

Machine Exercise 4: System Calls Probing

Salmon, Paulino III I.

2015-11557

paulino.salmon@eee.upd.edu.ph

I. EXPLAIN THE FUNCTIONALITY OF STRACE. WHAT ARE THE ALTERNATIVES TO IT AND HOW DO THEY DIFFER?

The 'strace' command traces system calls and signals made throughout the process of running an executable file. 'Strace' is based on the Linux exported facility 'ptrace' which allows it to interrupt the trace process every time a system call is invoked. A few alternatives to 'strace' are going to be discussed below.

A. DTrace

DTrace is one alternative to STrace. DTrace is based on the D programming language, and injects its created bytecodes to the system kernel, in which, these bytecodes get executed whenever a system call is invoked. Compared to STrace, it is much more efficient and flexible and it does not add in much overhead when tracing the entire process, but it requires deeper technical knowledge.

B. Sysdig

Sysdig is another alternative to STrace. Sysdig probes the captured events in the kernel and installs a handler that gets called when specific functions in the kernel get invoked. Sysdig's overhead is very ideal for running in production environments, which is an advantage of its architecture. Compared to STrace, it dominates in terms of versatility and to how easy it is to use.

C. GDB

The GNU Debugger is a portable debugger that can run on most Unix-like systems. It has features such as extensive facilities for tracing and altering execution of processes. It also features a remote debugging functionality.

D. Valgrind

Valgrind's advantage is that it helps you debug problems related to memory accesses. It ports out multiple functionalities such as a memory error detector, two thread error detectors, a cache and branch-prediction profiler, a call-graph generating cache and branch-prediction profiler, and a heap profiler.

II. WHEN RUNNING STRACE ON YOUR PROGRAMS, WHAT IS THE FIRST SYSTEM CALL THAT WAS USED? WHAT IS THE FUNCTION OF THIS SYSTEM CALL?

The first system call that strace invokes is the **execve()** command. This is where the program starts. The first code executes from the dynamic loader. Execve stands for execute program, and it runs the program pointed to by the filename, in which, this filename should either be a binary executable or a script.

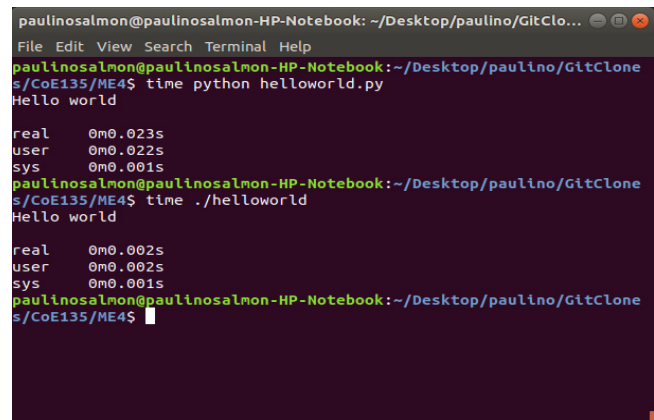
III. WHAT ARE THE SYSTEM CALLS THAT YOUR PROGRAMS USE TO ALLOCATE MEMORY? DOES IT ALLOCATE DIRECTLY INTO YOUR MEMORY HARDWARE? ELABORATE ON YOUR ANSWER.

The main system calls that are responsible for allocating memory during a process are the **mmap()/munmap()** commands. Mmap creates mappings in the virtual address space of the calling process. Munmap deletes the mappings for the specified address range, and causes further references to addresses within the range to generate invalid memory references.

Processes running on the user space cannot access the physical addresses directly and are confined in the virtual memory. If these such boundaries do happen and if a process tries to write to physical memory, a segmentation fault occurs since this memory region was not mapped for this specific process and these mappings are set by 'mmap()', and mmap creates virtual mapping for calling processes.

IV. EXPLAIN THE DIFFERENCE IN COMPLEXITY IN TERMS OF SYSTEM CALLS BETWEEN A SIMPLE C PROGRAM AND A SIMPLE PYTHON PROGRAM. IS THERE A SIGNIFICANT TIME DIFFERENCE IN RUNNING A SIMPLE HELLO WORLD PROGRAM BETWEEN THE TWO LANGUAGES? ELABORATE ON YOUR ANSWER.

C is always drastically faster in general as compared to Python as Python is interpreted as compared to C which is direct machine code. Python undergoes through a lot of machine instructions and translations as compared to C, which slows down its entire runtime.



```
paulinosalmon@paulinosalmon-HP-Notebook: ~/Desktop/paulino/GitClo...
File Edit View Search Terminal Help
paulinosalmon@paulinosalmon-HP-Notebook:~/Desktop/paulino/GitClone
s/CoE135/ME4$ time python helloworld.py
Hello world
real    0m0.023s
user    0m0.022s
sys     0m0.001s
paulinosalmon@paulinosalmon-HP-Notebook:~/Desktop/paulino/GitClone
s/CoE135/ME4$ time ./helloworld
Hello world
real    0m0.002s
user    0m0.002s
sys     0m0.001s
paulinosalmon@paulinosalmon-HP-Notebook:~/Desktop/paulino/GitClone
s/CoE135/ME4$
```

From the above screenshot, the C hello world program is faster by about **20ms**, which is already a very significant time difference when talking in the context of execution time.

V. DO YOU NEED ROOT PERMISSION WHEN RUNNING STRACE ON ANY PROGRAM? ELABORATE ON YOUR ANSWER.

Root privileges are required in order to plant strace on **any running program**, especially the ones that you do not even own. A normal user can only attach strace on his owned processes.

VI. FOR BONUS: SHOW SCREENSHOTS OF THE RELEVANT PARTS OF THE COMMUNICATION BETWEEN THE TWO PROGRAMS USING FIFOS AS INDICATED ABOVE.

A. Getting the PIDs of the Processes

[illegible]

B. Catching the SIGINT Ctrl+C Signal

[illegible]

C. Catching the SIGTSTP Ctrl+Z Signal

[illegible]