System Programming
01-Introduction to Unix-Linux systems

Sayed Ahmad Sahim

Kandahar University
Computer Science Faculty

February 18, 2019

Overview

1. Introduction to Unix
2. Introduction to Linux
3. Bash Shell

Introduction to Unix

- Unix (Thompson and Ritchie 1974, 1978) is a general purpose operating system.
- It was developed by Ken Thompson and Dennis Ritchie on the PDP-11 minicomputers at Bell Labs in the early 70s.
- Initial recipients of this Unix system were mostly universities and non-profit institutions.
- It was known as the V6 Unix.
- This early version of Unix started the Unix revolution on Operating Systems, with long-lasting effects even to this day.

AT&T Unix

- Development of Unix at AT&T continued throughout the 1980s, cumulating in the release of the AT&T System V Unix (Unix System V 2017), which has been the representative Unix of AT&T.
- System V Unix was a uniprocessor (single CPU) system. It was extended to multiprocessor versions in the late 80s (Bach 1986).

Berkeley Unix

- Berkeley Unix developed by the Berkeley Software Distribution (BSD) at the University of California, Berkeley, from 1977 to 1985.
- The most significant contributions of BSD Unix are the implementation of:
  - TCP/IP suite
  - Socket interface
- The two components are incorporated into almost all other operating systems as a standard means of networking.
- It helped tremendous growth of the Internet in the 90s.
- BSD Unix advocates open source from the beginning.
- Later releases of BSD Unix
  - FreeBSD
  - OpenBSD
  - NetBSD

HP Unix

- HP-UX (HP-UX 2017) is Hewlett Packards proprietary implementation of the Unix operating system, first released in 1984.
- Recent versions of HP-UX support the HP 9000 series computer systems, based on the PA-RISC processor architecture.
- The unique features of HP-UX include
  - Built-in logical volume manager for large file systems
  - Access control lists as an alternative to the standard rwx file permissions of Unix.

Sun Unix

- Solaris (Solaris 2017) is a Unix operating system originally developed by Sun Microsystems (Sun OS 2017).
- Since January 2010, it was renamed Oracle Solaris.
- Unix systems are proprietary and tied to specific hardware platforms.
- This present a challenge to readers who wishes to practice systems programming in the Unix environment.
- For this reason, we shall use Linux as the platform for programming exercises and practice.

Introduction to Linux

- Linux (Linux 2017) is a Unix-like system.
- It started as an experimental kernel for the Intel x86 based PCs by Linus Torvalds in 1991.
- Ambig milestone of Linux occurred in the late 90s when it combined with GNU (Stallman 2017) by incorporating utilities such as GCC, Emac text editor and bash.
- In very short period linux implemented TCP/IP suit to access network and later GUI (x-windows)

Linux vs Unix

- Linux includes many features of other Unix systems.
- In some sense, it represents a union of the most popular Unix systems.
- To a large extent, Linux is POSIX compliant.

Linux Versions

- The development of Linux kernel is under the strict supervision of the Linux kernel development group.
- All Linux kernels are the same except for different release versions.

Debian Linux

- Debian is a Linux distribution that emphasizes on free software.
- It supports many hardware platforms.
- Debian distributions use the .deb package format and the dpkg package manager and its front ends.

Ubuntu Linux

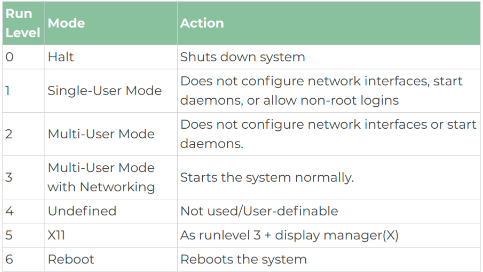
- Ubuntu is a Linux distribution based on Debian.
- buntu is designed to have regular releases, a consistent user experience and commercial support on both desktops and servers.
- Ubuntu Linux has several official distributions.

Linux Hardware Platforms

- Linux was originally designed for the Intel x86 based PCs.
- Earlier versions of Linux run on Intel x86 based PCs in 32-bit protected mode.
- It is now available in both 32-bit and 64-bit modes.
- We can run Linux on many virtual environment such as:
  - Virtual Box
  - VMWare player
  - Docker
  - Kubernetes

Linux Run-levels

- The Linux kernel starts up in the single user mode.
- It mimics the run-levels of System V Unix to run in multi-user mode.
- Conventionally, seven runlevels exist, numbered from zero to six; though up to ten can be used.



Run Level	Mode	Action
0	Halt	Shuts down system
1	Single-User Mode	Does not configure network interfaces, start daemons, or allow non-root logins
2	Multi-User Mode	Does not configure network interfaces or start daemons.
3	Multi-User Mode with Networking	Starts the system normally.
4	Undefined	Not used/User-definable
5	X11	As runlevel 3 + display manager(X)
6	Reboot	Reboots the system

Bahs Shell

- Bash is the shell, or command language interpreter, for the Linux operating system.
- The name is an acronym for the Bourne-Again SHell, a pun on Stephen Bourne.
- Developed by GNU project.
- The default Linux shell.
- Backward-compatible with the original sh UNIX shell.
- Bash is largely compatible with sh and incorporates useful features from the Korn shell ksh and the C shell csh.
- Bash is the default shell for Linux. However, it does run on every version of Unix and a few other operating systems such as ms-dos, os/2, and Windows platforms.

The improvements offered by BASH include:

- The Bash syntax is an improved version of the Bourne shell syntax. In most cases Bourne shell scripts can be executed by Bash without any problems.
  - Command line editing.
  - Command line completion.
  - Unlimited size command history.
  - Prompt control.
  - Indexed arrays of unlimited size (Arrays).
  - Integer arithmetic in any base from two to sixty-four.
  - Bash startup files You can run bash as an interactive login shell, or interactive non-login shell.
  - Bash conditional expressions: Used in composing various expressions for the test builtin.
  - The Directory Stack - History of visited directories.
  - The Restricted Shell: A more controlled mode of shell execution.
  - Bash POSIX Mode: Making Bash behave more closely to what the POSIX standard specifies.

Shell commands

- The bash shell comes with two types of commands:
  1. Internal commands (builtins) - part of the shell itself, i.e. built into the shell.
  2. External commands - separate binaries stored in /sbin, /usr/sbin, /usr/bin, /bin, or /usr/local/bin directories.

Commands types

- The bash shell understands the following types of commands:
    - Aliases such as ll
    - Keywords such as if
    - Functions (user defined functions such as genpasswd)
    - Built in such as pwd
    - Files such as /bin/date

Shell commands

- Commands we should already know:
  - ls, ll
  - pwd
  - cd
  - touch, copy, rename
  - environment varaibles
  - man