# Operating Systems and Useful Linux Commands

Guy Lebanon

August 29, 2006

As mentioned earlier, any assembly (or compiled or interpreted) program can be loaded into memory and executed by the processor. There are many problems, however, with this scheme: only one program can run at a certain time and only by one user. Connection with input/output devices and files are elementary and many services that are useful for programs such as the system clock need to be re-programmed repeatedly. The operating system is a partial solution to these and other difficulties. It forms an convenient interface layer between the computer hardware (processor, memory, hard drive) and the user or programmer.

The main roles of the operating system are outlined below. In parenthesis are the corresponding Linux commands. To get more information on these commands use the `man` command or look them up in internet (for example `http://www.linuxdevcenter.com/linux/cmd/`) or a book.

1. Allow multiple processes to run at the same time. This is done using context switch which is a mechanism that intermingles the execution of different processes (the processor actually runs only one assembly code line at any given moment). Once execution of a certain process is frozen it is saved in its current state and the execution of different process resumes. When the execution of the first process resumes, it does so from starting from the place it previously stopped. The task of choosing which process to run at which time and for how long is the task of the scheduler. Some processes are given higher priority than others and the processor spends more time on these processes. Useful Linux commands: `ps, kill -9, Ctrl-z, Ctrl-c, bg, fg, &, nice`

2. Manage the file system by enabling different processes to read and write data from and to the disk in a logically coherent manner. The OS also makes sure that two processes trying to access the same file at the same time do not mess it up. The files may be either text files (can be read and written by a text editor), or binary (represents assembly code or data in a non-text format). The files are organized in a file directory which resembles a tree. In Linux the symbol for separating branches in the tree is `/`. The current directory is `.` and the parent directory is `..`. Useful Linux commands: `ls -a, ls -F, cd, mkdir, pwd, rmdir, cp, rm, mv, grep, cat, more, df, gz, tar`

3. Allow multiple users to have separate accounts and use the computer without interfering with each other. Users can run programs independently of each other and can access only files in the file directory area for which they have been given permission. The permissions (viewable with `ls -l` command) are marked as `w` for write, `r` for read, `x` for execute in three different groups owner, group, and all. Useful Linux commands are `login, logout, passwd, ls -l, chmod 755, chmod 777, chmod 700, nohup`

4. The OS facilitates the access to input/output devices such as the screen, printers, external disks, etc. The user can access these I/O devices through simple commands without the need to write a program in a programming language or assembler. Useful Linux commands: `echo, cat, lpr, lpq, lprm, >, <, >>, |`.

5. Provides support for programmers. In contrast to some commercial software such as Microsoft Visual Studio that is an integrated development environment or IDE (a single program that supports the programmers in many ways), Linux's programming support is composed of several special-purpose tools that work individually to provide support in different areas. Popular C/C++ compiler is `gcc`, debugger is `gdb`, and profiler is `gprof`.

6. Miscellaneous tasks: command documentation (`man`), word processing (`pico, emacs, vi`), graphic console (`xterm`), email (`mail, pine`), text web-browser (`lynx`), and other useful commands `alias, noalias, backslash, ftp`.