Understand Linux OS via Dynamic Tracing Tool (Systemtap)

Abdullah Al Raqibul Islam  
*Department of Computer Science, College of Computing and Informatics*  
*University of North Carolina  
Charlotte*, USA  
[aislam6@uncc.edu](mailto:aislam6@uncc.edu)

*Abstract*—Operating system (OS) is a piece of software that helps users to operate the hardware. In understanding an OS, it is very important to know it’s internal logic and performance characteristics. In modern operating systems, kernel acts like what a heart does for human body. The purpose of this project is to understand and observe the internal behaviors and performance characteristics of Linux operating system. In exploring the relations among some of these quantities, I have used a dynamic tracing tool namely Systemtap.

Keywords—operating system, memory management, dynamic tracing, systemtap

# Introduction

Introduction about operating system, kernel, components of operating systems ...

To find software bugs due to incorrect system setup, system administrator perform instrumentation in finding those bugs, which includes, performance statistics collection and their analysis, debug or system audit. One of the common approaches to instrumentation is “sampling” by installing probes at specified places of software to collect state of the system: values of some variables, stacks of threads, etc. Sampling is very helpful when you do not know where issue happens.

For example, some function, say foo() that processes lists of elements, consumes 80% of the time, but doesn't say why. It could be because the list is too long or list is an inappropriate data structure for foo(). With tracing we can install a probe to that function, gather information on lists (i.e. their length) and collect cumulative execution of function foo(), and then cross-reference them, searching for a pattern in lists whose processing costs too much CPU time.

To gain deep understanding about the various parts of the Linux Operating System, I have conducted the following experiments using SystemTap:

1. Understand OS kernel: Addresses and contents of user frame and kernel frame of a user process.
2. Understand system-calls:
   * Within specific time duration, observe all the system-calls issued by a process, and, in the whole system.
   * Within specific time duration, trace time spent in all system calls in a per process way
3. Understand Processes
   * Trace parent-child tree of a process
   * Detailed current process information (including executable files, command line arguments, environment variables, uid, gid, cpuid, etc.)
   * Trace the time duration a process takes before the process actually starts to execute.
4. Understand Virtual File System
   * Trace major and minor page faults
5. Understand Virtual Memory System
   * Trace file open/close/read/write/mmap operations done in the system.
   * Trace the above mentioned operations in a per-process way.

# 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. It is designed to balance between several key requirements including easy of use, performance, tranparency, flexibility and sefty to use in production system etc. The structures and components of systemtap is presented in Figure 1.

One of the systemtap’s keyword is prob point which indicate the a particular point in kernel/userspace code, or a specific event. The systemtap input consists of a script which makes the assiciation of hendeler routines with the prob points.

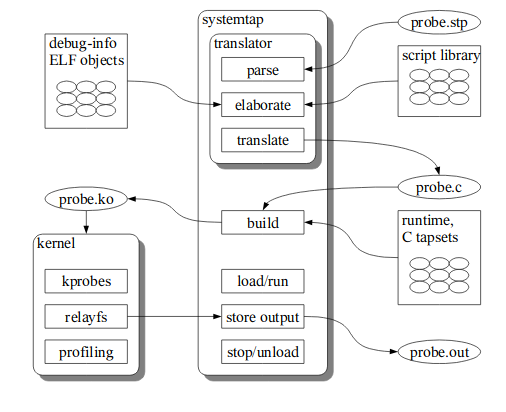


Figure 1: Systemtap processing steps

# Understand Systemcall

Before you begin to format your paper,

# Understand Process

After the text edi

# Understand Virtual File System

After the text edi

# Understand Virtual Memory System

After the text edi

##### Acknowledgment

This project is done as part of the completion of course ITCS 6144/8144: Operating System Design. This course has been taken in Fall 2019 semester, as part of my PhD degree at UNC Charlotte.

##### References

The template will number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2]. Refer simply to the reference number, as in [3]—do not use “Ref. [3]” or “reference [3]” except at the beginning of a sentence: “Reference [3] was the first ...”

Number footnotes separately in superscripts. Place the actual footnote at the bottom of the column in which it was cited. Do not put footnotes in the abstract or reference list. Use letters for table footnotes.

Unless there are six authors or more give all authors’ names; do not use “et al.”. Papers that have not been published, even if they have been submitted for publication, should be cited as “unpublished” [4]. Papers that have been accepted for publication should be cited as “in press” [5]. Capitalize only the first word in a paper title, except for proper nouns and element symbols.

For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [6].

1. G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. *(references)*
2. J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.
3. I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.
4. K. Elissa, “Title of paper if known,” unpublished.
5. R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.
6. Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].
7. M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.

**IEEE conference templates contain guidance text for composing and formatting conference papers. Please ensure that all template text is removed from your conference paper prior to submission to the conference. Failure to remove template text from your paper may result in your paper not being published.**