Lab Manual

for

System Administration Lab

4139

Diploma In Computer Engineering

4th Semester

by

SITTTR

Kalamassery

STATE INSTITUTE OF TECHNICAL TEACHERS TRAINING AND RESEARCH

GENERAL INSTRUCTIONS

Rough record and Fair record are needed to record the experiments conducted in the laboratory. Rough records are needed to be certified immediately on completion of the experiment. Fair records are due at the beginning of the next lab period. Fair records must be submitted as neat, legible, and complete.

INSTRUCTIONS TO STUDENTS FOR WRITING THE FAIR RECORD

In the fair record, the index page should be filled properly by writing the corresponding experiment number, experiment name , date on which it was done and the page number.

On the **right side** page of the record following has to be written:

1. **Title**: The title of the experiment should be written in the page in capital letters.

2. In the left top margin, experiment number and date should be written.

3. **Aim**: The purpose of the experiment should be written clearly.

4.**Apparatus/Tools/Equipments/Components used:** A list of the Apparatus/Tools /Equipments /Components used for doing the experiment should be entered.

5. **Principle**: Simple working of the circuit/experimental set up/algorithm should be written.

6. **Procedure:** steps for doing the experiment and recording the readings should be briefly described(flow chart/programs in the case of computer/processor related experiments)

7. **Results**: The results of the experiment must be summarized in writing and should be fulfilling the aim.

8. **Inference** : Inference from the results is to be mentioned.

On the **Left side** page of the record following has to be recorded:

1. **Circuit/Program**: Neatly drawn circuit diagrams/experimental set up.

2. **Design**: The design of the circuit/experimental set up for selecting the components

should be clearly shown if necessary.

3. **Observations:** i) Data should be clearly recorded using Tabular Columns.

ii) Unit of the observed data should be clearly mentioned

iii) Relevant calculations should be shown. If repetitive calculations are needed, only show a sample calculation and summarize the others in a table.

4. **Graphs :** Graphs can used to present data in a form that show the results obtained, as one or more of the parameters are varied. A graph has the advantage of presenting large

amounts of data in a concise visual form. Graph should be in a square format.

GENERAL RULES FOR PERSONAL SAFETY

1. Always wear tight shirt/lab coat , pants and shoes inside workshops.
2. REMOVE ALL METAL JEWELLERY since rings, wrist watches or bands, necklaces, etc. make excellent electrodes in the event of accidental contact with electric power sources.
3. DO NOT MAKE CIRCUIT CHANGES without turning off the power.
4. Make sure that equipment working on electrical power are grounded properly.
5. Avoid standing on metal surfaces or wet concrete. Keep your shoes dry.
6. Never handle electrical equipment with wet skin.
7. Hot soldering irons should be rested in its holder. Never leave a hot iron unattended.
8. Avoid use of loose clothing and hair near machines and avoid running around inside lab .

TO PROTECT EQUIPMENT AND MINIMIZE MAINTENANCE:

DO: 1. SET MULTIRANGE METERS to highest range before connecting to an unknown source.

2. INFORM YOUR INSTRUCTOR about faulty equipment so that it can be sent for repair.

DO NOT: 1. Do not MOVE EQUIPMENT around the room except under the supervision of an instructor.

INDEX

PART A

LINUX COMMANDS

Ex. No: 1 LOG ON AND LOG OFF

Aim/Objective :

Write a procedure to Logon and Logoff to linux in both Text mode and graphical mode.

Apparatus Required:

Hardware Requirements – Intel core II CPU
Software Requirements – Linux Operating System

Procedure:

In text mode:
Logon to Linux:
Step 1: A prompt ($) is appear on the screen.
Step 2: Type the username and password and press enter.

Logoff to Linux:
Step 1: At the prompt ($) type logout.

In Graphical mode:
Logon to Linux:
Step 1: A graphical login screen as appear on the screen.
Step 2: type Username and press enter.
Logoff to Linux:
Step 1: Select _ System menu _ click _ logout

Result:
Thus the above procedure has been checked successfully with system.

Ex. No: 2 DIRECTORY MANAGEMENT COMMANDS

Aim/Objective :
Write a syntax and usage the directory management commands with all options.

Apparatus Required:
Hardware Requirements – Intel core II CPU
Software Requirements – Red-hat Linux

Procedure:

Command: ls
Usage : It is used to display the files in the current working directory.
Syntax : ls [Options]
[Options] :
-l - list the files in the long format
-a - list all entries, including the hidden files
-d - list the directory files instead of its contents
-t - lists in order of last modification time
Examples : $ls
Output: list all files in specified directory.

Command: cd
Usage : It is used to Change from the current Working directory to any other directory Specified.
Syntax : cd [DIRECTORY]

Command: pwd
Usage : It is used to display the full path for the Current working directory.
Syntax : pwd
Example : $pwd
Output :To view the directory path.

Command: mkdir
Usage : It is used to create a new directory.
Syntax : mkdir <directory name>
Example : $mkdir newdir
Output : Creates directory newDir

Command : rmdir
Usage : It is used to remove a directory specified in the command line. It requires the specified directory to be empty before removing it.
Syntax : rmdir <directory name>
Example: $rmdir newdir
Output : deletes the directory newDir. newDir should be empty

Result:
Thus the above Commands have been executed and checked the output successfully.

Ex. No: 3 FILE MANAGEMENT COMMANDS

Aim/Objective :
Write a command with syntax and usage the file management commands.

Apparatus Required:
Hardware Requirements – Intel core II CPU
Software Requirements –Linux

Procedure:

Command : cat
Usage : This command is used to display the contents of a small file on terminal.
Syntax : cat <option> <file name>
Option : -s –Warning about non existing file.
Example: $cat sample3.txt
Output : To display the content of sample3 text file.

Command : Chmod
Usage : It is used to changing the permissions of the file.
File type: (i) owner (ii) group (iii) others

- Owner has all three permissions.
- Group has read and writes.
- Others have only read permission.
- read permission – 4, write – 2, execute 1
- eg. rwxrwr=764 , 673 = rwrwxwx

Syntax: chmod [OPTION] [MODE] [FILE]
Example: ˜$ chmod o+x Testing.java
˜$ ls -l Testing.java
Output: -rw-r--r-x 1 srihari srihari 3570 2009-03-23 10:
˜$ chmod 655 Testing.java
˜$ ls -l Testing.java
Output: -rw-r-xr-x 1 srihari srihari 3570 2009-03-23 10:

Command: cp
Usage : This command is used to create duplicate copies of ordinary files.
Syntax : cp [source file] [destination file]
Example: $cp testing.java testing1.java
Output : creates the copy of Testing1.java

Command: mv
Usage : This command is used to move or rename ordinary files and directories.

Syntax : mv [source file] [destination file]
Example: $mv testing.java test.java
Output : renames the file Testing.java to Test.java

Command: rm
Usage : It is used to remove the specified file in directory.
Syntax : rm [source file]
Example: rm file1
Output : To delete the text file name file1.

