HTTP://WWW.MYWEBUNIVERSITY.COM

THE DIRECT PATH TO

LINUX

UBUNTU

This book **"The Direct Path to Linux Ubuntu"** helps you learn the Linux Operating System using Linux Ubuntu. All you need is access to the internet and a Web browser using any device you prefer to use. You do not need to install the Linux Ubuntu Operating System on your particular device. Please simply visit the **MyWebUniversity.com** and enjoy learning it online at your own pace.

WAHID LUTFY AND HAMID LUTFY

This book is dedicated to my late beloved brother **Dr. Amanullah Lutfy** (formally Cardiologist at Kaiser Permanente), who was a role model for myself, my siblings, his beloved children and everyone else whom he touched throughout his life. We all are extremely thankful to Almighty God for having such a Nobel great brother. It is true at first glance losing such a beloved, educated, and compassionate man like yourself leaves a gaping hole in our lives and the world; luckily, that hole can be filled with all of the people you helped. You always inspired us and shined like the sun. You are not gone and the generation yet to come will benefit from your presence here on earth!

Dr. Aman Ullah Lutfy, MD

I became a doctor because I knew I could become a dedicated, patient, skilled clinician. I would feel honored to help you when you are in need of care.

The Direct Way to Linux Ubuntu

TABLE OF CONTENTS

- **Chapter 6**
- **UNIX and Linux Web Servers**

- **Chapter 7**
- **UNIX and Linux File Systems (UFS, ZFS, EXT2, EXT3, EXT4, XFS, BFS)**

The Direct Way to Linux Ubuntu

Preface

Agreement between you and us:

Please be advised that by providing this course "The Direct Way to Linux Ubuntu" I would like to provide you the opportunity to learn the "Linux Ubuntu" Operating systems in a practical manner. The material covered in this book may not be suitable for all condition, please read the manual pages in addition to the examples provided in this book so that you don't make any mistake inadvertently. You agree that you are fully responsible for any issues, problems, or damage as a result of any action taken by you. In other words, we are not taking any responsibility whatsoever for any inconvenience or damage that the material in this book may cause, please read it carefully, follow the manual pages and if you are in doubt of any command don't use it. Furthermore, You agree that you are not attempting to hack or do any damage to our websites.

If you have any suggestion, comments, or like to provide us any feedback please write me: Email US.

You can learn Linux using this class in two ways:

1. Use the web interface The Direct Way to Linux Ubuntu
2. Read the materials of this course or book and practice the examples on any computer that is running the Linux Ubuntu Operating systems.

Why this book?

The purpose of this book, "The Direct Way to Linux Ubuntu" is to provide you with a very simple method of learning Linux using command line examples. You will have hands on training while learning this book using the web interface. This book not only allows you to learn Linux by reading the materials covered in each chapter, but it will also allow you to have a web interface to the

commands listed in each chapter so that you can practice and memorize the commands as you learn them.

The Direct Way to Linux Ubuntu

Chapter 1

About UNIX Operating Systems

While there are many resources and information I could provide you related to the Linux Operating Systems, I would like to provide you a link from the bell-labs where the UNIX Operating systems was originally discovered in 1969. The AT&T bell laboratory later on become the Lucent Technology and merged with Alcatel as one company "Alcatel-Lucent". Here is The invention of UNIX for an overview of the UNIX* operating system. Please use this link to read about the history, the world's most important operating system invention, its success, features and ongoing research. In addition, you should check on UNIX multi-tasking, multi-processing, multi-threading, portability, libraries, developers' tools, and resources features and functionalities.

Notice: This page may redirect you to other websites for more information, please use your browser back button to return to this page.

Your Homework

Please use your favorite websites or search engines such as Google.com, Manual Pages @MyWebUniversity , or My book on Oracle Sun Solaris Express , or Oracle.com to search for the following strings and then read the one that is mostly beneficial to you.

- UNIX and LINUX
- UNIX and LINUX kernel
- UNIX and LINUX History

- UNIX and LINUX multitasking
- UNIX and LINUX multithreading
- UNIX and LINUX System administration
- UNIX and LINUX Solaris Sun Microsystems
- UNIX and LINUX rsh rcp rexec ftp telnet sshd
- UNIX and LINUX shells sh ksh bash csh tcsh zsh
- UNIX and LINUX ld as link make gcc mdb adb gdb
- UNIX and LINUX manual pages man whatis whereis locate
- UNIX and LINUX Web Servers Apache Netscape Tomcat Iplanet
- UNIX and LINUX C C++ Lisp Fortran Pascal Java Perl MySQL Oracle
- UNIX and LINUX programming languages applications libraries and tools
- My favorite UNIX Operating systems and Hardware company Sun Solaris by Sun Microsystems
- UNIX Solaris Mac OS X Linux FreeBSD BSD IRIX RedHat Fedora Cobalt Caldera AIX HP-UX Intergraph CLIX DEC OSF1 AT&T

You can also use MyWebUniversity.com or MyWebUniversity.com search engines to search for UNIX related websites

The Direct Way to Linux Ubuntu

Chapter 2

Basic UNIX and Linux Commands.

In this chapter you will learn about the following Linux Ubuntu commands that are listed in the table below. When you click on each command, you will be provided with three sections:

1. The **'whatis'** section which describes what this command does.
2. The **'man'** section which shows the manual pages for command.
3. The **'Example'** section which provides you with some of the examples.

banner	**cal**	**cat**	**clear**	**date**	**echo**	**head**	**hostname**	**host**	**id**
info	**paste**	**ps**	**pwd**	**uname**	**less**	**ls**	**man**	**mkdir**	**more**
pg	**ping**	**dig**	**nslookup**	**atrm**	**atq**	**sort**	**touch**	**w**	**who**

To better understand each command, you can also check some of the related documentation provided to you on the left side of this page under the "References" section by clicking on each of the hypertext. For instance, you can read the Linux Ubuntu documentation which is in both in "HTML, and PDF" format by clicking on the Official Ubuntu Documentation, or read the UNIX and Linux

manual pages for different flavors of UNIX and Linux at MyWebUniveristy.com by clicking at UNIX and Linux Manual Pages.

