

## International Journal of Computer Science and Mobile Computing



A Monthly Journal of Computer Science and Information Technology

ISSN 2320-088X

*IJCSMC, Vol. 4, Issue. 4, April 2015, pg.458 – 462*

### **RESEARCH ARTICLE**

# Shell Script & Advance Features of Shell Programming

**Kirti Kaushik\***

Roll No.15903, CS, Department of Computer science, Dronacharya College of Engineering, Gurgaon-123506, India  
Email: [kaushikkirti44@gmail.com](mailto:kaushikkirti44@gmail.com)

**Jyoti Yadav**

Roll No. 15040, CS, Department of Applied Computer science, Dronacharya College of Engineering, Gurgaon-123506, India  
Email: [riya.ana90@gmail.com](mailto:riya.ana90@gmail.com)

**Kriti Bhatia**

Roll No. 15048, CS, Department of Applied Computer science, Dronacharya College of Engineering, Gurgaon-123506, India  
Email: [kritibhatia02@gmail.com](mailto:kritibhatia02@gmail.com)

*Abstract-- In this research paper, the idea of shell scripting and writing computer programs is examined and different parts of shell programming are likewise contemplated. A shell script is a PC system intended to be controlled by the UNIX shell which is a charge line translator. The different tongues of shell scripts are thought to be scripting dialects. Regular operations performed by shell scripts incorporate document control, program execution, and printing content. A shell script can give an advantageous variety of a framework order where unique environment settings, charge alternatives, or post-transforming apply naturally, yet in a manner that permits the new script to still go about as a completely typical UNIX summon. The real ideas like Programming in the Borne and C-shell, in which it would be clarified that how shell programming could be possible in Borne and C-shell. This paper likewise experiences the study about root of Shell Programming, about the Shell Programming Constructs and development highlights of shell writing computer programs are additionally contemplated and took after. In this paper, it would likewise be talked about that how Shell Programming in how many ways is helpful in the UNIX Shell.*

## INTRODUCTION

A shell script is a PC system intended to be controlled by the UNIX shell, a charge line translator. The different lingos of shell scripts are thought to be scripting dialects. Ordinary operations performed by shell scripts incorporate document control, program execution, and printing content. In its most essential shape, a shell script can give an advantageous variety of a framework order where extraordinary environment settings, charge choices, or post-transforming apply naturally, however in a manner that permits the new script to still go about as a completely ordinary UNIX summon. At the heart of UNIX is a part whose schedules aren't anything but difficult to utilize straightforwardly. There are likewise numerous prewritten programs. They're less demanding to utilize, yet they don't comprehend the utilization of '\*' as a special case character, furthermore, you require a proofreader when you write summons in case you're going to utilize the summon line. The "shell" is the interface between the client and the framework. The shell isn't just a line editorial manager and an order line translator that sees trump cards; its additionally a dialect with variables, exhibits, capacity sand control structures. Summon line sweep be put into a document and executed. These purported "shell scripts" can rapidly be composed and tried and ought to be attempted in relationship with other standard UNIX utilities before setting out on a more elevated amount dialect, in any event for prototyping purposes

Shell scripts permit a few charges that would be entered physically at a charge line interface to be executed consequently, and without needing to sit tight for a client to trigger every phase of the succession. For instance, in a catalog with three C source code documents, instead of physically running the four summons needed to fabricate the last program from them, one could rather make a C shell script, here named form and kept in the index with them, which would incorporate them naturally.

The script would permit a client to spare the record being altered, stop the proofreader, and afterward simply run. Construct to make the overhauled system, test it, and afterward come back to the manager. Since the 1980s or somewhere in the vicinity, on the other hand, scripts of this sort have been supplanted with utilities like make which are particular for building projects. Shell scripts frequently serve as a beginning stage in programming advancement, and are regularly subject to change later to an alternate hidden usage, most usually being changed over to Perl, Python, or C. The mediator mandate permits the execution point of interest to be completely covered up inside the script, as opposed to being uncovered as a filename

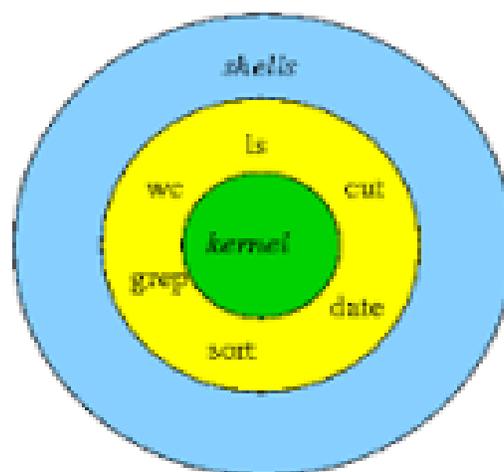


Fig1. Shell & Kernel in Unix Operating System

## RELATED WORK

There are different operations performed under shell programming, some of them are record control, program execution, and printing content. Each of the operations performed under shell writing computer programs is considered in subtle element.

