# OS Project Ideas (COSC519)		Fall 2010

Operating system projects are intended to provide hands on experience by building an OS component such as memory management, process management, and storage management. A group of students work on a project and each student must contribute to the success of the project.

The project ideas can be categorized into 5 areas as described below.

(1) Simulation Type

Choose an OS component and simulate the function by modeling its characteristics. You can choose either C++ or Java programming language to simulate the function. For example, if you choose a process management; you may consider a model that simulates "n" number of processes running, where each process can be randomly selected to load, run, suspend, and abort. You may choose your favorite scheduling mechanism to select a process to run. You may also assume a single ready queue to process the arriving processes. The model should run indefinitely and print some statistical information such as average number of processes, average execution time, no of processes waiting in the queue, and so on. Visualization of the model is not required, but some primitive user interface should be provided to control the input attributes to the system. When a system is simulated, you need to collect data, study the behavior, and derive some knowledge from the operation of the system.

(2) Linux OS Based

There can be a variety of projects that are based on Linux operating system. Some examples of these projects include: kernel compilation, module management, timing analysis of the kernel by inserting probes, modification of kernel components, adding new modules (small), and so on. Each one of these project requires a thorough understanding and compiling of the kernel modules. One can also use kernel tools such as S-trace to identify and analyze system calls.

Some other project ideas may include writing device interfaces using API instead of system calls. For example, writing a raw interface to USB using API.

(3) Device Drivers

USB being a popular device in today's computing, one can consider building a device driver from scratch for a NIC, Mouse, Keyboard, or any other type of device. One can also dissect an existing device driver in Linux and understand and analyze the code in reference to OS and its interfaces.

(4) Industrial Applications

Some students with industry background may bring their OS based applications knowledge to this project and to the classroom to construct and analyze a particular OS based application. Such projects may include system administration, system configuration, shell scripts, backup, recovery, and checkpoints.

(5) Bare Machine Computing Applications

The faculty has spent over ten years in the development of bare machine computing paradigm and its applications. The bare machine computing laboratory provides numerous applications where many doctoral students are involved in the development of these projects. You may choose a component that can be used in the bare machine computing lab and make your contributions. Some examples of such projects include: graphics drivers, USB drivers, 32/64 bit CPU interfaces, Dual-core interfaces, implementing network protocols, math libraries, study security vulnerabilities on bare machines, bare file mangers, and so on.

When you choose these projects, you may be working with one of the doctoral student in the bare machine computing lab and gain additional knowledge in their current research.

NOTE: No paper study projects are encouraged.