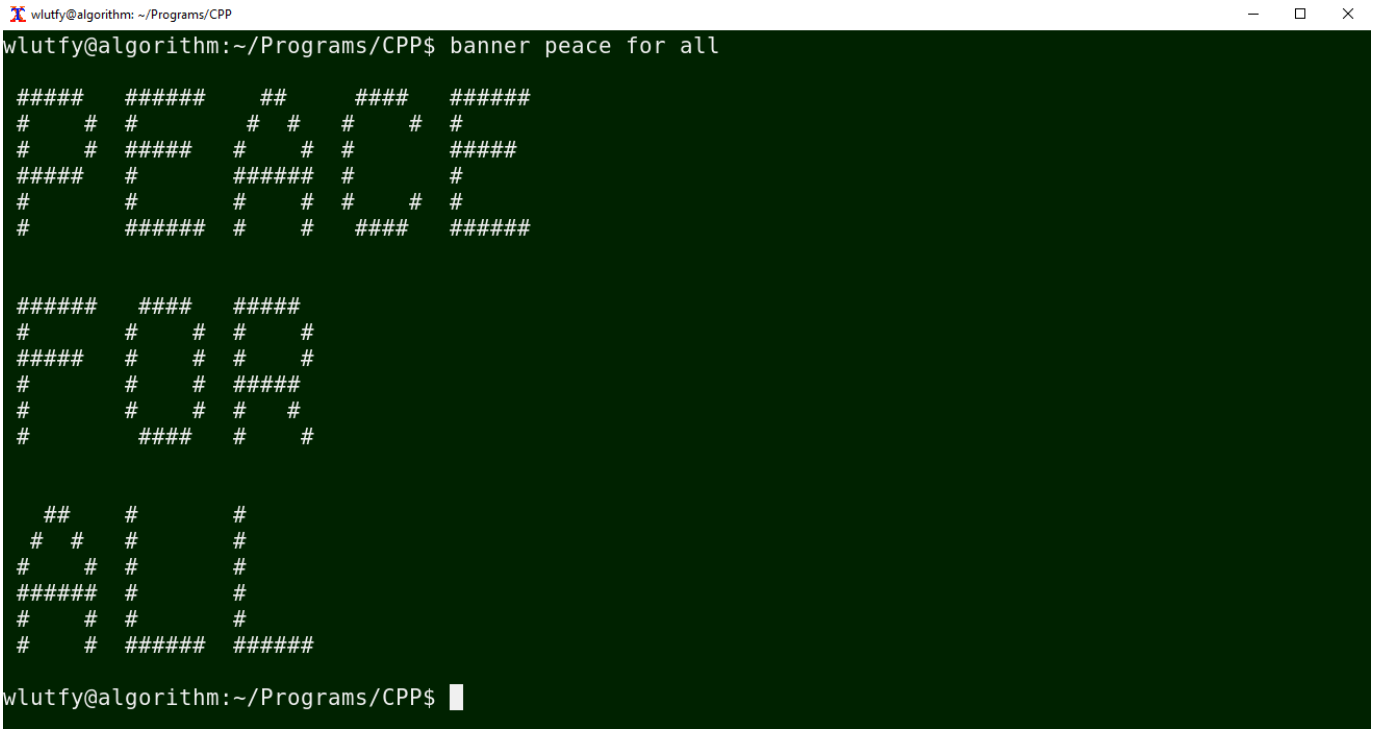
The Direct Way to Linux Ubuntu

$ whatis banner

banner banner (1) - make posters

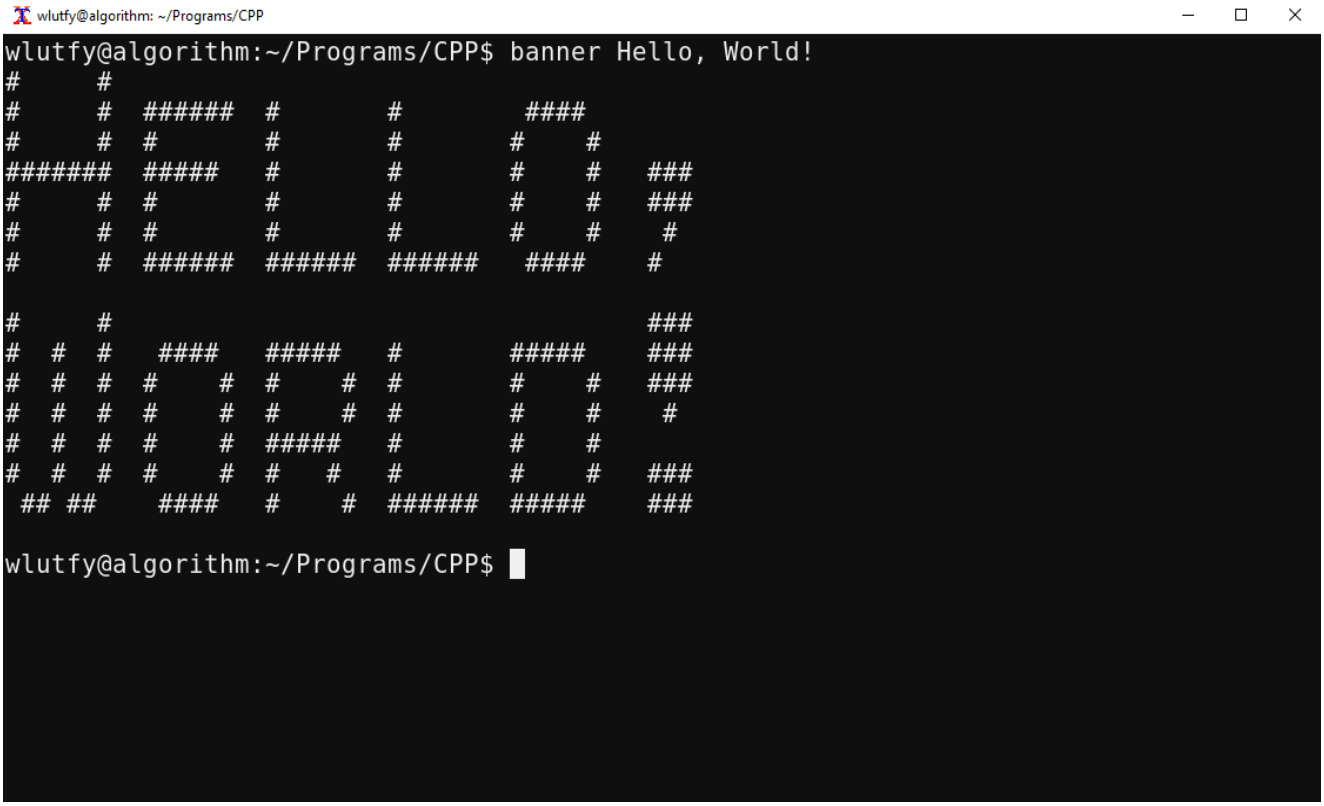
Examples: In the two examples below, you can see the command **"banner Peace For All!"** that displays the string '**Peace For All!**'.