Command: more
Usage : This command is used to display the output of a command on the screen

page by Page.
Syntax : more [options] filename
Options : n – skip n lines and display another screen [n as an integer]
-h displays all the options in more
Example: $more test.java
Output: To display the output of test.java in page by page.

FILE COMMANDS:

Command: diff
Usage : This command is used to find difference between two files.
Syntax : diff [options..] from-file to-file
Options : -w Ignore white space when comparing lines.
-a Treat all files as text and compare them line-by-line.
-b Ignore changes in amount of white space.
Example: $ diff -w file1.txt file2.txt

Command: cmp
Usage : This command is used to compare two files whether they are identical or not.
- The two files are compared byte by byte and the location of the first mismatch is printed on the screen.
- If two files are identical, then it does not print anything on the screen

Syntax : cmp <file1> <file2>
Example: $ cmp sample1.txt sample2.txt

Command : cat
Usage : This command is used to when supplied with more than one file will concatenate the files without any header information.
Syntax : cat <file name>
Example: $ cat sample3.txt sample4.txt
Output:
/*contents of sameple3.txt*/
/*followed by contents of sample4.txt

Result:

Thus the above Commands have been executed and checked the output successfully.

Ex. No: 4 GENERAL PURPOSE COMMANDS

Aim/Objective :
Write a command with syntax & usage and execute the general purpose commands.

Apparatus Required:
Hardware Requirements – Intel core II CPU
Software Requirements – Red-hat Linux

Procedure

Command: wc (statistic of file)

Usage : This command is used to count lines, words and characters, Depending on the option used.
Syntax : wc [options] [file name]
Options : -l: Number of lines
-w: Number of words
-c: Number of characters
Example : $ wc sample1.txt
Output : 65 2776 17333 sample1.txt
Which means sample1.txt file has 65 lines, 2776 words, and 17333 Characters.

Command: od

Usage : This command is used to display the actual file content on octal value format.
Syntax: od [option] filename
Example: od -c file1.txt

Command: lp

Usage : This command is used to send file to a printer.
Syntax : ip [option] filename
Example : ip sample file
Output : To view the file on printer format.
Command : cal
Usage : This Command is used to see calendar for any specific month or a complete year
Syntax : cal [[month] year]
Example: $ cal April 2009
Output : April 2009

```
Su Mo Tu We Th Fr Sa
1 2 3 4
5 6 7 8 9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30
```

Command : date

Usage : This command is used to display the current date.
Syntax : $ date
Options:
d - The day of the month (1-31)

y - The last two digits of the year
H,M,S - Hour Minute and second respectively
D - the date in mm/dd/yy
Example: $ date
Output: Tue Apr 21 21:33:49 IST 2009
Example: $ date +"%D %H:%M:%S"
Output: 04/21/09 21:35:02

Command: who

Usage : This command is used to list the users currently logged on to the system.
Syntax : $ who
Output:

```
srihari pts/0 2009-04-15 11:58 (:10.129.41.3)
nithin pts/1 2009-04-15 16:09 (:10.129.20.5)
avadhut pts/2 2009-04-13 14:39 (:10.129.45.20)
anil pts/3 2009-04-13 16:32 (:10.129.23.45)
```

Command : tty

Usage : The Linux "tty" command display the name of the connected terminal.
Syntax : tty [*OPTION*]...
Example: [root@RHEL01 ~]# tty

Command : ln

Usage : The ln command is used to create link to a file (or) directory. It helps to provide soft link for desired files.
Syntax : ln [options] existing file (or directory)name new file(or directory)name
Options:
-n - Does not overwrite existing files.
-s - Used to create soft links
Example: ln -s file1.txt file2.txt
Output : Creates a symbolic link to 'file1.txt' with the name of 'file2.txt'.

Result:
Thus the above General purpose Commands have been executed and checked

Ex. No: 5 USING SIMPLE FILTERS

Aim/Objective :
Write a command with syntax & usage and execute using simple filters.

Apparatus Required:
Hardware Requirements – Intel core II CPU
Software Requirements – Red-hat Linux

Procedure :

Command : pr
Usage : This command is used to convert text files for printing type.
Syntax : pr [*OPTION*]... [*FILE*]...
Example: pr sample file
Output: To view the specified file on printing method.
Command : head
Usage : This command is used to displays the top of the file. When used without any option, it will display first 10 lines of the file.
Syntax : $ head –n filename
Example: $ head -n 20 sample1.txt
Output: /* will display first 20 lines*/
when used without any option it will display first 10 lines of the file.
Command : tail
Usage : The tail command used to displays the end of the file. By Default it will display last 10 lines of the file
Syntax :$ tail –n filename
Example: $ tail -n 20 sample1.txt
Output : /* will display last 20 lines*/
when used without any option it will display last 10 lines of the file.
Command : cut
Usage : (cutting columns)
This command can be used to cut the columns from a file with -c option.
Syntax :$ cut -c [numbers delimited by comma or range] <file name>
Option : -f option you can cut the fields delimited by some character.
-d option is used to specify the delimiter and -f option used to specify the field number.
Example : $ cut -c 1, 2, 3-5 students.txt
Output : 1 ani
2 das
3 shu
4 sin
Example : $ cut -d" " -f1, 4 students.txt
Output : 1 Mtech
2 Btech
3 Mtech
Command : paste
Usage : The paste command will paste the contents of the file side by side

Syntax : $ paste <file1 name> <file2 name>…..<filen name>
Example: $ paste cutlist1.txt cutlist2.txt

```
Output: 1 Mtech 1 anil H1
2 Btech 2 dasgupta H4
3 Mtech 3 shukla H7
4 Mtech 4 singhvi H12
5 Btech 5 sumit H13
```

Command : nl

Usage : It is numbering lines tool, it's default action is to write it's input (either the file names given as an argument, or the standard input) to the standard output.

Syntax : nl [options]... [File]

Option :-h header type
-f footer type
-i increment type

Example: $ samplefile

Command : sort (ordering a file)

Usage : It is used to sort re-orders lines in ASCII collating sequences, white spaces, then numerals, uppercase and finally lowercase.

Syntax : sort [-options] filename

Options: -r display input taken from the reverse alphabetical order
-f output arranged in alphabetical order
-n arrange the input according to numerical values

Example: $ sort –n sample.txt

Result:

Thus the above Commands have been executed using simple filters and checked the output successfully.

Ex. No: 6 ADVANCED FILTERS

Aim/Objective :
Write a command with syntax & usage and execute the advance filters such as grep, egrep, fgrep.

