Department and Course Number	**CEG 333**	Course Coordinator	Prabhaker Mateti
Course Title	**Introduction to Unix**	Total Credits	2

BS CE: Required; BS CS: Required.

This document was prepared by: Prabhaker Mateti Date: April 04, 2005

Catalog Data

Introduction to the use of Unix and Unix tools as a problem-solving environment. Emphasis on the shell, files and directories, editing files, user process management, compiling, and debugging.
Prerequisite: CS 241.

Text Books and Other Source Materials

PABL

Paul W. Abrahams, Bruce R. Larson, UNIX for the Impatient , Second Edition, Addison-Wesley, 1995, ISBN 0201823764.

Home Page

`www.cs.wright.edu/people/faculty/pmateti/Courses/333/` Contains notes for each week.

News Group

`wright.ceg.333`

Course Objectives

The student should have learned the following:

1. Understand the command-line user interface
2. Understand the edit-compile-test-debug cycle of program development
3. Understand what scripting is
4. Develop a feel for a Windows-alternative OS
5.

Course Content

Intro to Unix. Linux, X11, `xterm`, and KDE, Emacs, `ssh`, `sftp`
`emacs` and `bash`
`regex`; Ten Most Essential Commands: `ls`, `cd`, `mkdir`, `rm`, `...`
The UNIX File System: `ls -lisa`, `ln`, `chmod`, `file`, `size`
Unix Prog Environment: `g++`, `libc`, `env`, **Midterm Exam**

Unix Programming Environment: `indent, make, gdb, man, nm`
Text Searching and Editing: `grep` and `sed` ; `find`
`emacs`
Unix Utilities: `ps, top, kill, cat, dd`
`bash` scripting
Final Exam

Class/Laboratory Schedule:

Each week has one lecture of 50-minutes, and one practice session of 50 minutes. There is no scheduled lab. Students are expected to work in open labs as needed.
The projects contribute 30% to the final grade. There is a project in five parts worth 5+5+10+5+5% respectively. Each piece takes two weeks.

Contribution to Professional Component

CEG 333 contributes 2 hours to the Criterion 4(b), and also contains engineering design.

Course Contribution to Program Educational Objectives

CEG 333 contributes to Objectives 1 and 2. Through exposure to the use of an operating system other than Windows, it widens the skills. The design experience gained through the course project is realistic.

Course Contribution to Program Outcomes and Assessment

a	b	c	d	e	f	g	h	i	j	k
0	PX	PX	0	PX	0	P	PX	PX	PX	PX

Estimate CAC Category Content

	Core	Advanced			Core	Advanced
Data Structures				Concepts of PL	1	
Algorithms				Comp Organization + Architecture		
Software Design	1			Other		

Prerequisites by Topic

1. Programming (C++) Concepts: Variables, arrays, loops, if statements, recursion
2. Program development tools: editors, C++ compilers, linkers, debuggers
3. Windows OS power user
4. Can read email using Unix
5. Can surf the web using Unix
6. Remote computers: telnet, ftp, rlogin, ssh, sftp

Oral and Written Communications

There are no oral presentations. Students submit source code of their projects along with a "ReadMe", a text file that highlights the design details as well as problems and defects in their program. We do not claim that the ReadMe.txt constitutes written communications in the sense intended by this section.

Social and Ethical Issues

The development of user interfaces, and permissions to various computer resources is presented in the societal context..

Theoretical Content

None.

Problem Analysis

The projects are about using an OS and a shell in performing various typical tasks. The requirements of the problem are analyzed by the student before implementing them.

Solution Design

Skeletal solutions of parts of the project are given by the instructor at the conceptual level in the lectures, and also in source code files. The student needs to design further details and implement them.

Learning Objectives and Desired Outcomes

Outcomes

The student should be able to apply the concepts above to the following:

1. Comfortable in using Unix/Linux at the command-line level
2. Can manage files on Unix: `ls, cd, mkdir, rm, ln, chmod ...`
3. Can use Unix program development tools: `emacs, make, man, g++, gdb`