Same command, but a different using a different **xterm.**

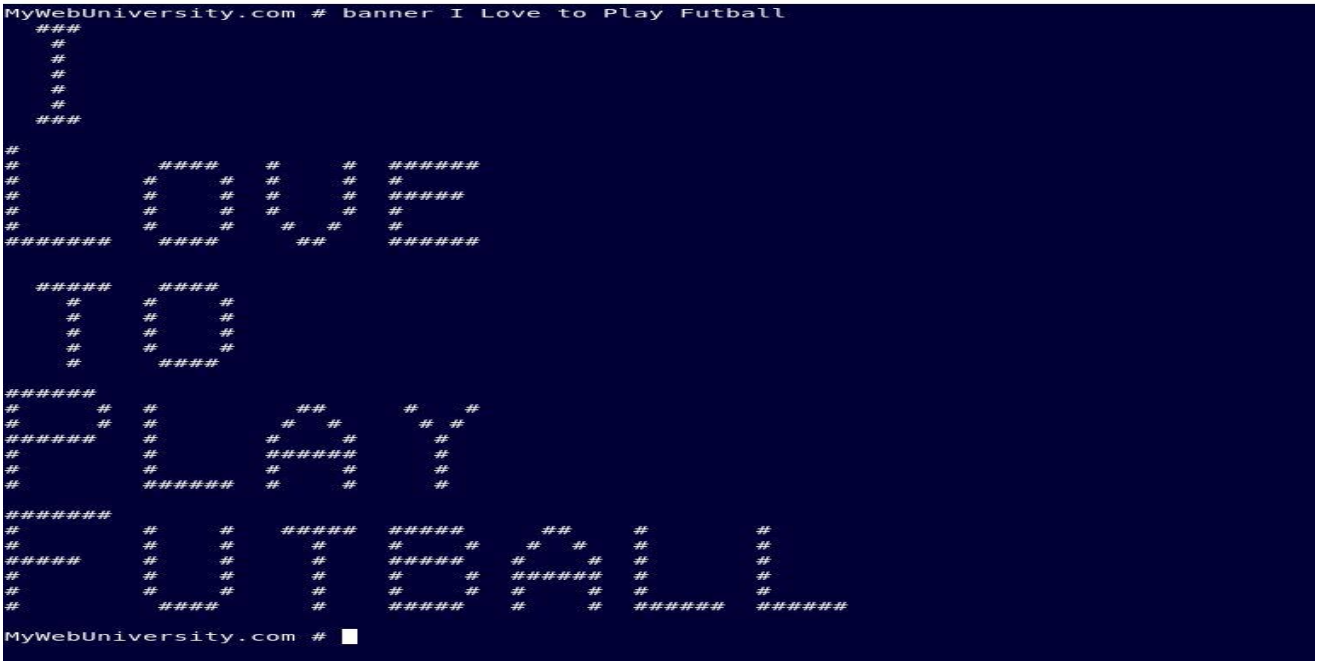


In the two examples below, you can see the command **"banner Hello, World!"** displays the string **'Hello, World!'.**

Same command, but a different using a different **xterm.**



In the example below, you can see the command "**banner I Love to Play Futball**" that displays the string '**I Love to Play Futball**'.



Do you prefer to use UNIX Operating System or Linux Operating System? I love both!

How about C++ Programming Language or Python Programming Language? I love both!

Since both C++, and Python supports for Functional well as Object Oriented Programming.

1. banner A
2. banner The Direct Way To Linux Ubuntu

Please click on man banner. to see the man pages for this command.

The Direct Way to Linux Ubuntu

$ whatis cal

cal cal (1) - display a calendar

Examples:

1. cal
2. cal 1931
3. cal 1970
4. cal 2 1998
5. cal 2011
6. cal 6 2001

Please click on man cal. to see the man pages for this command.

Examples: Here are some of the command usages for **'cal'** command:

The output of your command "cal" is:

The command **"cal"** without any arguments will print the current month calendar.

```
   November 2021
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
```

We can also get the same result if I type the command **"cal 11 2021"** in this case since November is the 11 month of the year 2021. The command below as actually run on the Linux server using the **secure shell "ssh"** connection on a terminal.

```
wlutfy@MyWebUniversity:~$ cal 11 2021
   November 2021
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
```

The output of your command "cal 1931" is:

The command **"cal 1931"** with 1931 as the argument will print the calendar for the entire year of 1931.

```
                              1931
      January               February                March
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
          1  2  3   1  2  3  4  5  6  7   1  2  3  4  5  6  7
 4  5  6  7  8  9 10   8  9 10 11 12 13 14   8  9 10 11 12 13 14
11 12 13 14 15 16 17  15 16 17 18 19 20 21  15 16 17 18 19 20 21
18 19 20 21 22 23 24  22 23 24 25 26 27 28  22 23 24 25 26 27 28
25 26 27 28 29 30 31                        29 30 31


       April                 May                   June
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
          1  2  3  4               1  2     1  2  3  4  5  6
 5  6  7  8  9 10 11   3  4  5  6  7  8  9   7  8  9 10 11 12 13
```

```
12 13 14 15 16 17 18  10 11 12 13 14 15 16  14 15 16 17 18 19 20
19 20 21 22 23 24 25  17 18 19 20 21 22 23  21 22 23 24 25 26 27
26 27 28 29 30        24 25 26 27 28 29 30  28 29 30
                      31

       July              August            September
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
        1  2  3  4              1        1  2  3  4  5
 5  6  7  8  9 10 11   2  3  4  5  6  7  8   6  7  8  9 10 11 12
12 13 14 15 16 17 18   9 10 11 12 13 14 15  13 14 15 16 17 18 19
19 20 21 22 23 24 25  16 17 18 19 20 21 22  20 21 22 23 24 25 26
26 27 28 29 30 31     23 24 25 26 27 28 29  27 28 29 30
                      30 31

      October           November           December
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
           1  2  3   1  2  3  4  5  6  7        1  2  3  4  5
 4  5  6  7  8  9 10   8  9 10 11 12 13 14   6  7  8  9 10 11 12
11 12 13 14 15 16 17  15 16 17 18 19 20 21  13 14 15 16 17 18 19
18 19 20 21 22 23 24  22 23 24 25 26 27 28  20 21 22 23 24 25 26
25 26 27 28 29 30 31  29 30                27 28 29 30 31
```

The output of your command "cal 1970" is:

Similarly, the command **"cal 1970"** with 1970 as the argument will print the calendar for the UNIX EPOCH calendar that started on 1/1/1970.

```
                      1970
      January           February          March
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
           1  2  3   1  2  3  4  5  6  7   1  2  3  4  5  6  7
 4  5  6  7  8  9 10   8  9 10 11 12 13 14   8  9 10 11 12 13 14
11 12 13 14 15 16 17  15 16 17 18 19 20 21  15 16 17 18 19 20 21
18 19 20 21 22 23 24  22 23 24 25 26 27 28  22 23 24 25 26 27 28
25 26 27 28 29 30 31                       29 30 31
```

```
      April               May                 June
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
        1  2  3  4                1  2     1  2  3  4  5  6
 5  6  7  8  9 10 11   3  4  5  6  7  8  9   7  8  9 10 11 12 13
12 13 14 15 16 17 18  10 11 12 13 14 15 16  14 15 16 17 18 19 20
19 20 21 22 23 24 25  17 18 19 20 21 22 23  21 22 23 24 25 26 27
26 27 28 29 30        24 25 26 27 28 29 30  28 29 30
                      31

       July              August              September
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
        1  2  3  4                   1        1  2  3  4  5
 5  6  7  8  9 10 11   2  3  4  5  6  7  8   6  7  8  9 10 11 12
12 13 14 15 16 17 18   9 10 11 12 13 14 15  13 14 15 16 17 18 19
19 20 21 22 23 24 25  16 17 18 19 20 21 22  20 21 22 23 24 25 26
26 27 28 29 30 31     23 24 25 26 27 28 29  27 28 29 30
                      30 31

      October             November            December
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
           1  2  3   1  2  3  4  5  6  7        1  2  3  4  5
 4  5  6  7  8  9 10   8  9 10 11 12 13 14   6  7  8  9 10 11 12
11 12 13 14 15 16 17  15 16 17 18 19 20 21  13 14 15 16 17 18 19
18 19 20 21 22 23 24  22 23 24 25 26 27 28  20 21 22 23 24 25 26
25 26 27 28 29 30 31  29 30                 27 28 29 30 31
```