Apparatus Required:
Hardware Requirements – Intel core II CPU
Software Requirements – Red-hat Linux

Procedure

Command: grep

Usage : This command is used to search files for lines that match a given pattern.
Syntax : grep file pattern
Example : grep sample.txt

Command : egrep

Usage : It is used to extended grep is the same as grep -E. This command uses a somewhat different, extended set of Regular Expressions, which can make the search a bit more flexible. It also allows the Boolean | (or) operator.
Syntax : egrep [options] 'PATTERN' files …
Example: $egrep [JNR] employee.txt
Output :

```
200 Jason Developer Technology $5,500
400 Nisha Manager Marketing $9,500
500 Randy DBA Technology $6,000
```

Command : fgrep

Usage : fast grep is the same as grep -F. It does a literal string search (no Regular Expressions), which generally speeds things up a bit.
Syntax : fgrep <options>...
Option: -b -Precede each line by the block number on which it was found. This can be useful in locating block numbers by context (first block is 0).
-c - Print only a count of the lines that contain the pattern.
-h - Suppress printing of files when searching multiple files.
Example: $fgrep [option] sample.txtx

Result:
Thus the above Commands have been executed using advanced filters and checked the output successfully.

Ex. No: 7 PROCESS MANAGEMENT COMMANDS

Aim/Objective :
Write a command with syntax & usage then execute the ps command, process management commands: &nohub, kill, nice.

Apparatus Required:
Hardware Requirements – Intel core II CPU
Software Requirements – Red-hat Linux

Procedure:

Command : ps

Usage: This command commonly used to list down all the process which are currently running in a machine.
Syntax : (Display Process info) ps [option]
Option: -u option to displays the process that belongs to a specific username
-e - to display all the processes.
-f - to display full format listing.
-l - List all the keyword options
Example: $ ps –e f
Output:

```
root 26551 5 0 Feb10 ? 00:03:41 [pdflush]
root 26570 5 0 Feb10 ? 00:00:20 [pdflush]
root 30344 3382 0 Feb21 ? 00:00:11 sshd: root@pts/14
```

The above example shows the options of ps command to get all the processes.

Command : nohup

Usage : The nohup utility which allows to run command./process or shell script that can continue running in the background after you log out from a shell.
Syntax : nohup command-name &
Where,

- Command-name: is name of shell script or command name. You can pass argument to command or a shell script.
- & : nohup does not automatically put the command it runs in the background; you must do that explicitly, by ending the command line with an & symbol.

Example:
1) Login to remote server

```
$ ssh user@remote.server.com
```

2) Execute script called pullftp.sh

```
# nohup pullftp.sh &
```

3) Type exit or press CTRL + D exit from remote server.

```
# exit
```

Command : kill

Usage: Kill command is use to send signal to a process or to kill a process. Typically use kill -SIGNAL PID, where you know the PID of the process. There are other ways to effectively kill a process —

- killing a process by name,

- killing a process by specifying part of the name,
- killing a process by pointing out the process with cursor etc.,

Syntax : kill [options] [pids|commands]

Options : -a -> all process matching with PIDs between 2 and 35767

- p -> list only the PID

PID - > PID identification number

Example: $kill pid

21

Command : nice

Usage : To run a program with modified scheduling priority. Kernel decides how much processor time is required for a process based on the nice value.

Possible nice value range is: -20 to 20

Syntax: nice [*OPTION*] [*COMMAND* [*ARG*]...]

Option: -n, --adjustment=*ADJUST*

Increment priority by ADJUST first

--Help - display this help and exit

--Version - output version information and exit

Result:

Thus the above process management Commands have been executed and checked the output successfully.

Ex. No: 8 COMMUNICATION COMMANDS

Aim/Objective :
Write a command with syntax & usage then execute the communication commands.

Apparatus Required:
Hardware Requirements – Intel core II CPU
Software Requirements – Red-hat Linux

Procedure:

Command : news

Usage : This command is used to Display message to all the users.
Syntax : $ news [options] [news item]
Options : -a displays all of the news items
-s displays the names of all of the news items.
-c displays a count of all of the news
Example: A typical news item is stored in the file winner in that directory.

```
$ Cat / user /news/winner
MCA guys are winners in s/w contest at GRD Granddays at 8pm
$
```

When a user invokes the news command, This is see by use;

```
$ news
Winner (root) wed dec 10 12:53:50 MCA guys are winners in s/w
contest at GRD Granddays at 8pm.
$
```

Any news item already seen once cant be displayed again, so news either outputs this message or simply returns the prompt if all news items have been seen before.

Command : write

Usage : This command is used to send a message to another user.
• Write allows you to communicate with other users, by copying lines from your terminal to theirs.
• When you run the write command, the user you are writing to gets a message of the form:
Syntax : write user [ttyname]
Example: write hope
Output : Typing the above command would start a talking session with the user hope. If you type your name, you can echo back write - send a message to another user

Command : mail

Usage : This command is used to send and receive mail.
Syntax : Sending:
Mail [-t] [-w] [-m message_type] recipient ...
rmail [-t] [-w] [-m message_type] recipient ...
Reading:
Mail [-e] [-h] [-p] [-P] [-q] [-r] [-f file]
Option: -v - Verbose mode. The details of delivery are displayed on the user's

terminal.
-i - Ignore tty interrupt signals. This is particularly useful when using mail on noisy phone lines.
-c - Send carbon copies to *list* of users.
-f - Read in the contents of your *mbox* (or the specified file) for processing; when you *quit* mail writes undeleted messages back to this file.
Example: $ mail - "Hello" boss@yahoo.com

Command : wall

Usage : This command is used to read a message from standard input and send the message to all users currently logged in.

Syntax : # wall
/*message*/

Example: For example, you can warn users of unexpected shutdowns:

```
# Wall
The system is coming down in ten minutes for unexpected
maintenance. Please log off soon in order to save your files.
<CTRL-d>
#
```

Command : calender

Usage : This command is used to reads the calendar file and displays any line in the file that contains today's or tomorrow's date.
The calendar file is user-created and must be in the same directory from which you run the calendar command. Typically, the calendar file resides in your home directory.

Syntax : calendar [-]

Result:
Thus the above communication Commands have been executed and checked the output successfully.

Ex. No: 9 DEVICE PATTERN

Aim/Objective :
To execute device pattern using Meta character to match each of the following situation:
i. All two character filenames.
ii. All filenames consisting of two lowercase letters.
iii. All filenames ending with c.
iv. All filenames beginning with c and ending with a digit.
v. All filenames beginning with p and having at somewhere.

Apparatus Required:
Hardware Requirements – Intel core II CPU
Software Requirements – Red-hat Linux

Procedure:

Command :
i) ALL TWO CHARACTER FILENAMES:

```
Syntax : $ ls ??
Example: [root@sample root]$ ls ??
F1 f2 f6 f9
26
```

ii) ALL FILENAMES CONSISTING OF TWO LOWERCASE LETTERS.

```
Syntax: $ls *[a-z]??
Output:
[root@sample root]$ ls *[a-z]??
1 anaconda-ks.cfg f1 f2 f3 addn copy jey
Greet1 f9 doc C3 c5 c6 purple
linux:
mydir:
f1 f2
newproject:
f4
```

iii) ALL FILENAMES ENDING WITH C.

```
Syntax: $ ls *c

Output:
[root@sample root]$ ls *c
Doc
```

iv) ALL FILENAMES BEGINNING WITH C AND ENDING WITH DIGIT

```
Syntax:$ ls c*[0-9]
Output:
[root@sample root]$ ls c*[0-9]
C3 c5 c6
```

v) ALL FILENAMES BEGINNING WITH P AND HAVING AT SOMEWHERE

```
Syntax: $ls p*?p*
Output:
```

```
[root@sample root]$ ls p*?p*
Purple
```

Result:
Thus the above Commands have been executed with device pattern using Meta character to match different situation and checked the output successfully.

