

# ITCS 6144/8144 Course Project Plan Report

Project Title: Understand Linux OS via Modern Tool (Systemtap)

Prepared by: Abdullah Al Raqibul Islam (UNCC ID # 801151189)

---

The main purpose of this project will be to understand and observe the internal behaviors and performance characteristics of Linux operating system. Additionally, I will try to explore the relations between some of these quantities. For doing this, I will use a dynamic tracing tool, named Systemtap.

SystemTap is both scripting language and tool which can be used for profiling program on Linux kernel-based operating systems in runtime. SystemTap scripts are generally focuses on tracking events. Systemtap scripts may be executed with one of three following backends:

1. Loadable kernel module (has the fullest capability to inspect and manipulate any part of the system, requires most privilege)
2. DynInst (using to instrument the user's own user-space programs only, requires least privilege)
3. BPF (limited to the Linux kernel interpreter's capabilities, requires an intermediate level of privilege)

**Background Study:** Initially I need to perform the following background study

1. Systemtap and what it can do
2. How to profile/debug/optimize complex system in runtime

**Platform for the project:** I will perform the necessary experiments for this project on operating system centos7. I already have installed centos7 on a virtual machine (VirtualBox).

**Project structure:** This project will have two parts,

1. Writing several SystemTap scripts and some analytic programs (using C/C++) to gain deep understanding about the various parts of the Linux Operating System.
2. Will develop a visualization tool (D3J) to present the experiments done.

**Project Plan:** To gain deep understanding about the various parts of the Linux Operating System, I will try to conduct the following experiments using SystemTap:

1. By the meadterm, I will try to cover the following things
  - Understand OS Kernel: Addresses and contents of user frame and kernel frame of a user process.

- Understand Processes: Parent-Child tree of an process as well as detailed process specific information i.e. executable files, command line arguments, environment variables, uid, gid, cpuid, etc. Also observe, how long it takes before your user process actually starts to execute its main.
  - Understand System Calls: With time duration, observe all the system calls issued by a process or in the whole system in a given period of time.
2. By the end of the semester, I will try to cover multiple experiments from the following list
- Understand Context Switch: Observing how often and how long task scheduling and context switch occur.
  - Understand Scheduler: Observing the latency between a thread being woken up and it actually being dispatched to a CPU. Tracing the amount of time that each process spends in running, sleeping, queuing, and waiting for io. Show how often and how long process scheduling takes.
  - Understand Virtual Memory System: Tracing page faults, user space frees, page ins, copy on writes and unmaps. Tracing and analyzing memory usage of a user process, identify the memory leak.
  - Understand Virtual File System: Recording file open/close/read/write/mmap operations done by process. Will store these ops in a per-process way (together with the info of processes). Tracing slow file system synchronous reads and writes.