be provided via the web. See [http://facultyweb.cs.wwu.edu/~phil/classes/f17/513](http://facultyweb.cs.wwu.edu/~phil/classes/f17/513).

**Text:** McKusick, Bostic, Karels, Quarterman, *The Design and Implementation of the 4.4BSD Operating System*, Addison Wesley, 1996. ISBN 0201549794


**Other Reference:** NetBSD man pages, NetBSD web site (www.netbsd.org), MSDN library, and MSDN web site (msdn.microsoft.com).

**513 Catalog Description:** Systems software such as linkers, loaders, system utility software and operating system internals. Investigation of real operating systems, augmentation or modification of the operating system source code, process creation and management; inter process communication; process scheduling; I/O hardware and software; memory management; file system design and implementation; security and protection mechanisms.

**462 Catalog Description:** Operating System kernel programming, system call mechanisms, system call processing, file system internals, device drivers, interrupt processing, methods for controlling hardware from software. Includes study and augmentation of real Operating Systems, both open-source and closed-source.

**Graded Work:** The graded work will be 2 tests and assignments.

**Tests:** The tests are scheduled for Friday, November 3 and the regular final time of Thursday, December 14 at 8:00AM. Each test covers only 1/2 of the class and is worth 23% of your grade.
Minimum Points on Tests: To pass this class, you must earn at least 50% of all test points.

Assignments: Assignments will be worth 54% of your grade and may consist of both written work and programming. Assignments may be worth a different number of points and will contribute to “total” for the assignments. (Best estimate is that you will have about 1000 points in assignments over the quarter.)

Programs: For this class, we will be doing kernel level programming. So there really won’t be a “program” to complete, but programming tasks. These may require user level test programs to complete the task. The environments you are to use are NetBSD and Windows 10. We will be using the computers in CF 416 with Qemu software with access to the hardware. More details in class.

Class Outcomes: At the end of the class, the student should have a good understanding of basic device driver development in both a free UNIX-like environment and the Windows environment.

Outcomes: On competition of this course, students will demonstrate:

- The understanding of Kernel level development.
- The understanding of Kernel control of hardware.
- Basic understanding of interrupt driven software.
- The understanding of Kernel interaction with user space processes.
- The understanding of Device Driver Development under both an open source Operating System and a closed source operating system.

Final Assignment: The final assignment will be due on the last class of dead week, December 8, 2017.

Late Work: Work is due at the beginning of class on the day due. Work will be accepted up to TWO class meetings late and will be worth 75% of the original value. (This gives you an extra week since we meet only twice a week.) Work later than two class meetings will be worth nothing. A late final assignment is worth 75% of the original up until the start of the final test.

Grading: Grading is done by a percentage of the top score. The following is an example grade scale.

A: 100% – 90%
B: 89% – 80%
C: 79% – 65%
D: 64% – 50%

Collaboration: Each student MUST do their own programming. Original work is required. You should not see the source code of any other student, past or present, for these assignments. In many cases, you may follow code examples provided in your books or in class, but the modifications must be your original work.
Cheating: Is (obviously) not allowed. If you do cheat and are caught you will receive an F as your grade for the class. This includes ALL students knowingly involved in any cheating event. Not properly protecting your source code may be considered knowingly involved. If you give your password to your friend or allow access to your files or a machine on which your sources are stored, this can be considered knowingly involved. I use mechanical means to compare student programs, not only all students this quarter, but may include students who took this class in the past. These comparisons are used to raise the possibility of cheating, but all decisions about cheating will be made by me after inspecting the programs of all students involved.