PART B SHELL SCRIPTS

Ex. No: 10 Display the decremented numerical value of N

Aim:
To write a shell script that accepts a numerical value N. Then display the decrementing value of N till it reaches 0.

Apparatus Required:
Hardware Requirements: Intel core II CPU
Software Requirements: Red-Hat Linux

Procedure:
1. Create a new vi editor file.
2. Get the integer value of n.
3. Check the n value greater than zero then print the n value.
4. Decrement the variable n.
5. Repeat the steps up to this condition false.(i.e. n>0)
6. Print the result.

Program:

```
#!bin/bash
echo “ENTER THE INTEGER VALUE: ”
read n
while [ $n –ge 0 ]
do
echo “$n”
let n—
done
```

29

Output:

```
[root@sample raja]# bash addn
ENTER THE INTEGER VALUE:
6
6
5
4
3
2
1
0
```

Result:
Thus the Shell script program to display the decremented integer value was executed successfully.

Ex.No: 11 File Moment Using Command Line Arguments

Aim:
To write a shell-script program that takes three command line arguments. The first argument is the name of the destination file and the other two arguments are names of files to be placed in the destination file.

Apparatus Required:
Hardware Requirements: Intel core II CPU
Software Requirements: Red-Hat Linux

Algorithm:

```
1. Create a new vi editor file.
2. Get the name of the destination file, 1st source file and 2nd source file.
3. Concatenate the two files and copied into the destination file.
4. Check the file status with 0.
5. If it is equal then print “File Copied successfully” otherwise print
“Problem copying file”.
```

Program:

```
#!/bin/bash
echo “Enter the name of the destination file : “
read dest
echo “Enter the name of the 1st source file : “
read src1
echo “Enter the name of the 2nd source file : “
read src2
cat $src1 $src2 > $dest
status=$?
If [ $status –eq 0 ]
Then
echo “File Copied Successfully”
Else
echo “Problem copying file”
fi
```

Output:

```
Enter the name of the destination file: F1
Enter the name of the 1st source file: F2
Enter the name of the 2nd source file: F3
File copied successfully.
```

Result:
Thus the program to concatenate the two files into another file using command line arguments was executed successfully.

Ex. No: 12 Printing the file Content

Aim:
To Write a shell-script program to print content of the file from given line number to next given number of lines.

Apparatus Required:
Hardware Requirements: Intel core II CPU
Software Requirements: Red-Hat Linux

Algorithm:

```
1. Create a new vi editor file.
2. $# -eq 0, then print the Error msg.
3. If $# -eq 3
a. If $# -eq 3,
then print the content from the given line number.
b. Else print the Error opening file
4. Else print the missing arguments.
```

Program:

```
#!/bin/bash
if [ $# -eq 0 ]
then
echo "$0:Error command arguments missing!"
echo "Usage: $0 start_line uptoline filename"
echo "Where start_line is line number from which you would like to
print file"
echo "uptoline is line number upto which would like to print"
echo "For eg. $0 5 5 myfile"
echo "Here from myfile total 5 lines printed starting from line no. 5 to"
echo "line no 10."
exit 1
fi
if [ $# -eq 3 ]; then
if [ -e $3 ]; then
tail +$1 $3 | head -n$2
else
echo "$0: Error opening file $3"
exit 2
fi
else
echo "Missing arguments!"
fi
```

Output:

```
[root@sample jeya]# bash print file 5 5 copy
read src1
echo “Enter the name of the 2nd source file : “
read src2
```

```
cat $src1 $src2 > $dest
status=$?
```

Result:

Thus the Shell script program to print the content of the file from the given line number to the next given line number was executed successfully.

Ex.No: 13 Display the Message in Login Session

Aim:
Write a shell script program to say Good morning/ Afternoon /Evening as you log in to system.

Apparatus Required:
Hardware Requirements: Intel core II CPU
Software Requirements: Red-Hat Linux

Algorithm:
1. Create a new vi editor file.
2. Set the value of hour.
3. Compare the hour with time of zero, 12and 18.
4. If it is greater than 0 and less than 12 print the string is GOOD MORNING Then if it is greater than 12 and less than 18 print the string is GOOD AFTERNOON otherwise print the string is GOOD EVENING.

Program:

```
#!/bin/bash
Hour=$(date + “%H”)
If [ $hour –ge 0 –a $hour –lt 12 ]
Then
Greet=”GOOD MORNING,$USER”
elif [ $hour –ge 12 –a $hour –lt 18 ]
Then
Greet=”GOOD AFTERNOON,$USER”
Else
Greet=”GOOD EVENING,$USER”
Fi
Echo $greet
```

Output:

```
[root@sample raja]# bash greeting
GOOD AFTERNOON, root
```

Result:
Thus the Shell script program to say Good morning/Afternoon/Evening as u log into system was executed successfully.

Ex. No: 14 To Print the date in a Decided Format

Aim:
Write a shell script program that print out date information in this order: time, day of the week, day number, year-that is like this 13:55:00 IST sun 9 june1996.

Apparatus Required:
Hardware Requirements: Intel core II CPU
Software Requirements: Red-Hat Linux

Algorithm:
1. Create a new vi editor file.
2. Enter the command is date.
3. Print the today date information.

Program:

```
#!/bin/bash
TODAY=$(date + “%X %a %d %b %Y”)
Echo “----------------------------------------------“
Echo “Today Date : $TODAY”
ECHO “----------------------------------------------“
```

Output:

```
[root@sample jeya]# bash date
-------------------------------------------------------
Today Date : 03: 52 :33 PM Thu 17 May 2012
-------------------------------------------------------
```

Result:

Thus the Shell script program to print the date information was executed successfully.

Ex. No: 15 Tells name and PID

Aim:
To Write a shell script that that tells you its name and PID.

Apparatus Required:
Hardware Requirements: Intel core II CPU
Software Requirements: Red-Hat Linux

Procedure:
1. Create a new vi editor file.
2. Enter the command is Ps –u into file.
3. Print the name and PID information.

Program:

```
#!/bin/bash
Ps –u
```

39

Output:

```
[root@sample jeya]# bash p1
USER PID %CPU %MEM VSZ RSS TTY STAT START
TIME COMMAND
Root 850 0.0 0.9 4300 1664 pts/0 s 11.13 0:02
Bash
```

Result:
Thus the Shell script program was executed successfully

Ex.No: 16 Basic Calculator

Aim:
To Develop a basic math calculator using case statement.
Apparatus Required:
Hardware Requirements: Intel core II CPU
Software Requirements: Red-Hat Linux

Procedure:
1) Create a new file.
2) Read the operands.
3) Select any one operation from the list.
4) Perform the operation.
5) Print the result.

