EI338 Computer System Engineering Homework 4

Zhou Litao 518030910407 F1803016

October 15, 2020, Fall Semester

Exercise 1 List five services provided by an operating system, and explain how each creates convenience for users. In which cases would it be impossible for user-level programs to provide these services? Explain your answer.

Solution.

File Manipulation System services, such as creating, deleting, copying, and renaming files or directories, provide users with an interface to the system calls that manipulates files. OS users don't actually have to know the details of system call APIs.

Status Information The system will maintain status information like date, time, and number of users, or even more complex information, such as logging, and debugging information. Operating systems usually provide users with a GUI window or command line interface to display these status, which will let users access such status in a simple action.

Programming language support To run programs written in popular programming languages, services like compilers, assemblers, debuggers or interpreters usually come with the operating system. They don't belong to kernal, but are necessary to make the system useful for users.

Communication facilities These programs provide the mechanism for creating virtual connections among processes, users, and computer systems. With these services, users can connect to a greater network (such as other devices, or worldwide web network).

Background services They launch at boot time and do jobs like disk checking, process scheduling, error logging and printing. They may be invisible from users, but are necessary to make the operating system function well.

For some application utilities, such as web browsing services, and emails services, they can be substituted by other user-level programs. However, when it comes to the security and stability of the whole operating systems, such services must be provided by the operating system itself, so that they can work along with the kernal well.

□

Exercise 2 Describe three general methods for passing parameters to the operating system.

Solution.

1. Parameters can be passed through registers
2. For parameters that outnumber the registers, they are stored in a block or table in the memory, and the address of the block will be passed through registers
3. Parameters also can be placed, or pushed, onto a stack by the program and popped off the stack by the operating system.

□

Exercise 3 List at least five types of system calls, and describe the main activities of each system call.

Solution.

Process control Create process, terminate process, load and execute programs, get process attributes, set process attributes, wait and signal event, allocate and free memory.

File management Create, delete, open and close files. Get and set file attributes.

Device management Request device, release device. Read, write from devices. Get and set device attributes, and logically attach or detach devices

Information maintenance Get time or date, set time or date. Get system data, set system data. Get process, file, or device attributes. Set process, file, or device attributes

Communications Create, delete communication connection. Send, receive messages. Transfer status information. Attach or detach remote devices.

□

Exercise 4 Perf is a performance analyzing tool in Linux. It is capable of statistical profiling of the entire system. Please use the perf tool to analyze the performance of the program in the attachment. Requirements are as follows.

1. The experimental analysis results of the program should be in the form of table or picture.
2. An detailed explanation of your result is required.

Solution. By running the compiled program using `sudo perf record -g ./test` command, we obtain a file called `perf.data`. By running `perf report`, we can view the performance report in the call graph format, shown in Figure 4. The `Children` column indicates the percentage of time when the labeled function is in the call stack, while the `Self` column counts the percentage of the time spent executing the function in case. Here is the explanation.

- Note that there are 10^6 loops in the `longa()` function, it makes sense that about 99.64% of the runtime is spent on this function.
- By tracking the caller of `longa()`, we find that 90.97% of them are called by `foo1()`, which has 100 inner loops. And 8.67% of them are called by `foo1()`, which has 10 inner loops.

The analysis coincides with what we have expected.

□

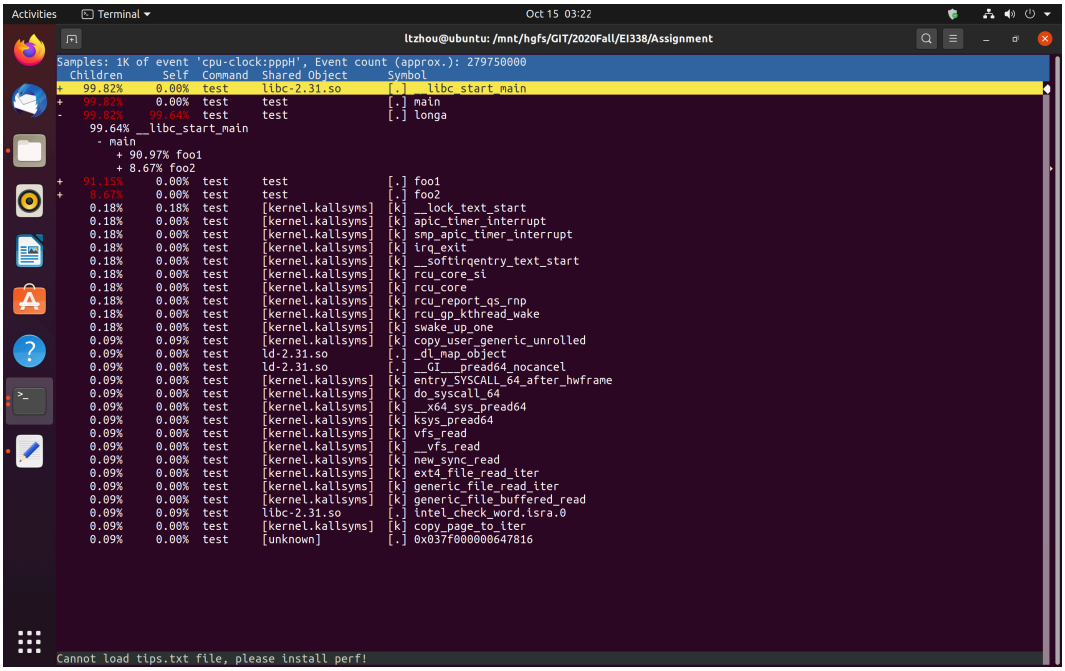

Figure 1: `perf` analysis result