The output of your command "cal 2011" is:

To get the entire calendar for 2011, you type "**cal 2011**".

```
                           2011
      January             February            March
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
                 1        1  2  3  4  5        1  2  3  4  5
 2  3  4  5  6  7  8   6  7  8  9 10 11 12   6  7  8  9 10 11 12
 9 10 11 12 13 14 15  13 14 15 16 17 18 19  13 14 15 16 17 18 19
16 17 18 19 20 21 22  20 21 22 23 24 25 26  20 21 22 23 24 25 26
```

```
23 24 25 26 27 28 29  27 28                27 28 29 30 31
30 31

      April                May                June
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
            1  2   1  2  3  4  5  6  7           1  2  3  4
 3  4  5  6  7  8  9   8  9 10 11 12 13 14   5  6  7  8  9 10 11
10 11 12 13 14 15 16  15 16 17 18 19 20 21  12 13 14 15 16 17 18
17 18 19 20 21 22 23  22 23 24 25 26 27 28  19 20 21 22 23 24 25
24 25 26 27 28 29 30  29 30 31              26 27 28 29 30


       July               August             September
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
            1  2      1  2  3  4  5  6           1  2  3
 3  4  5  6  7  8  9   7  8  9 10 11 12 13   4  5  6  7  8  9 10
10 11 12 13 14 15 16  14 15 16 17 18 19 20  11 12 13 14 15 16 17
17 18 19 20 21 22 23  21 22 23 24 25 26 27  18 19 20 21 22 23 24
24 25 26 27 28 29 30  28 29 30 31           25 26 27 28 29 30
31

      October            November            December
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
               1        1  2  3  4  5           1  2  3
 2  3  4  5  6  7  8   6  7  8  9 10 11 12   4  5  6  7  8  9 10
 9 10 11 12 13 14 15  13 14 15 16 17 18 19  11 12 13 14 15 16 17
16 17 18 19 20 21 22  20 21 22 23 24 25 26  18 19 20 21 22 23 24
23 24 25 26 27 28 29  27 28 29 30           25 26 27 28 29 30 31
30 31
```

The output of your command "cal 2 1998" is:

To get the second month (February) of the year 1998, we run the above command**.**

```
   February 1998
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

The output of your command "cal 6 2001" is:

Similarly, to get the sixth month (June) of the year 2001, we run the above command**.**

```
     June 2001
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
```

$ whatis cat

cat cat (1) - concatenate files and print on the standard output

cat cat (1) - concatenate and display files

Examples:

1. cat -n /etc/networks
2. cat -n /etc/os-release
3. cat /etc/networks
4. cat /etc/networks /etc/os-release
5. cat /etc/os-release
6. cat /etc/os-release /etc/networks

Please click on man cat. to see the man pages for this command.

Here are some of the above examples shown with their output below:

The below command is using the option **"-n"** to display the content of the file /etc/networks line by line number. Thus, the **"-n"** flag of the "cat" command will display the line numbers as well as each line.

The output of your command "cat -n /etc/networks" is:

```
1       # symbolic names for networks, see networks(5) for more information
2       link-local 169.254.0.0
```

The below command is using the option **“-n”** to display the content of the file /etc/os-release line by line number. Thus, the **“-n”** flag of the “cat” command will display the line numbers as well as each line. As you can see the /etc/os-release file shows the System Name, Version, Distribution, and many other related fields about the Operating System Release.

The output of your command "cat -n /etc/os-release" is:

```
    1	NAME="Ubuntu"
    2	VERSION="18.04.5 LTS (Bionic Beaver)"
    3	ID=ubuntu
    4	ID_LIKE=debian
    5	PRETTY_NAME="Ubuntu 18.04.5 LTS"
    6	VERSION_ID="18.04"
    7	HOME_URL="https://www.ubuntu.com/"
    8	SUPPORT_URL="https://help.ubuntu.com/"
    9	BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"
   10	PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-
policy"
   11	VERSION_CODENAME=bionic
   12	UBUNTU_CODENAME=bionic
```

In the example below we use the Linux concatenation command **“cat”** to see the content of the **/etc/networks** without line numbers.

The output of your command "cat /etc/networks" is:

```
# symbolic names for networks, see networks(5) for more information
link-local 169.254.0.0
```

The concatenation command has many different functionalities, you can use the command “cat” to display more than one file at the same time. In this example below, we are using the “cat” command to look at the contents of two files at the same time.

The output of your command "cat /etc/networks /etc/os-release" is:

```
# symbolic names for networks, see networks(5) for more information
link-local 169.254.0.0
NAME="Ubuntu"
VERSION="18.04.5 LTS (Bionic Beaver)"
ID=ubuntu
ID_LIKE=debian
PRETTY_NAME="Ubuntu 18.04.5 LTS"
VERSION_ID="18.04"
HOME_URL="https://www.ubuntu.com/"
SUPPORT_URL="https://help.ubuntu.com/"
```

BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"
PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"
VERSION_CODENAME=bionic
UBUNTU_CODENAME=bionic

Below, we are showing how to see the contents of the **/etc/os-release** file without line numbers as we did before with "=n" option.

The output of your command "cat /etc/os-release" is:

NAME="Ubuntu"
VERSION="18.04.5 LTS (Bionic Beaver)"
ID=ubuntu
ID_LIKE=debian
PRETTY_NAME="Ubuntu 18.04.5 LTS"
VERSION_ID="18.04"
HOME_URL="https://www.ubuntu.com/"
SUPPORT_URL="https://help.ubuntu.com/"
BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"
PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"
VERSION_CODENAME=bionic
UBUNTU_CODENAME=bionic

This time, we use the **"cat"** command to display the two files, but we changed the order so that we see the content of the file **/etc/os-release** before the **/etc/networks** file as shown below:

The output of your command "cat /etc/os-release /etc/networks" is:

NAME="Ubuntu"
VERSION="18.04.5 LTS (Bionic Beaver)"
ID=ubuntu
ID_LIKE=debian
PRETTY_NAME="Ubuntu 18.04.5 LTS"
VERSION_ID="18.04"
HOME_URL="https://www.ubuntu.com/"
SUPPORT_URL="https://help.ubuntu.com/"
BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"
PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"
VERSION_CODENAME=bionic
UBUNTU_CODENAME=bionic
# Symbolic names for networks, see networks(5) for more information
link-local 169.254.0.0

For each command I also provide the Linux Manual Pages so that you can check further options online using http://www.MyWebUniversity.com Website which is also available for multiple Operating Systems including Mac OS X, Linux Ubuntu, Oracle Sun Solaris 11 (Sun Solaris Express, Open Solaris) and even Windows Help, and Windows PowerShell Cmdlets by visiting http://www.mywebuniversity.com/Man_Pages/

For instance, here is the Linux manual pages for the command **'cat',** you can also login to a UNI**X, Mac OS X**, or **Linux server** or **VM** and get the manual pages by simply typing the command **"man command-name**" where the **command-name** is any **UNIX/Linux** commands.

Here are some more examples of the command **"cat**" using the input file name "**cat_examples.txt**" that has some control characters such as space ' ', or five spaces ' ', tabs that are shown with **'^I'** and end of line as **'$**' symbol or new line character '**\n**' if you are using the **octal dump "od"** command shown below: In the **"od -c" octal dump "od"** with **'-c"** for character shows the tabs are as **'\t**', new lines as **'\n'** and spaces are as empty characters.

$ **od -c cat_examples.txt**
0000000 T h i s i s t h e f i r s
0000020 t l i n e o f t h i s f
0000040 i l e . **n** **n** **n** T h i s i s t
0000060 h e f o u r t h l i n e .
0000100 T h a t h a s **t** o n e t a b
0000120 h e r e , a n d t w o **t** **t**
0000140 t a b s h e r e . **n** T h i s
0000160 t h e f i f t h l i n e t
0000200 h a t h a s f i v e
0000220 s p a c e s a n d **t** **t** **t** **t** h
0000240 r e e t a b s . **n**
0000252

To see the content of the file "cat_examples.txt" we use the concatenation command **"cat"** as shown below:

$ **cat cat_examples.txt**
This is the first line of this file.

This is the fourth line. That has	one tab here, and two	tabs here.
This the fifth line that has	five spaces and		three tabs.

To see the content of the file **"cat_examples.txt"** with line numbers we use the concatenation command **"cat -n"** as shown below: In this case the flag **"-n"** will display the line numbers.

$ cat -n cat_examples.txt

```
1  This is the first line of this file.
2
3
4  This is the fourth line. That has	one tab here, and two	tabs here.
5  This the fifth line that has	five spaces and		three tabs.
```

To see the content of the file **"cat_examples.txt"** and display the number nonempty output lines, which overrides the -n flag, we use the **"-b"** flag of the **"cat"** command as shown below:

$ cat -b cat_examples.txt

```
1  This is the first line of this file.



2  This is the fourth line. That has	one tab here, and two	tabs here.
3  This the fifth line that has	five spaces and		three tabs.
```

The **"-s"** flag or option of the **"cat"** command will suppress repeated empty output lines as shown below:

$ cat -s cat_examples.txt

```
This is the first line of this file.

This is the fourth line. That has	one tab here, and two	tabs here.
This the fifth line that has	five spaces and		three tabs.
```

To suppress the repeated empty output lines first and then line number them we use the **"-sn"** flags.

$ cat -sn cat_examples.txt

```
1  This is the first line of this file.
2
3  This is the fourth line. That has	one tab here, and two	tabs here.
4  This the fifth line that has	five spaces and		three tabs.
```

The **"-A"** flag is used to show-all non-visible characters as well as printable characters that is equivalent to **"-vET".**

$ cat -A cat_examples.txt

This is the first line of this file.$
$
$
This is the fourth line. That has^Ione tab here, and two^I^Itabs here.$
This the fifth line that has five spaces and^I^I^Ithree tabs.$

The **"-E"** flag is used to show End of line characters with **'$'** symbol at the end of each line.

$ cat -E cat_examples.txt
This is the first line of this file.$
$
$
This is the fourth line. That has one tab here, and two tabs here.$
This the fifth line that has five spaces and three tabs.$

The combination of the **'-EA"** is not necessarily since the **'-A'** shows all non-visible as well as printable characters including tabs, new lines and any other control characters.

$ cat -EA cat_examples.txt
This is the first line of this file.$
$
$
This is the fourth line. That has^Ione tab here, and two^I^Itabs here.$
This the fifth line that has five spaces and^I^I^Ithree tabs.$
wlutfy@algorithm:~/Programs/Shells$ cat -s cat_examples.txt
This is the first line of this file.

This is the fourth line. That has one tab here, and two tabs here.
This the fifth line that has five spaces and three tabs.

The **"-sn"** will suppresses the two empty lines into one using the "-s" flag and the **"-n**" flag write line numbers for each of the lines afterward.

$ cat -sn cat_examples.txt
1 This is the first line of this file.
2
3 This is the fourth line. That has one tab here, and two tabs here.
4 This the fifth line that has five spaces and three tabs.

The below combination of the **"-sb"** with the flag **"-s"** which will suppress the two empty lines into one and the **"-b"** like the **"-n"** will add the line number for each non-empty lines in the final output.

$ cat -sb cat_examples.txt
1 This is the first line of this file.

```
2  This is the fourth line. That has       one tab here, and two          tabs here.
3  This the fifth line that has     five spaces and
```

The output of your command "man cat" is:

```
CAT(1)                    User Commands                    CAT(1)

NAME
       cat - concatenate files and print on the standard output

SYNOPSIS
       cat [OPTION]... [FILE]...

DESCRIPTION
       Concatenate FILE(s) to standard output.

       With no FILE, or when FILE is -, read standard input.

       -A, --show-all
              equivalent to -vET

       -b, --number-nonblank
              number nonempty output lines, overrides -n

       -e     equivalent to -vE

       -E, --show-ends
              display $ at end of each line

       -n, --number
              number all output lines

       -s, --squeeze-blank
              suppress repeated empty output lines

       -t     equivalent to -vT

       -T, --show-tabs
              display TAB characters as ^I

       -u     (ignored)

       -v, --show-nonprinting
              use ^ and M- notation, except for LFD and TAB
```

--help display this help and exit

--version
output version information and exit

EXAMPLES

cat f - g
Output f's contents, then standard input, then g's contents.

cat Copy standard input to standard output.

AUTHOR

Written by Torbjorn Granlund and Richard M. Stallman.

REPORTING BUGS

GNU coreutils online help:
Report cat translation bugs to

COPYRIGHT

Copyright (C) 2017 Free Software Foundation, Inc. License GPLv3+: GNU GPL version 3 or later .
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

SEE ALSO

tac(1)

Full documentation at:
or available locally via: info '(coreutils) cat invocation'

GNU coreutils 8.28 January 2018 CAT(1)

In the below, I am using the 'cat' command to execute several commands using the **insertion "<<"** operator with combination of the **"DOC"** Document script. The string "**DOC**" is your choice, but whatever you string started with you have to use the same string to end your documentation as shown below:

```
wlutfy@MyWebUniversity::~/Programs/Shells$ cat << DOC
> $(echo "Wahid is using the cat command to run these commands.")
> $(date)
> `pwd`
> `who`
> $(ls -ld *.sh)
> $(find . -name "script1.sh" -exec ls -ld {} \;)
```

> `cat -n script1.sh`
> DOC

Once, I type the string “DOC”, the documentation of the script completes, and the output of the commands are shown below provided that all the commands were valid and had the correct syntax.

Wahid is using the cat command to run these commands.
Sat Nov 13 12:19:36 PST 2021
/home/wlutfy/Programs/Shells
wlutfy :0 2021-10-27 00:21 (:0)
wlutfy pts/1 2021-11-13 10:50 (192.168.254.16)
-rwxr-xr-x 1 wlutfy wlutfy 658 Nov 6 13:50 mydg.sh
-rwxr-x--- 1 wlutfy wlutfy 225 Nov 13 12:11 script1.sh
-rwxr-x--- 1 wlutfy wlutfy 225 Nov 13 12:11 ./script1.sh
1 /home/wlutfy/Programs/Shells
2 -rwxr-xr-x 1 wlutfy wlutfy 658 Nov 6 13:50 ./mydg.sh
3 -rw-rw-r-- 1 wlutfy wlutfy 151 Nov 13 12:10 ./script1.sh
4 Sat Nov 13 12:11:46 PST 2021
5 -rw-rw-r-- 1 wlutfy wlutfy 151 Nov 13 12:10 script1.sh
wlutfy@MyWebUniversity::~/Programs/Shells$

In the example below, I am using the standard input ‘-‘ as the option to the ‘cat’ command in order to use the **‘cat’** command to echo every character, word, or sentence that I write. Once I press enter the character, word, or sentence is repeated as shown below. This conversation ends when I press the **“control”** and the letter **‘d’** together at the same time.

wlutfy@MyWebUniversity::~$ **cat -**
The cat command with the standard input using the symbol '-' in this case works like echo command.
The cat command with the standard input using the symbol '-' in this case works like echo command.
What I type will be echoed until I press control and d to end my interaction.
What I type will be echoed until I press control and d to end my interaction.
MyWebUniversity.com
MyWebUniversity.com
Welcome to OurUnix.com and MyWebUniversity.com and MyWebTut.com
Welcome to OurUnix.com and MyWebUniversity.com and MyWebTut.com
You will find more than hundred thousand of UNIX, LINUX, MAC OS X, Windows, and Windows PowerShell Manual Pages.
You will find more than hundred thousand of UNIX, LINUX, MAC OS X, Windows, and Windows PowerShell Manual Pages.
In addition, at both MyWebUniversity.com and OurUNIX.com you can learn the Linux and Oracle Solaris Operating Systems.
In addition,
at both MyWebUniversity.com and OurUNIX.com you can learn the Linux and Oracle Solaris Operating Systems.
All you need is access to the internet and a browser of your choice.

All you need is access to the internet and a browser of your choice.
Best of Luck
Best of Luck
Love and Peace for All!!!
Love and Peace for All!!!
wlutfy@MyWebUniversity::~$

Here are the examples of the concatenation **'cat'** command using the **standard input** "-", **standard out**, and **redirection ">"** to a new file method of using the **'cat'** command. The two **"<< "** symbol in this case is used as **an insertion operator**, the **"DOC"** can be any string, but once you choose that string, you have to complete your inputs by putting the same string at the end, in this case the **"DOC"** string. The greater than **">"** symbol is used for redirecting what is inside the standard input from first line to the last line and sending that in the output file **myscript.sh**. Be careful with using the **">"** redirection, if the file already existed, you will over-write the existing file. If that is not the intention, please backup the file or provide a new name or use the double greater than sign **">>"** if you like to append the file instead of overwriting it with the ">" symbol.

wlutfy@MyWebUniversity::~$ **cat << DOC > myscript.sh**
```
#/bin/bash
# This script is written with the concatenation 'cat' command.
set -x
hostname
uname -a
echo -e "The content of this script is shown with line number."
cat -n myscript.sh
chmod 750 myscript.sh
```
wlutfy@MyWebUniversity::~$

Running this "myscript.sh" which was created by the **"cat"** command as shown above will now run each command listed in this script with **"++command-name",** followed by the output of that command**.** For example**,** the **"++hostname**" is the result of executing the command **"hostname**" and the output **"MyWebUniversity.com"** is the result of that command executing. This is as a result of enabling the execution of the command from the **"set -x**" in the third line.

wlutfy@MyWebUniversity::~$ **./myscript.sh**
```
++ hostname
MyWebUniversity
++ uname -a
Linux MyWebUniversity 5.4.0-67-generic #75~18.04.1-Ubuntu SMP Tue Feb 23 19:17:50 UTC 2021
x86_64 x86_64 x86_64 GNU/Linux
++ echo -e 'The content of this script is shown with line number.'
The content of this script is shown with line number.
++ cat -n myscript.sh
   1  #/bin/bash
   2  # This script is written with the concatenation 'cat' command.
```