Program:

```
# Implementation of Calculator application
#!bin/bash
j=1
while [ $j -eq 1 ]
do
echo "Enter the First Operand;"
read f1
echo "Enter the second operand:"
read f2
echo "1-> Addition"
echo "2-> Subtraction"
echo "3-> Multiplication"
echo "4-> Division"
echo "Enter your choice"
read n
case "$n" in
1)
echo "Addition"
f3=$((f1+f2))
echo "The result is:$f3"
;;
2)
echo "Subtraction"
let "f4=$f1 - $f2"
echo "The result is:$f4"
;;
3)
echo "Multiplication"
let "f5=$f1 * $f2"
echo "The result is:$f5"
;;
```

```
4)
echo "Division"
let "f6=$f1 / $f2"
echo "The result is:$f6"
;;
esac
echo "Do you want to continue(press:1 otherwise press any key to
quit)"
read j
done
```

OUTPUT:

```
[su@localhost su]$ bash u
Enter the First Operand;
23
Enter the second operand:
23
1-> Addition
2-> Subtraction
3-> Multiplication
4-> Division
42
Enter your choice
1
Addition
The result is:46
Do you want to continue (press:1 otherwise press any key to quit)
1
Enter the First Operand;
20
Enter the second operand:
2
1-> Addition
2-> Subtraction
3-> Multiplication
4-> Division
Enter your choice
2
Subtraction
The result is:18
Do you want to continue(press:1 otherwise press any key to quit)
1
Enter the First Operand;
24
Enter the second operand:
2
1-> Addition
2-> Subtraction
3-> Multiplication
4-> Division
```

```
Enter your choice
3
Multiplication
The result is:48
Do you want to continue(press:1 otherwise press any key to quit)
1
43
Enter the First Operand;
24
Enter the second operand:
12
1-> Addition
2-> Subtraction
3-> Multiplication
4-> Division
Enter your choice
4
Division
The result is:2
Do you want to continue(press:1 otherwise press any key to quit)
q
```

Result:

Thus the above program to develop a calculator application was executed successfully.

Ex. No: 17 Sent the Greeting to user

Aim:
To write shell script that takes a login name as a command line argument and reports to you when that person log in. have it sent a greeting to that person.

Apparatus Required:
Hardware Requirements: Intel core II CPU
Software Requirements: Red-Hat Linux

Procedure:
1. Create a new vi editor file.
2. Enter the command in to file.
3. Print the logged on to the system information.

Program:

```
#!/bin/bash
Echo “The user $argv[1] is logged on to the system on”
Who | grep “$argv[1]” | cut –c22-38
```

Output:

```
[root@sample jeya]# bash e2
The user [1] is logged on to the system on
May 17 11:12
```

Result:
Thus the Shell script program to that that takes a login name as command line argument & reports when that person logs in. Send greeting to that person was executed successfully.

Ex. No: 18 Report Using command line arguments

Aim:
To write a shell script that takes a command line argument and reports on whether it is a directory a file or something else.

Apparatus Required:
Hardware Requirements: Intel core II CPU
Software Requirements: Red-Hat Linux

Procedure:
1. Create a new vi editor file.
2. Get the name of the file or directory.
3. Compare the name of the file or directory.
4. If it is file name print the string is file then if it is directory name print the string is Directory otherwise print the string is Not.

Program:

```
#!/bin/bash
echo "Enter a file name:"
read f
if [ -f $f ]
then
echo "File"
elif [ -d $f ]
then
echo "Directory"
else
echo "Not"
fi
```

Output:

```
[root@sample jeya]# bash report
Enter a file name:
F1
File
```

Result:
Thus the Shell script program to takes a command line argument & reports on whether it's a file or directory or something was executed successfully.

Ex. No: 19 Palindrome Checking

Aim:

Write a program to convert the lowercase characters into uppercase and uppercase characters into lowercase character of a given string.

Apparatus Required:
Hardware Requirements: Intel core II CPU
Software Requirements: Red-Hat Linux

Procedure:
1. Create a new vi editor file.
2) Read the input string.
3) Find the length of the given string.
4) Separate the string character by character using substring function.
5) Reverse the string and store the string temporarily
6) Compare the reversed string with input string.
7) If it is equal then print the given string is palindrome otherwise
print not palindrome

Program:

```
#!bin/bash
echo "Enter the string:"
read str
len=${#str}
k=$len-1
i=0
flag=1
while [ $i -le $((len / 2))
do
s1=${str:$i:1}
s2=${str:$k-i:1}
if [ $s1 != $s2 ]
then
flag=0
fi
let i++
done
if [ $flag -eq 1 ]
then
echo "Palindrome"
Operating System
else
echo "Not Palindrome"
```

Output:

```
Enter the string:
AMMA
Palindrome
```

```
Enter the string:
Lecturer
Not Palindrome
```

Result:

Thus the above program to covert the upper case into lower case and vice versa were executed successfully.

Ex. No: 20 Find the LENGTH of given string

Aim:
To find the length of given string.

Apparatus Required:
Hardware Requirements: Intel core II CPU
Software Requirements: Red-Hat Linux

Procedure:
1. Create a new vi editor file.
2. Read the input string.
3. Find the length and substring of the given string.
4. Initialize the count value (i.e.) count=0.
5. If the substring character is match with any vowels character
Then increase the count value.
6. Print the count value

Program:

```
#!bin/bash
echo "Enter the string "
read str
len=${#str}
echo "$len"
```

Output:

```
Enter the string:
COMPUTER
8
```

Result:
Thus the above program to find the number of vowels present in a given string was executed successfully.

PARTC

LINUX SERVER INSTALLATION

Ex. No: 21 Installation of Linux Server

Aim:
Installation of Linux Server.

Apparatus Required:
Hardware Requirements: Intel core II CPU
Software Requirements: Linux Server Software

Procedure

Installing from CD
The basic steps to install Ubuntu Server Edition from CD are the same as those for installing any operating system from CD. Unlike the *Desktop Edition*, the *Server Edition* does not include a graphical installation program. The Server Edition uses a console menu based process instead.

- Download and burn the appropriate ISO file from the *Ubuntu web site*.

- Boot the system from the CD-ROM drive.

- At the boot prompt you will be asked to select a language.

- From the main boot menu there are some additional options to install Ubuntu Server Edition. You can install a basic Ubuntu Server, check the CD-ROM for defects, check the system's RAM, boot from first hard disk, or rescue a broken system. The rest of this section will cover the basic Ubuntu Server install.

- The installer asks which language it should use. Afterwards, you are asked to select your location.

- Next, the installation process begins by asking for your keyboard layout. You can ask the installerto attempt auto-detecting it, or you can select it manually from a list.

- The installer then discovers your hardware configuration, and configures the network settings using DHCP. If you do not wish to use DHCP at the next screen choose "Go Back", and you have the option to "Configure the network manually".

- Next, the installer asks for the system's hostname.

- A new user is set up; this user will have *root* access through the sudo utility.

- After the user settings have been completed, you will be asked if you want to encrypt your home directory.

- Next, the installer asks for the system's Time Zone.

