

CS 377 : Lab assignment #6

system Calls in the Pranali

14 Feb 2014

Pranali, an instructional OS, is a part of an os workbench. The workbench is a software platform that facilitates assignments in design and implementation of an OS at various levels of complexity. It has three key components:

1. Multi-2-Sim: A simulator for the X86 architecture. The platform runs as a single process under the OS of the computer. (We refer to the OS of the computer as the *host computer OS*, and use Linux as the host computer OS.)
2. Rudimentary OS working on the Multi-2-Sim platform. It provides some functionalities and provides a way to invoke some functionalities of the host computer OS. we refer to this OS as the *Guest OS*. This guest OS is tentatively named Pranali.
3. The host computer OS.

details of this arrangement are described in the document at the *URL* : http://www.cse.iitb.ac.in/dmd/cs347/OS_workbench_document.pdf. A set of slides Pranali.ppt is accessible through the Moodle site of CS 377.

The code of the OS workbench can be found at the following *URL* : http://www.cse.iitb.ac.in/dmd/cs347/OS_workbench_code

Notes on operation of the Guest OS

The guest OS performs booting and then executes a shell, which accepts user commands. The “create process” command takes user id and process priority as parameters. The guest OS puts this information in the PCB. The guest OS multiplexes operation of user processes on an n-instructions-at-a-time basis. we call-n the *instruction-slice*.

The boot cod reads a “configuration” file and performs the following tasks: It creates a “virtual disk” by using the configuration parameters concerning number of disk blocks. It sets the instruction-slice in accordance with a configuration parameter.

The assignment

Study the guest OS and find the following details / answer the following questions / perform the following tasks:

1. Identify the code that perform booting actions.
2. Change the specification of the virtual disk in the “configuration” file to create a disk according to your own choice.
3. Study how system calls are implemented in the Guest OS.
 - (a) How does a system call handler in the Guest OS get invoked?
 - (b) How many system calls does the Guest OS have at present?
 - (c) How is a system call passes on to the Host OS?
 - (d) How does a system call handler access some part of the address space of a process? For example, if a parameter of the system call is the address of a memory area, how does the system call handler access contents of this memory area? Study the system call handler of the ”read” system call(system call #3), or the Hello system call to understand it.
4. Study the scheduling arrangement of the Guest OS.
 - (a) How to find the PCBs of all processes?
 - (b) What is the field that contains the instruction-slice of a process?
 - (c) How is scheduling performed in the guest OS?
5. Add new system calls
 - (a) Add a “set instruction-slice” system call for changing the instruction-slice for a process. The parameter would be a single integer, passed through a register.
 - i. Write the library function and a test program.

- ii. create two or more processes and see the effect of changing their instruction-slices on their speed or response time.
- (b) Add system calls for performing read and write operations on the virtual disk.
 - i. Each system call would take the following parameters:
 - A. The operation to be performed, whether a read or a write.
 - B. Number of bytes to be read/written.
 - C. Address of memory area in the process address space.
 - D. The disk block number.
 - E. Offset into the disk block.

The system call for reading would read the specified number of bytes from the given offset in the disk block and deposit them in the memory area. The system call for writing to the disk would perform the converse operation.
 - ii. The system calls should perform protection checks as follows:
 - A. A read or write operation should be rejected if the disk block contains data written by some other user and an error message should be printed.
 - B. To perform this check, you can maintain a table showing which disk block is in use by which user. Initially, the table should be blank. (*Hint:* Create this table at boot time when the virtual disk is created. It should have as many entries as the number of disk blocks.)

Submission deadline: Deadline of this assignment is **5 pm**.