LECTURE 1: ADVANCED UNIX/LINUX SHELL PROGRAMMING

1.0 INTRODUCTION AND DEFINITIONS

1.0.1 WHAT IS UNIX ?

The Unix operating system is a set of programs that act as a link between the computer and the user.

The computer programs that allocate the system resources and coordinate all the details of the computer's internals is called the **operating system** or the **kernel**.

Users communicate with the kernel through a program known as the **shell**. The shell is a command line interpreter; it translates commands entered by the user and converts them into a language that is understood by the kernel.

- Unix was originally developed in 1969 by a group of AT&T employees Ken Thompson, Dennis Ritchie, Douglas McIlroy, and Joe Ossanna at Bell Labs.
- There are various Unix variants available in the market. Solaris Unix, AIX, HP Unix and BSD are a few examples. Linux is also a flavor of Unix which is freely available.
- Several people can use a Unix computer at the same time; hence Unix is called a multiuser system.
- A user can also run multiple programs at the same time; hence Unix is a multitasking environment.

1.0.2 WHAT IS MEANT BY UNIX AND LINUX?

Linux is an open source, free to use operating system widely used for computer hardware and software, game development, tablet PCS, mainframes etc. **Unix** is an operating system commonly used in internet servers, workstations and PCs by Solaris, Intel, HP etc.

1.0.3 WHAT IS THE DIFFERENCE BETWEEN UNIX AND LINUX?

The biggest **difference between** the two will be that **UNIX** is a full fledged operating system, whereas **Linux** is just a kernel of many free operating systems like Arch **Linux**, Fedora, GNU Sense, **Linux** Mint, Red Hat, Ubuntu, etc.

1.1 UNIX ARCHITECTURE

The basic block diagram of a Unix system is shown in the diagram in the figure below.

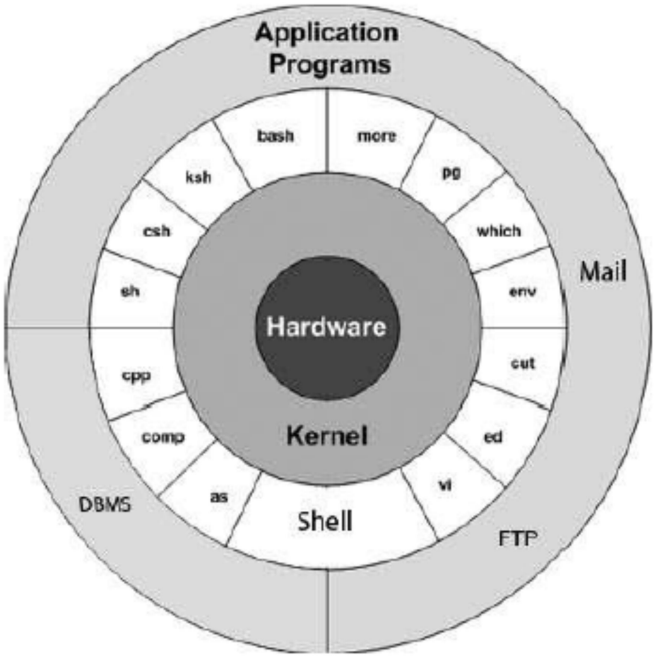


Figure A: Unix Architecture

The main concept that unites all the versions of Unix is the following four basics −

- **Kernel** − The kernel is the heart of the operating system. It interacts with the hardware and most of the tasks like memory management, task scheduling and file management.
- **Shell** − The shell is the utility that processes your requests. When you type in a command at your terminal, the shell interprets the command and calls the program that you want. The shell uses standard syntax for all commands. C Shell, Bourne Shell and Korn Shell are the most famous shells which are available with most of the Unix variants.
- **Commands and Utilities** − There are various commands and utilities which you can make use of in your day to day activities. **cp**, **mv**, **cat** and **grep**, etc. are few examples of commands and utilities. There are over 250 standard commands plus numerous others provided through 3rd party software. All the commands come along with various options.

- **Files and Directories** − All the data of Unix are organized into files. All files are then organized into directories. These directories are further organized into a tree-like structure called the **filesystem**.

1.2 TYPES OF SHELL

The some of the existing types of shell in UNIX, paths and default prompts are highlighted below.

Shell	Path	Default Prompt (non-root user)
The Bourne Shell (sh)	/bin/sh and /sbin/sh	$
The C Shell (csh)	/bin/csh	%
The Korn Shell (ksh)	/bin/ksh	$
The GNU Bourne-Again Shell (Bash)	/bin/bash	bash-x.xx$

1.2.1 THE BOURNE SHELL (sh)

The Bourne shell (sh) written by Steve Bourne at AT&T Bell Lab, is the original UNIX shell. It is the 'preferred shell programming because of its compactness and speed.

The Bourne shell is the Solaris OS default shell. It is the standard shell for Solaris system administration scripts. For the Bourne shell the:

- Command full-path is **/bin/sh** and **/sbin/sh**.
- Non-root user default prompt is **$**.
- Root user default prompt is #.