- You can then choose from several options to configure the hard drive layout. Afterwards you are asked which disk to install to. You may get confirmation prompts before rewriting the partition table or setting up LVM depending on disk layout. If you choose LVM, you will be asked for the size of the root logical volume.

- The Ubuntu base system is then installed.

- The next step in the installation process is to decide how you want to update the system. There are three options:
- *No automatic updates*: this requires an administrator to log into the machine and manually install updates.
- *Install security updates automatically*: this will install the unattended-upgrades package, which will install security updates without the intervention of an administrator.
- *Manage the system with Landscape*: Landscape is a paid service provided by Canonical to help manage your Ubuntu machines. See the *Landscape*4 site for details.
- You now have the option to install, or not install, several package tasks.

- Finally, the last step before rebooting is to set the clock to UTC.

If at any point during installation you are not satisfied by the default setting, use the "Go Back" function at any prompt to be brought to a detailed installation menu that will allow you to modify the default settings. At some point during the installation process you may want to read the help screen provided by the installation system. To do this, press F1.

2.1. Package Tasks

During the Server Edition installation you have the option of installing additional packages from the CD.

The packages are grouped by the type of service they provide.

- DNS server: Selects the BIND DNS server and its documentation.
- LAMP server: Selects a ready-made Linux/Apache/MySQL/PHP server.
- Mail server: This task selects a variety of packages useful for a general purpose mail server system.
- OpenSSH server: Selects packages needed for an OpenSSH server.
- PostgreSQL database: This task selects client and server packages for the PostgreSQL database.
- Print server: This task sets up your system to be a print server.
- Samba File server: This task sets up your system to be a Samba file server, which is especially

suitable in networks with both Windows and Linux systems.

- Tomcat Java server: Installs Apache Tomcat and needed dependencies.
- Virtual Machine host: Includes packages needed to run KVM virtual machines.
- Manually select packages: Executes aptitude allowing you to individually select packages.

For example, to list the packages installed with the *DNS Server* task enter the following:

```
tasksel --task-packages dns-server
```

The output of the command should list:

```
bind9-doc
bind9utils
bind9
```

If you did not install one of the tasks during the installation process, but for example you decide to make your new LAMP server a DNS server as well, simply insert the installation CD and from a terminal:

```
sudo tasksel install dns-server
```

Result:

Linux Server Installed Successfully.

PARTD

WINDOWS SERVER INSTALLATION

Ex. No: 19 Installation of Windows Server

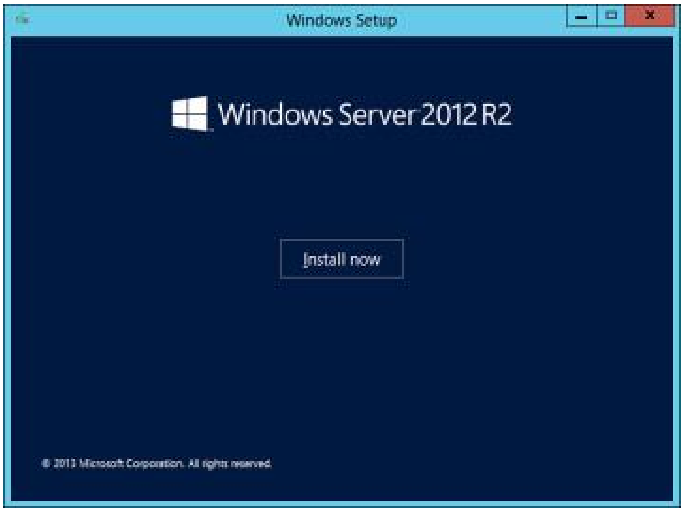
Aim:
Installation of Windows Server.

Apparatus Required:
Hardware Requirements: Intel core II CPU
Software Requirements: Server Software

Procedure

Installing Windows Server 2012 R2
you can install Windows Server 2012 R2 as either Server with a GUI, which presents a full graphical user interface, or Server Core, which presents only a command prompt window.

Installing a Windows Server Core Computer
Windows Server Core includes a minimal version of the server software without the GUI; you perform all configuration tasks from the command prompt. Follow this procedure to install Windows Server Core and perform initial configuration tasks:

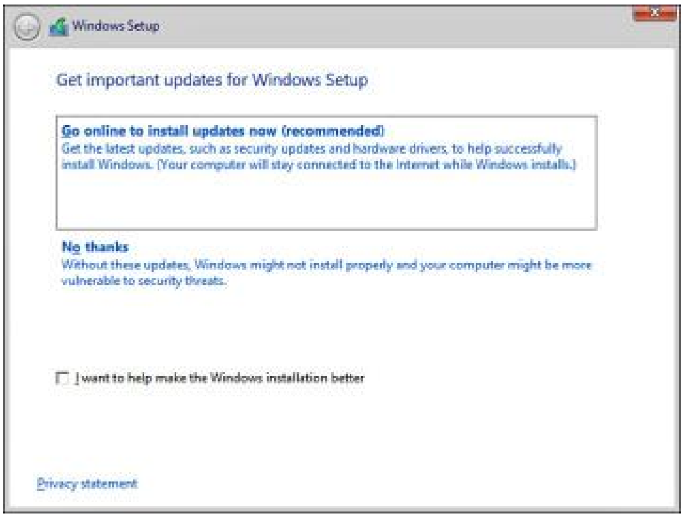


Starting the installation of Windows Server 2012 R2.

1. Insert the Windows Server 2012 R2 DVD-ROM and turn on your computer. You should see a message informing you that Windows is copying temporary files; if not, you should access the BIOS setup program included with your computer and modify the boot sequence so that the computer boots from the DVD.

2. After a few minutes, you receive the Windows Server 2012 R2 screen shown below. Install now to begin the installation.

3. Windows copies temporary files and then displays the Get important updates for Windows Setup screen shown in Figure 2-2. If you're connected to the Internet, select Go online to install updates now (recommended).



Get important updates for Windows Setup

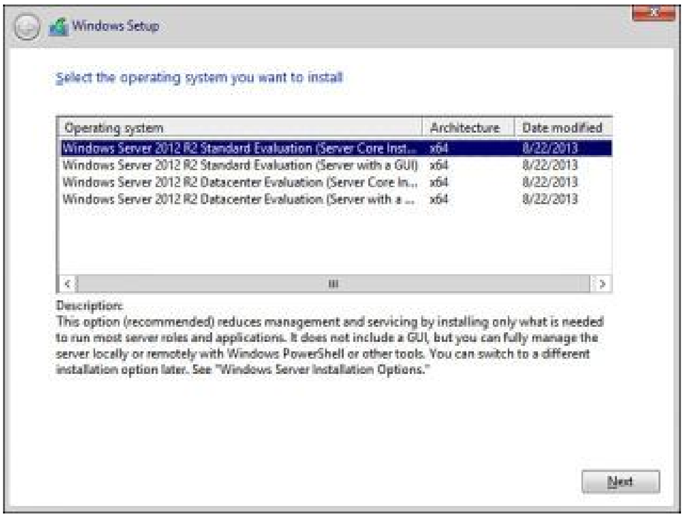
Go online to install updates now (recommended)
Get the latest updates, such as security updates and hardware drivers, to help successfully install Windows. (Your computer will stay connected to the Internet while Windows installs.)