```
   3  set -x
   4  hostname
   5  uname -a
   6  echo -e "The content of this script is shown with line number."
   7  cat -n myscript.sh
   8  chmod 750 myscript.sh
++ chmod 750 myscript.sh
```

Now we can see the script is created with proper permission, ready to be executed any time we like.

wlutfy@MyWebUniversity::~$ **ls -ld myscript.sh**
-rwxr-x--- 1 wlutfy wlutfy 204 Nov 13 14:17 myscript.sh

In the above Linux Manual Pages for **'man cat',** we observed that under the **"SEE ALSO"** section it references the command "**tac(1)".** As we can see from the name of the command, **"tac(1)"** is fount in section 1 of the Linux command as "tac" by simply running **"man tac",** or **"man -s 1 tac"**. In this example, the option **'-s 1 '** refers to section one of the **Linux Manual Pages**.

As the name for the command **"tac"** is the reverse of the command **"cat"**, the command "tac" will print the content of the input file in **reverse**, that is the last line first and the first line last. Here are a few examples:

Example 1: When a **file1.txt** only contains two lines.

Using the **"cat file1.txt"** command will display both lines in the order they are written in the file.

$ cat file1.txt

1: This is first line.

2: This is the second line.

Now, we are using the **"tac file1.txt"** command that displays both lines in reverse order of the way they are written in the file. In this example we can see the last line which line number 2 is written before the first line which is line number 1.

$ tac file1.txt

2: This is the second line.

1: This is first line.

Example 2: let's use the second file **"file2.txt"** that contains five lines.

Using the **"cat file2.txt"** command will display all five lines in the order they are written in the file.

$ cat file2.txt

1: This is first line.

2: This is the second line.

3: This is the third line.

4: This is the fourth line.

5: This is the fifth line.

Now, we are using the **"tac file2.txt"** command that displays all five lines in reverse order of the way they are written in the file. In this example we can see the last line which line number five is written before the fourth line which is written before the third line which is written before the third line which is written before the second line and finally the first line.

$ tac file2.txt

5: This is the fifth line.

4: This is the fourth line.

3: This is the third line.

2: This is the second line.

1: This is first line.

$ whatis cat

rev (1) - reverse lines characterwise

We also like to show you the **"rev"** command that copies the specified files to standard output, and then reverses the order of all characters in every line. The standard input is read, whenever no files are specified as input file.

Here are some examples of the usage for the **"rev"** command.

To display the content of the input file **mywebuni.txt**, we use the **"cat"** command below:

$ **cat mywebuni.txt**

Welcome To MyWebUniversity.com

To show each character from the file **mywebuni.txt** to standard output in reverse order.

$ **rev mywebuni.txt**

moc.ytisrevinUbeWyM oT emocleW

To read from standard input.

$ **rev**

Welcome To OurUNIX.com

moc.XINUruO oT emocleW

Mom

moM

Dad

daD

We ended the conversation by pressing "control" and 'd' keystrokes at the same time.

$

The Direct Way to Linux Ubuntu

$ whatis clear

clear clear (1) - clear the terminal screen

The following commands shows how to clear your screen on a Unix-based, Mac-based, or Linux-based Operating system in a terminal or console access.

Examples:

$ clear

$ tput clear

You can also use the "**tput clear"** command to clear your screen. This is similar to the **'cls'** command in the Windows platform. If you like to use the 'cls' command instead, you can create an alias as shown below:

$alias cls='clear'
$ alias cls
alias cls='clear'

$ **cls**

To define the alias 'cls' and assign the command 'clear' value to it. You can add your alias to your .bashrc or .cshrc file depending on which default SHELL you are using .bashrc for /bin/bash and .cshrc for /bin/csh, and /bin/tcsh. These /bin/bash shell is Born Again Shell which is the new version of the /bin/sh that was the original Born Shell. The /bin/csh is the C style syntax Shell, and the /bin/tcsh which stands for True C Shell is the newer version of the /bin/csh. There are other SHELLs /bin/ksh Korn Shell that has a syntax similar to BASH and /bin/zsh that has the syntax rich features of /bin/bash, /bin/ksh, and /bin/tcsh are all incorporated into the /bin/zsh.

For now, I like to introduce you with the below two commands output and the rest of the discussion will be covered in future chapters.

$ **echo $SHELL**
/bin/bash

$ cat /etc/shells
/etc/shells: valid login shells
/bin/sh
/bin/dash
/bin/bash
/bin/rbash
/usr/bin/tmux
/usr/bin/screen

Please click on man clear. to see the man pages for this command.

$ whatis date

date date (1) - print or set the system date and time

1. date
2. date '+%m/%d/%Y'

When I click on the first command **'date'** above, it will be executed on the Linux Server at MyWebUniversity.com and the output will be displayed as shown below:

The output of your command "date" is:

Sun Nov 14 03:24:44 EST 2021

Similarly, when I click on the second command **"'date '+%m%d%Y'"** above, it will be executed on the Linux Server at MyWebUniversity.com and the output will be displayed as shown below:

The output of your command "date '+%m/%d/%Y'" is:

Below are more examples of the command "date" usage. However, in order to synchronize the date and time of your system, you should consider using the Network Time Protocol daemon (ntpd) which is an **operating system program** that maintains the system time in synchronization with time servers using the Network Time Protocol (NTP). Below, I checked on my Linux Dedicated server, and observed that I was not running the "**ntp"** daemon. Therefore, I used the below commands as root to install the "ntp" daemon and synchronize my system date as shown below: In order my system to start the **"ntp"** daemon after a system reboot, I used the **"systemctl enable ntp".**

root@MyWebUniversity:/lib/systemd/system# **systemctl status ntp**
Unit ntp.service could not be found.
root@MyWebUniversity:/lib/systemd/system# **ntpq -p**

Command 'ntpq' not found, but can be installed with:

apt install ntp
apt install ntpsec

root@MyWebUniversity:/lib/systemd/system# **apt install ntp**
After this operation, 2,393 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://mirror.us.oneandone.net/ubuntu/ubuntu bionic/universe amd64 libopts25 amd64 1:5.18.12-4 [58.2 kB]
Get:2 http://mirror.us.oneandone.net/ubuntu/ubuntu bionic-updates/universe amd64 ntp amd64 1:4.2.8p10+dfsg-5ubuntu7.3 [640 kB]
Get:3 http://mirror.us.oneandone.net/ubuntu/ubuntu bionic-updates/universe amd64 sntp amd64 1:4.2.8p10+dfsg-5ubuntu7.3 [86.5 kB]
Fetched 785 kB in 0s (6,337 kB/s)
Selecting previously unselected package libopts25:amd64.
(Reading database ... 103419 files and directories currently installed.)
Preparing to unpack .../libopts25_1%3a5.18.12-4_amd64.deb ...
Unpacking libopts25:amd64 (1:5.18.12-4) ...
Selecting previously unselected package ntp.
Preparing to unpack .../ntp_1%3a4.2.8p10+dfsg-5ubuntu7.3_amd64.deb ...
Unpacking ntp (1:4.2.8p10+dfsg-5ubuntu7.3) ...
Selecting previously unselected package sntp.
Preparing to unpack .../sntp_1%3a4.2.8p10+dfsg-5ubuntu7.3_amd64.deb ...
Unpacking sntp (1:4.2.8p10+dfsg-5ubuntu7.3) ...
Setting up libopts25:amd64 (1:5.18.12-4) ...
Setting up sntp (1:4.2.8p10+dfsg-5ubuntu7.3) ...
Setting up ntp (1:4.2.8p10+dfsg-5ubuntu7.3) ...
Created symlink /etc/systemd/system/network-pre.target.wants/ntp-systemd-netif.path → /lib/systemd/system/ntp-systemd-netif.path.
Created symlink /etc/systemd/system/multi-user.target.wants/ntp.service → /lib/systemd/system/ntp.service.
ntp-systemd-netif.service is a disabled or a static unit, not starting it.
Processing triggers for systemd (237-3ubuntu10.44) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for ureadahead (0.100.0-21) ...

