WINTER 14~15

OPERATING SYSTEMS LAB - ITE209 SESSION 2

ARIVUSELVAN.K

Asst .Prof (Senior)

SITE

VIT UNIVERSITY - VELLORE

Observing Operating System Behaviour

Objective:

1. To report the behaviour of the OS to get the CPU type and model, kernel version.

2. To get the amount of memory configured into the computer, amount of memory currently available.

We can observe behavior of operating system (kernel) using / **proc** (process information pseudo-file system) utility.

Linux, Solaris, and other versions of UNIX provide a very useful mechanism for inspecting the kernel state, called the /proc file system. In this lab you will learn about this key mechanism by doing some exercise.

What is a /proc File System?

The **/proc** file system isn't a file system in the standard sense. Rather, the proc file system is a pseudo-file system which is used as an **interface to kernel data structures**. It is commonly mounted at /proc. Most of it is read-only, but some files allow kernel variables to be changed. The **/proc** file system is an OS mechanism whose interface appears as a directory in the conventional UNIX file system (in the root directory). You can change to **/proc** just as you change to any other directory. For example,

$ **cd /proc**

makes **/proc** the current directory. Once you have made **/proc** the current directory, you can list its contents by using the **ls** command.

A sample listing from /proc:

$ ls -l /proc

total 69

```
dr-xr-xr-x   3 root    root        0 Sep 10 01:35 1
dr-xr-xr-x   3 root    root        0 Sep 10 01:35 10314
dr-xr-xr-x   3 root    root        0 Sep 10 01:35 10315
dr-xr-xr-x   3 root    root        0 Sep 10 01:35 10317
dr-xr-xr-x   3 root    root        0 Sep 10 01:35 10318
dr-xr-xr-x   3 root    root        0 Sep 10 01:35 10320
dr-xr-xr-x   3 root    gradics     0 Sep 10 01:35 10337
dr-xr-xr-x   3 root    gradics     0 Sep 10 01:35 10340
::::              ::          ::              : :  : : : :
```

Contents of /proc File System : The proc file system can be used to obtain information about the system and to change certain kernel parameters at runtime.

1.   What's the CPU type and model? **(cat /proc/cpuinfo)**

2.   What version of the Linux kernel are you using? **(uname -a)**

3.   How long has it been since your PC last booted? **(uptime)**

4.   How much of the total CPU time has been spent executing in user mode? idle? **(top)**

5.   How much memory is configured in your PC? **(cat/proc/ meminfo)**

6.   How much memory is currently available? **(cat/proc/meminfo)**

7.   How many disk read/write requests have been made? **(cat /proc/diskstats)**

8.   How many context switches has the kernel performed? **(cat /proc/stat)**

9.   How many context switches has a process had? **(/proc/ [pid] /status)**

   Note: Get Process id (**pid**) using PS command

10.   How many processes have been created since the system was booted? **(cat /proc/ stat)**

11.   How many processes are blocked waiting for I/O to complete? **(cat /proc/ stat)**