CSE 30341: Home Work Project 1

Assigned: Fri, Jan 18

Due: Mon, Feb 04, 10:40AM

Late submissions will not be accepted

Group effort

Goal:

The primary goal of this project is to familiarize your-self with installing and configuring an operating system. We will install a new kernel, configure it and measure the impact of this configuration change on a small program. We will vary the kernel timer granularity for our experiments. The kernel timer granularity affects the smallest timer events supported by a particular operating system. It also affects the granularity with which processes are scheduled. More discussion on "Timers and Time management" is available in **Linux Kernel Development.** Linux kernel allows one to modify the timer to values such as 100 Hz, 250 Hz and 1000 Hz. A finer timer allows for more control at the expense of increased overhead.

For this project, we will analyze these effects for the following C program. On your machine, open many terminal windows and run this program, simultaneously, in each of the windows. As you increase the number of simultaneously running program, analyze the time it takes per loop.

```
#include <sys/time.h>
#include <stdio.h>
#include <unistd.h>

void main(int argc, char *argv[], char *envp[])
{
        struct timeval start, end;
        int count;
        int i, j;
       int array[1024][1024];

      while(1) {
        gettimeofday(&start, NULL); /* Beginning of interval */
        for(count=0; count<1000; count++)
            for (j=0; j<1024; j++)
                    for (i=0; i<1024; i++)
                          array[i][j] = 1;
       gettimeofday(&end, NULL); /* End of interval */

        end.tv_sec -= start.tv_sec; start.tv_sec = 0;
      printf("%d\n", (end.tv_sec*1000000 + end.tv_usec) -
          (start.tv_sec*1000000+start.tv_usec));
   }
}
```

Plan:

Note that you are not required to follow these steps. If you know of other ways to achieve the same goals, go for it.

1. **Download the latest kernel**: For your experiments, download the latest Linux kernel (such as 2.6.23.14) from a source code repository such as ftp.kernel.org. Note that the machines in the lab are not backed up – do not store important files in those machines. After downloading, uncompress the kernel files.
2. **Configure the kernel**: An easy way to configure the kernel is to use a command such as 'make menuconfig'. This command allows you to configure the kernel using a simple menu system. Explore the various options and find our how you could configure the timer granularity. For the experiments, change the timer to 10 Hz and 1000 Hz (for two set of results).
3. **Compile and install the new kernel**: Once you have configured a particular kernel, save the configuration and then compile the kernel (using 'make; make modules; make modules_install; make install')
4. **Experiment with the new kernel**: Reboot to the new kernel (you may have to choose your kernel during the boot up process). Once you have logged into the correct kernel, run the above program and report your results.

Report:

Write a succinct report which reports the specific set of parameters that you configured for your kernel for the experiments. Also, describe the C program described above, what the expected output was and the observed behavior for the two timer values that you configured for the kernel.

Resources:

You can use the 6 desktops, expsys-dtp1 through expsys-dtp6.cse.nd.edu, available at Cushing 208 for your experiments. You also have access to two Itanium 2 servers (donated by HP), expsys-svr2.cse.nd.edu and expsys-svr4.cse.nd.edu for this course. For this project, we will not use these servers (these servers use EFI for booting and hence may cause some problem during booting). **There are other machines in the lab, do not install Linux on the other machines – other classes use it for their course projects.** Coordinate with your peers so that each one gets some time slot to try the experiments. Always install new kernels from the console (and not while remotely logged in). In case something goes wrong, it is helpful to be on the console. **Please notify me or the TA asap if a machine becomes unusable during your experiments. This will allow us to restore the machines quickly.** You are also free to use your own machines at home/lab etc.