DISADVANTAGES

Bourne shell (sh) drawbacks are as follows:

- It lacks features for interactive use such as the ability to recall previous commands (history).
- It lacks built-in arithmetic and logical expression handling.

1.2.2 Tcsh/Csh SHELL

Tcsh is enhanced **C** shell, it can be used as a interactive login shell and shell script command processor.

Tcsh has the following features:

i. C like syntax
ii. Command-line editor
iii. Programmable word and filename completion

iv. Spelling correction

v. Job control

```
TCSH(1)                    General Commands Manual                    TCSH(1)

NAME
       tcsh - C shell with file name completion and command line editing

SYNOPSIS
       tcsh [-bcdefFimnqstvVxX] [-Dname[=value]] [arg ...]
       tcsh -l

DESCRIPTION
       tcsh  is  an enhanced but completely compatible version of the Berkeley
       UNIX C shell, csh(1).  It is a command language interpreter usable both
       as an interactive login shell and a shell script command processor.  It
       includes a command-line editor (see The command-line editor), program-
       mable word completion (see Completion and listing), spelling correction
       (see Spelling correction), a history mechanism (see  History  substitu-
       tion),  job  control  (see Jobs) and a C-like syntax.  The NEW FEATURES
       section describes major enhancements of tcsh over  csh(1).   Throughout
       this  manual, features of tcsh not found in most csh(1) implementations
       (specifically, the 4.4BSD csh) are labeled  with  `(+)',  and  features
       which are present in csh(1) but not usually documented are labeled with
       `(u)'.

 Manual page tcsh(1) line 1 (press h for help or q to quit)
```

1.2.3 Ksh SHELL

Ksh stands for **Korn shell** and was designed and developed by **David G. Korn**. It is a complete, powerful, high-level programming language and also an interactive command language just like many other Unix/GNU Linux shells.

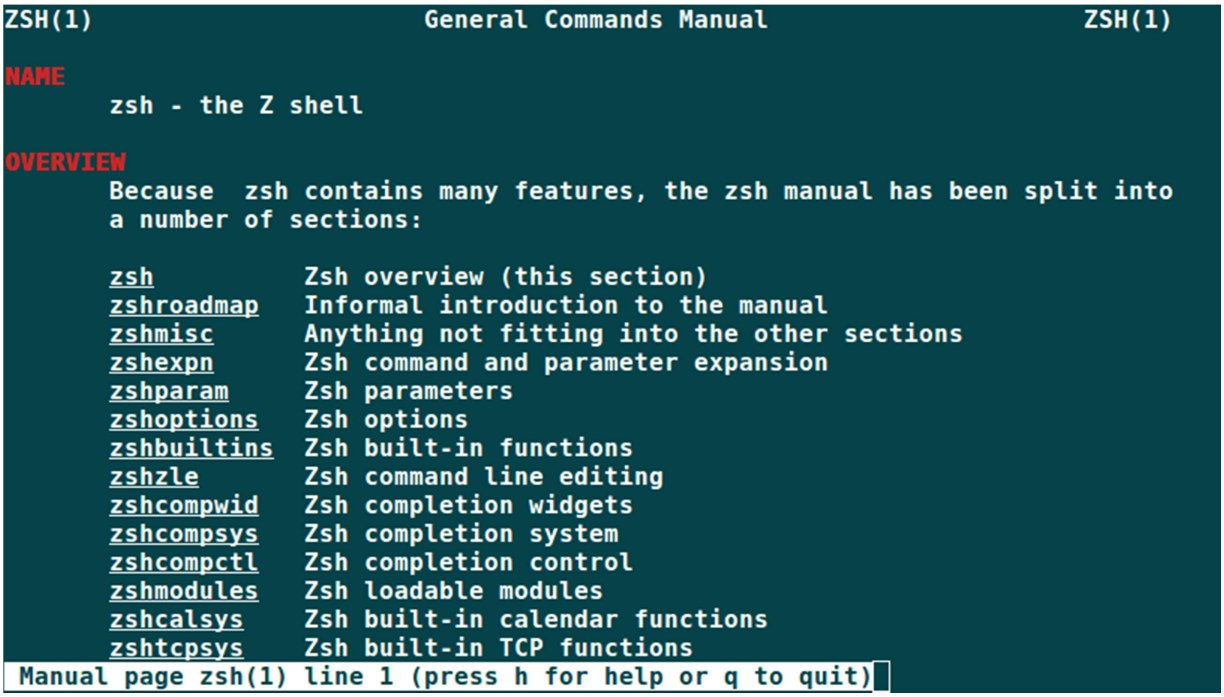
```
KSH(1)                      General Commands Manual                     KSH(1)

NAME
       ksh, ksh93 - KornShell, a command and programming language

SYNOPSIS
       ksh  [  ±abcefhikmnoprstuvxBCDP ] [ -R file ] [ ±o option ] ... [ - ] [
       arg ... ]
       rksh [ ±abcefhikmnoprstuvxBCD ] [ -R file ] [ ±o option ] ...  [ - ]  [
       arg ... ]

DESCRIPTION
       Ksh  is  a command and programming language that executes commands read
       from a terminal or a file.  Rksh is a restricted version of the command
       interpreter  ksh;  it is used to set up login names and execution envi-
       ronments whose capabilities are more controlled than those of the stan-
       dard  shell.   Rpfksh  is a profile shell version of the command inter-
       preter ksh; it is used to to execute commands with the attributes spec-
       ified by the user's profiles (see pfexec(1)).  See Invocation below for
       the meaning of arguments to the shell.

   Definitions.
       A metacharacter is one of the following characters:
 Manual page ksh(1) line 1 (press h for help or q to quit)
```

