

Course Number	CS 306	Course Title	Linux/UNIX Programming			
Semester Hours	3	Course Coordinator	Koushik Sinha			
Catalog Description	<p>This course will prepare students to develop software in and for Linux/UNIX environments. Topics to be covered include basic operating system concepts, effective command line usage, shell programming, the C language, programming development tools, system programming, network programming (client-server model and sockets), and GUI programming.</p>					
Textbooks						
FA20						
<p>Matthew, N. & Stones, R. (2007). <i>Beginning Linux Programming</i>. Wrox, 4th Edition, ISBN 9780470147627.</p> <p>Jones, M.T. (2008). <i>GNU/Linux Application Programming (with CD)</i>. Charles River Media, 2nd Edition, ISBN: 9781584505686.</p>						
References						
<ul style="list-style-type: none"> Barrett, D. (2004). <i>Linux Pocket Guide</i>. O'Reilly. 						
Course Learning Outcomes						
<ul style="list-style-type: none"> Understanding the basic set of commands and utilities in Linux/UNIX systems. To learn to develop software for Linux/UNIX systems. To learn the C language and get experience programming in C. To learn the important Linux/UNIX library functions and system calls. To understand the inner workings of UNIX-like operating systems. To obtain a foundation for an advanced course in operating systems. 						
Assessment of the Contribution to Student Outcomes						
SP20						
Outcome →	1	2	3	4	5	6
Assessed →	X	X				X
Prerequisites by Topic						
CS 220 and 221 with a grade of C or better.						

Major Topics Covered in the Course

1. Introduction to operating systems: OS functions, OS types, components (kernel, drivers, etc.); command-line interfaces (CLIs) vs. GUI interface (GUIs) {1 class}
2. Introduction to UNIX: Linux/UNIX history; CLI: shells, bash, C shell; distributions and application software; file system structure, pathnames, file permissions {3 classes}
3. Effective shell (CLI) usage: basic commands and utilities (cd, mkdir, rm, cp, cat, etc.); piping, redirection, filters (grep, sed etc.); command line editing, history, etc.; shell startup files, aliases {3 classes}
4. Bash shell programming: variables, parameters; metacharacters, shell expansions; control constructs (if, for, while, etc.); I/O {3 classes}
5. GNU programming and development tools: gcc, make, gdb, etc. editors, IDEs, libraries {1 class}
6. Introduction to C language programming: header files, preprocessor directives; pointers, arrays, strings; dynamic memory allocation; command-line arguments, return values; standard library I/O {3 classes}
7. System programming: files and I/O (open, close, read, write, dup, etc.); directories (opendir, readdir, etc.); processes (fork, exec, etc.); signals; pipes and IPC (pipo, kfifo, etc.) {14 classes}
8. Network programming: IP basics; TCP, UDP client-server model; sockets system calls {6 classes}
9. GUI programming: curses/ ncurses , Qt and GTK toolkits; event-oriented programming {6 classes}