## FILE MANIPULATION IN SHELL PROGRAMMING

This prologue to content control on UNIX stages gives an outline of some basic summons generally accessible and introduced standard on most UNIX-based discharges. Commonly these standard utilities are disregarded for more current content processors, for example, Perl, Python, or Ruby, which are not generally introduced on a framework. An early on audit of these instruments help spractitioner who are learning UNIX or Linux or the individuals who may be looking to recharge overlooked information

## PROGRAM EXECUTION IN SHELL PROGRAMMING

A UNIX shell is a summon dialect mediator, the main role of which is to decipher order lines wrote at a terminal into framework activities. The shell itself is a system through which different projects are conjured. Albeit there are a few distinctive UNIX shells, among them the C shell, the Bourne shell and the Korn shell, the one most every now and again utilized on Blake inside Berkeley

UNIX is the C shell.

The shell has some inherent capacities, which it performs straightforwardly, however most orders that you enter cause the shell to execute programs that are outside to the shell. This separates the shell from other order translators, on the grounds that it is only one more client project while it works solely as a component for summoning different projects.

The UNIX shell project deciphers client orders, which are either specifically entered by the client, or which can be perused from a record called the shell script or shell program. Shell scripts are deciphered, not gathered. The shell peruses summons from the script line every line and scans for those orders on the framework, while a compiler changes over a system into machine meaningful structure, an executable document - which might then be utilized as a part of a shell script. Aside from passing summons to the portion, the primary undertaking of a shell is giving a client domain, which can be arranged separately utilizing shell asset arrangement documents.

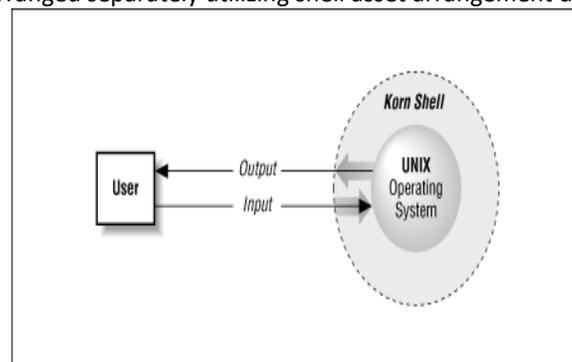


Fig2. Shell in UNIX operating system

## PROGRAMMING IN BOURNE AND C SHELL

A Bourne Shell Script is a file containing a series of Bourne Shell commands, as well as control structures such as if statements, while loops, etc. Parameters can be passed to the script and data can be read from the keyboard. The Bourne shell (sh) is shell, or command-line interpreter, for computer operating systems.

The Bourne shell was the default UNIX shell of Unix Version 7. Most Unix-like systems continue to have /bin/sh—which will be the Bourne shell, or a symbolic link or hard link to a compatible shell even when other shells are used by most users.

Developed by Stephen Bourne at Bell Labs, it was a replacement for the Thompson shell, whose executable file had the same name—sh. It was released in 1977 in the Version 7 Unix release distributed to colleges and universities. Although it is used as an interactive command interpreter, it was also intended as a scripting language and contains most of the features that are commonly considered to produce structured programs.

Example of Bourne shell programming is given below:

To read commands from the terminal and process them:

```
#!/bin/sh
```

```
# usage: process sub-directory
```

```
dir=`pwd`
```

```
for i in *
do
if test -d $dir/$i
then
cd $dir/$i
while echo "$i:"
read x
do
eval $x
done
cd ..
fi
done
```

The user types the command:

### *A. Process Sub-Directory*

This script will read and process commands in the named sub-directory. The user is prompted to supply the name of the command to be read in. This command is executed using the builtin eval function.

The C shell (csh or the improved version, tcsh, on most machines) is a UNIX shell that was created by Bill Joy while he was a graduate student at University of California, Berkeley in the late 1970s. It has been distributed widely, beginning with the 2BSD release of the BSD Unix system that Joy began distributing in 1978. Other early contributors to the ideas or the code were Michael Ubell, Eric Allman, Mike O'Brien and Jim Kulp.

The C shell is a command processor typically run in a text window, allowing the user to type commands. The C shell can also read commands from a file, called a script. Like all Unix shells, it supports filename wild-carding, piping, documents, command, variables and control structures for condition-testing and iteration. What differentiated the

C shell from others, especially in the 1980s, were its interactive features and overall style. Its new features made it easier and faster to use. The overall style of the language looked more like C and was seen as more readable.

Example of C shell programming is given below:

```
#!/bin/csh
If($days> 365) then
Echo This is over a year.
Endif
```

## **ADVANTAGES OF SHELL PROGRAMMING**

- A. Easy to use.
- B. Quick start and interactive debugging.
- C. Time saving.
- D. System admin task automation.
- E. Shell scripts can execute without any additional effort on nearly any modern UNIX / Linux / BSD / Mac OS X operating system
- F. as they are written an interpreted language.

## **DISADVANTAGES OF SHELL PROGRAMMING**

- A. Compatibility problem between different platforms.
- B. Slow execution speed.

## **REFERENCES**

- [1] <http://programmers.stackexchange.com/>
- [2] <http://techforum4u.com/>
- [3] <http://linuxshellscripting.blogspot.in/>
- [4] Concept and applications of UNIX- SumitabhaDas
- [5] Windows NT shell scripting- Timothy Hill
- [6] Shell Scripting Recipes: A Problem-Solution Approach- Chris F.A Johnson.