1.2.4 Zsh Shell

Zsh is designed to be interactive and it incorporates many features of other Unix/GNU Linux shells such as **bash**, **tcsh** and **ksh**.

It is also a powerful scripting language just like the other shells available. Though it has some unique features that include:

i. Filename generation
ii. Startup files
iii. Login/Logout watching
iv. Closing comments
v. Concept index
vi. Variable index
vii. Functions index
viii. Key index and many more that you can find out in man pages



1.2.5 Fish

Fish in full stands for "**friendly interactive shell**" and was authored in 2005. It was intended to be fully interactive and user friendly, just like the other shells, it has some pretty good features that include:

i. Man page completions
ii. Web based configuration

iii. Auto-suggestions

iv. Fully scriptable with clean scripts

v. Support for term256 terminal technology

1.2.6 BASH SHELL

Bash stands for **Bourne Again Shell** and it is the default shell on many Linux distributions today. It is also a sh-compatible shell and offers practical improvements over "sh" for programming and interactive use which includes:

i. Command line editing

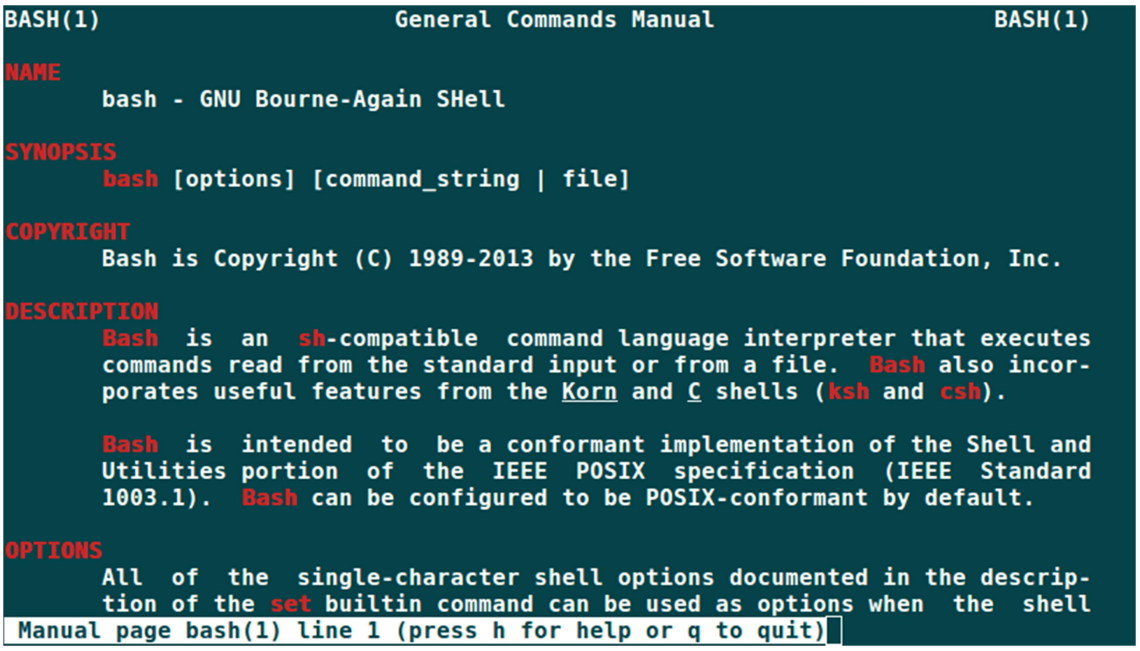
ii. Job Control

iii. Unlimited size command history

iv. Shell Functions and Aliases

v. Unlimited size Indexed arrays

vi. Integer arithmetic in any base from two to sixty-four


BASH(1) General Commands Manual BASH(1)
NAME
bash - GNU Bourne-Again SHell
SYNOPSIS
bash [options] [command_string | file]
COPYRIGHT
Bash is Copyright (C) 1989-2013 by the Free Software Foundation, Inc.
DESCRIPTION
Bash is an sh-compatible command language interpreter that executes commands read from the standard input or from a file. Bash also incorporates useful features from the Korn and C shells (ksh and csh).
Bash is intended to be a conformant implementation of the Shell and Utilities portion of the IEEE POSIX specification (IEEE Standard 1003.1). Bash can be configured to be POSIX-conformant by default.
OPTIONS
All of the single-character shell options documented in the description of the set builtin command can be used as options when the shell
Manual page bash(1) line 1 (press h for help or q to quit)