No thanks
Without these updates, Windows might not install properly and your computer might be more vulnerable to security threats.

☐ I want to help make the Windows installation better

4. On the next Install Windows screen, click Install now.

5. You receive the options shown in Figure 2-3, which enable you to install the complete Standard or Datacenter version of Windows Server 2012 R2 with a GUI or Windows Server 2012 R2 Server Core. Select the Windows Server 2012 R2 Datacenter (Server Core Installation) option and then click Next.



Select the operating system you want to install

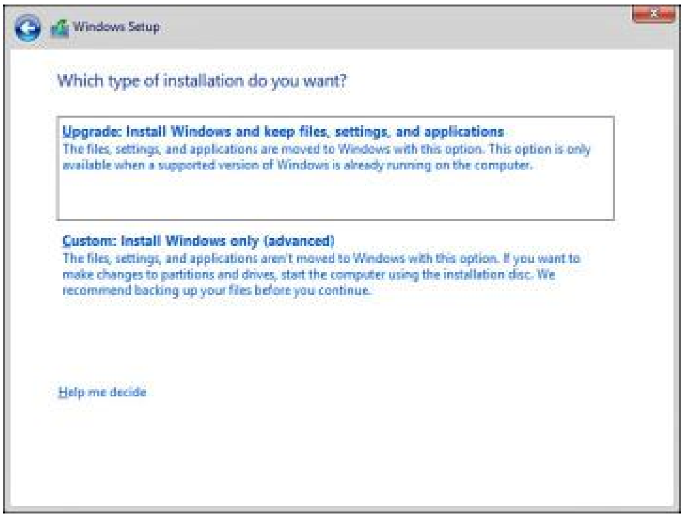
Operating system	Architecture	Date modified
Windows Server 2012 R2 Standard Evaluation (Server Core Inst...	x64	8/22/2013
Windows Server 2012 R2 Standard Evaluation (Server with a GUI)	x64	8/22/2013
Windows Server 2012 R2 Datacenter Evaluation (Server Core In...	x64	8/22/2013
Windows Server 2012 R2 Datacenter Evaluation (Server with a ...	x64	8/22/2013

Description:
This option (recommended) reduces management and servicing by installing only what is needed to run most server roles and applications. It does not include a GUI, but you can fully manage the server locally or remotely with Windows PowerShell or other tools. You can switch to a different installation option later. See "Windows Server Installation Options."

This screen enables you to select either the complete installation of Windows Server 2012 R2 or the Server Core option.

6. You are asked to accept the license terms. Select the check box labeled I accept the license terms and then click Next.

7. You receive the options shown in Figure 2-4 to upgrade or install a clean copy of Windows Server 2012 R2. Select Custom (advanced) to install a clean copy of Windows Server 2012 R2. The upgrade option is available only if you have started the installation from within Windows Server 2008, Windows Server 2008 R2, or the original version of Windows Server 2012.



Which type of installation do you want?

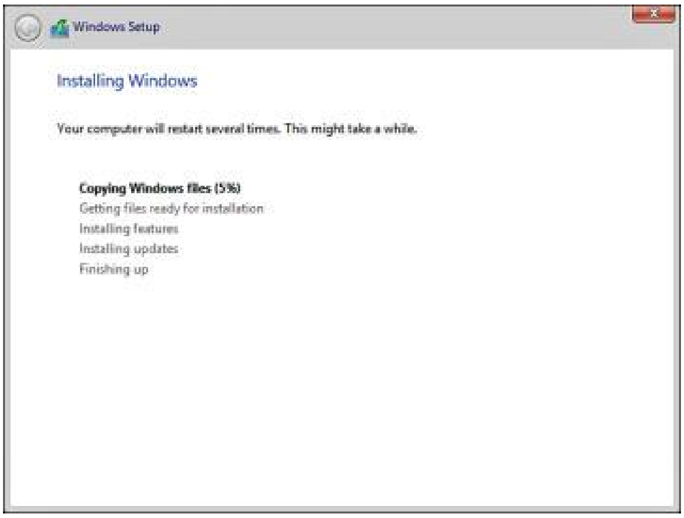
Upgrade: Install Windows and keep files, settings, and applications
The files, settings, and applications are moved to Windows with this option. This option is only available when a supported version of Windows is already running on the computer.

Custom: Install Windows only (advanced)
The files, settings, and applications aren't moved to Windows with this option. If you want to make changes to partitions and drives, start the computer using the installation disc. We recommend backing up your files before you continue.

Help me decide

8. Select the disk on which you want to install Windows and then click Next.

9. Take a coffee break while the installation proceeds. This takes some time (particularly when installing on a virtual machine), and the computer restarts several times. As shown in Figure 2-5, Setup charts the progress of installation.



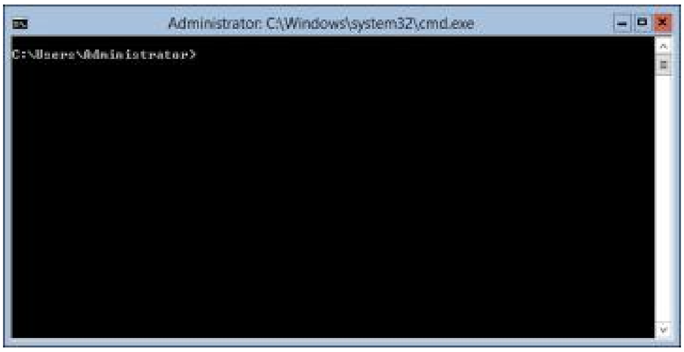
Installing Windows

Your computer will restart several times. This might take a while.

Copying Windows files (5%)
Getting files ready for installation
Installing features
Installing updates
Finishing up

10. After 15–30 minutes (depending on your hardware), Windows restarts a last time and informs you that your password must be changed before logging on for the first time. Click OK.

11. Type and confirm a strong password. When informed that the password is changed, click OK. After a minute or so, the desktop appears, containing a command window but no Start screen or desktop icons (see Figure 2-6). This is the standard Windows Server Core interface.



```
C:\Users\Administrator>
```

At startup, Windows Server Core 2012 R2 displays only a command window.

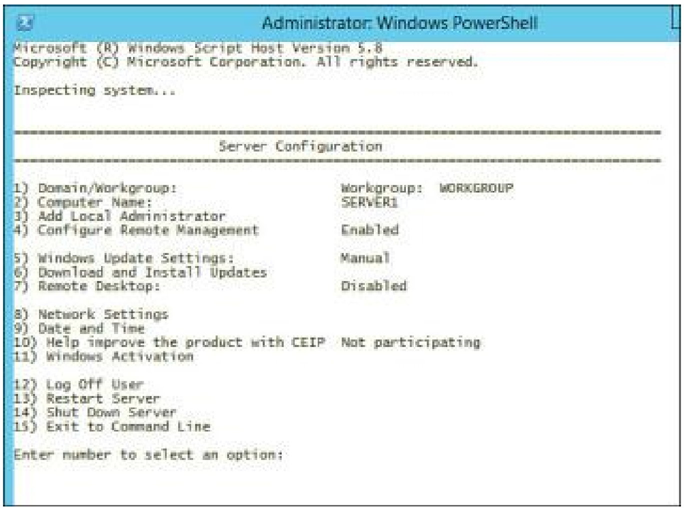
12. To set the correct time, type control timedate.cpl. By default, Server Core sets the time zone to Pacific Time. If you are in a different time zone, you will need to change this. Set the appropriate time zone, change the date and time if necessary, and then click OK.
13. Windows installs Server Core with a randomly generated computer name. To set a name of your choice, type netdom renamecomputer %computername% /newname:*ServerC1* (where, in this instance, *ServerC1* is the name you're assigning; substitute your desired server name).
14. Windows warns you that the rename process might have an adverse impact on some services. Type Y to proceed.
15. You are informed that the computer needs to be restarted in order to complete the rename. Type shutdown /r /t 0 to reboot your server.
16. After the server reboots, press Ctrl+Alt+Delete and log on using the password you set in step 11.

Useful Server Core Commands

All configuration, management, and troubleshooting of Windows Server Core is done from the command line. Available utilities enable you to perform almost all regular configuration tasks in this fashion. In Windows Server 2012 and 2012 R2, many of the commands used with Server Core in Windows Server 2008 have been replaced by PowerShell cmdlets. Table describes some of the more useful available commands.

Useful Windows Server Core Commands and Cmdlets

Command	Meaning
netdom join*computername*/domain:*domainname*	Joins an Active Directory domain. You will be prompted for the username and password of a user with domain administrator privileges.
Sconfig.cmd	Configures and manages a series of common Server Core installation properties..
cscript scregedit.wsf	Enables automatic updates.
Get-WindowsFeature	Displays roles and features currently installed on the server.
Install-WindowsFeature	Adds roles or features.
Uninstall-WindowsFeature	Removes roles or features.
netsh interface IPv4	Includes a series of subcommands that enable you to configure IPv4 networking.
netsh advfirewall	Includes subcommands that enable you to configure the Windows firewall.
Help	Provides a list of all available Windows Server Core commands.



```
Microsoft (R) Windows Script Host Version 5.8
Copyright (C) Microsoft Corporation. All rights reserved.

Inspecting system...

===============================================================================
                         Server Configuration
===============================================================================

1) Domain/Workgroup:                    Workgroup:  WORKGROUP
2) Computer Name:                       SERVER1
3) Add Local Administrator
4) Configure Remote Management          Enabled

5) Windows Update Settings:             Manual
6) Download and Install Updates
7) Remote Desktop:                      Disabled

8) Network Settings
9) Date and Time
10) Help improve the product with CEIP  Not participating
11) Windows Activation

12) Log Off User
13) Restart Server
14) Shut Down Server
15) Exit to Command Line

Enter number to select an option:
```

The Sconfig.cmd utility enables you to perform many basic configuration actions on a Server Core computer.

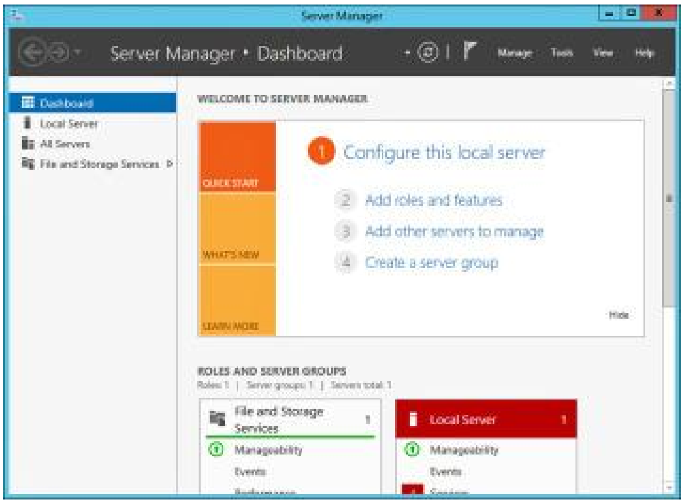
Installing the Full GUI Server

Although Microsoft markets Server Core as being the default Windows Server 2012 R2 installation, the full GUI version still represents the most easily managed version of the server. The procedure for installing the full GUI server is the same whether you're installing directly from a DVD-ROM or a network share, except that you must have some type of network client installed on your computer to access a network share. The following procedure outlines installation from a DVD-ROM:

1. Follow the procedure outlined earlier for installing Windows Server Core
2. Select either Windows Server 2012 R2 Standard (Server with a GUI) or Windows Server 2012 R2 Datacenter (Server with a GUI), and then click Next.
3. Complete steps 6–10 of the earlier procedure. Installation will take 15–45 minutes, depending on your hardware.
4. Type and confirm a secure password. Windows informs you that your password has been changed. Click OK.
5. Windows displays a Welcome message and prepares your desktop. Then the desktop with Server Manager shown in Figure appears.

6. After you have performed the initial configuration steps, you will be prompted to press Ctrl+Alt+Deleteand enter your password when you restart your server.

7. When you shut down a Windows Server 2012 R2 computer, it displays the Shutdown Event Tracker dialog box, which asks you for a reason for shutting down the server. For learning purposes, it is helpful to disable this item. You can do so by typing gpedit.msc to open the Local Group Policy Object Editor. Navigate to Computer Configuration\Administrative Templates\System, right-click the Display Shutdown Event Tracker policy, and click Properties. On the dialog box that appears, click Disabled and then click OK.



When you log on to Server with a GUI for the first time, Server Manager appears.

Converting Between Core and GUI

You can convert the full GUI version of Windows Server 2012 R2 to Server Core or vice versa at any time;

Use the following procedure to convert the full GUI server to Server Core:

1. From the Search charm, type powershell and select Windows PowerShell.
2. Type the following cmdlet:

```
Uninstall-WindowsFeature Server-Gui-Mgmt-Infra –Restart
```

3. A display on the PowerShell window tracks the removal process, which takes several minutes. A Configuring Windows features message then appears as the GUI is removed. After a few more minutes, the server restarts. When the logon screen appears, pressCtrl+Alt+Delete and log back on as an administrator.

Use the following procedure to convert a Server Core computer to run the full GUI:

1. From the administrative command prompt, type powershell.
2. Type the following command:

```
Get-WindowsImage -ImagePath <path to wim>\install.wim
```

3. This command returns the index number for the server with a GUI image. Then type the following command:

```
4.      Install-WindowsFeature Server-Gui-Mgmt-Infra, Server-Gui-Shell –
```

```
Restart –Source wim:<path to wim>\install.wim:<Index #>
```

5. Wait as Windows configures features and restarts; then press Ctrl+Alt+Delete and log back on as an administrator.

Result:
Windows Server Installed Successfully.