```
ureadahead will be reprofiled on next reboot
Processing triggers for libc-bin (2.27-3ubuntu1.4) ...

root@MyWebUniversity:/lib/systemd/system# systemctl status ntp
● ntp.service - Network Time Service
   Loaded: loaded (/lib/systemd/system/ntp.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2021-11-14 03:35:40 EST; 17s ago
     Docs: man:ntpd(8)
 Main PID: 16709 (ntpd)
    Tasks: 2 (limit: 4915)
   CGroup: /system.slice/ntp.service
           └─16709 /usr/sbin/ntpd -p /var/run/ntpd.pid -g -u 113:124

Nov 14 03:35:45 MyWebUniversity ntpd[16709]: Soliciting pool server 209.50.63.74
Nov 14 03:35:45 MyWebUniversity ntpd[16709]: Soliciting pool server 209.51.161.238
Nov 14 03:35:45 MyWebUniversity ntpd[16709]: Soliciting pool server 69.28.91.73
Nov 14 03:35:45 MyWebUniversity ntpd[16709]: Soliciting pool server 91.189.91.157
Nov 14 03:35:46 MyWebUniversity ntpd[16709]: Soliciting pool server 91.189.89.198
Nov 14 03:35:46 MyWebUniversity ntpd[16709]: Soliciting pool server 193.187.181.6
Nov 14 03:35:46 MyWebUniversity ntpd[16709]: Soliciting pool server 207.192.69.118
Nov 14 03:35:47 MyWebUniversity ntpd[16709]: Soliciting pool server 91.189.94.4
Nov 14 03:35:47 MyWebUniversity ntpd[16709]: Soliciting pool server 52.6.160.3
Nov 14 03:35:47 MyWebUniversity ntpd[16709]: Soliciting pool server 2600:3c03:e002:1300::10

root@MyWebUniversity:/lib/systemd/system# ps -efa | grep -i ntp
ntp      16709     1  0 03:35 ?        00:00:00 /usr/sbin/ntpd -p /var/run/ntpd.pid -g -u 113:124
root     18058 15531  0 03:36 pts/0    00:00:00 grep --color=auto -i ntp

root@MyWebUniversity:/lib/systemd/system# ntpq -p
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
 0.ubuntu.pool.n .POOL.          16 p    -   64    0    0.000    0.000   0.000
 1.ubuntu.pool.n .POOL.          16 p    -   64    0    0.000    0.000   0.000
 2.ubuntu.pool.n .POOL.          16 p    -   64    0    0.000    0.000   0.000
 3.ubuntu.pool.n .POOL.          16 p    -   64    0    0.000    0.000   0.000
 ntp.ubuntu.com  .POOL.          16 p    -   64    0    0.000    0.000   0.000
-ntp1.ny1.ap.fou 239.9.71.195     2 u   23   64    1   32.609   -0.686   0.879
-50-203-248-23-s 130.207.244.240  2 u   22   64    1   48.964   -1.085   1.151
-time.cloudflare 10.125.8.4       3 u   19   64    1    0.885    1.751   0.710
+lithium.constan 192.5.41.40      2 u   22   64    1   39.200    0.850   1.148
+ntp3.junkemailf 216.218.254.202  2 u   22   64    1   37.047    0.506   1.841
-mail.trafficsys 132.163.97.2     2 u   21   64    1   22.300   -4.345   0.985
#144.34.193.110. 66.220.9.122     2 u   17   64    1   41.783    5.025   0.727
*B1-66ER.matrix. 129.6.15.30      2 u   17   64    1   37.628    0.625   0.522
-atlantic-net1.z 139.78.97.128    2 u   20   64    1   12.591   -2.352   0.845
+clock.nyc.he.ne 127.67.113.92    2 u   20   64    1   30.629    0.583   0.850
-karhu.miuku.net 207.197.87.124   4 u   18   64    1   35.460    0.130   0.646
 alphyn.canonica 132.163.96.1     2 u   26   64    1   32.989   -1.047   0.000
```

```
#207.192.69.118  194.0.5.123      3 u   16   64    1   33.078  -81.815  28.428
#193.187.181.6   192.12.19.20     2 u   15   64    1   48.874    3.357   0.371
 chilipepper.can 145.238.203.14   2 u   25   64    1   95.435   -0.636   0.000
#strongbad.voice 130.207.244.240  2 u   14   64    1   34.329    1.381   0.325
 pugot.canonical 193.6.176.59     2 u   28   64    1  100.544   -0.130   0.000
```

root@MyWebUniversity:/lib/systemd/system# **date**
Sun Nov 14 03:36:22 EST 2021
root@MyWebUniversity:/lib/systemd/system#

Using the below "**systemctl enable ntp"** command and checking the status, we confirm that **"ntp"** daemon is enabled to start after a system reboot.

root@MyWebUniversity:/lib/systemd/system# **systemctl enable ntp**
Synchronizing state of ntp.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable ntp

root@MyWebUniversity:/lib/systemd/system# **systemctl status ntp**
● ntp.service - Network Time Service
  Loaded: loaded (/lib/systemd/system/ntp.service; enabled; vendor preset: **enabled)**
  Active: active (running) since Sun 2021-11-14 03:35:40 EST; 12min ago
    Docs: man:ntpd(8)
 Main PID: 16709 (ntpd)
   Tasks: 2 (limit: 4915)
  CGroup: /system.slice/ntp.service
          └─16709 /usr/sbin/ntpd -p /var/run/ntpd.pid -g -u 113:124

Nov 14 03:35:45 MyWebUniversity ntpd[16709]: Soliciting pool server 209.50.63.74
Nov 14 03:35:45 MyWebUniversity ntpd[16709]: Soliciting pool server 209.51.161.238
Nov 14 03:35:45 MyWebUniversity ntpd[16709]: Soliciting pool server 69.28.91.73
Nov 14 03:35:45 MyWebUniversity ntpd[16709]: Soliciting pool server 91.189.91.157
Nov 14 03:35:46 MyWebUniversity ntpd[16709]: Soliciting pool server 91.189.89.198
Nov 14 03:35:46 MyWebUniversity ntpd[16709]: Soliciting pool server 193.187.181.6
Nov 14 03:35:46 MyWebUniversity ntpd[16709]: Soliciting pool server 207.192.69.118
Nov 14 03:35:47 MyWebUniversity ntpd[16709]: Soliciting pool server 91.189.94.4
Nov 14 03:35:47 MyWebUniversity ntpd[16709]: Soliciting pool server 52.6.160.3
Nov 14 03:35:47 MyWebUniversity ntpd[16709]: Soliciting pool server 2600:3c03:e002:1300::10

Here is the related ntp important configuration files, that if you find out your system is not synchronized the date and time accurately to modify and restart the ntp daemon process.

root@MyWebUniversity:/etc**# ls -l /etc/ntp.conf /lib/systemd/system/ntp.service**
-rw-r--r-- 1 root root 2517 Aug 17 2020 **/etc/ntp.conf**
-rw-r--r-- 1 root root 354 Dec 13 **2017 /lib/systemd/system/ntp.service**

To stop the ntp daemon process, please use the command below:
root@MyWebUniversity:/etc# **systemct**l **stop ntp**

To check the status of the ntp daemon process, you can use the command below:
root@MyWebUniversity:/etc# **systemctl status ntp**
● ntp.service - Network Time Service
  Loaded: loaded (/lib/systemd/system/ntp.service; enabled; vendor preset: enabled)
  Active: inactive (dead) since Sun 2021-11-14 03:55:47 EST; 7s ago
    Docs: man:ntpd(8)
 Main PID: 16709 (code=exited, status=0/SUCCESS)

Nov 14 03:55:47 MyWebUniversity ntpd[16709]: 69.28.91.73 local addr 74.208.220.231 -> <null>
Nov 14 03:55:47 MyWebUniversity ntpd[16709]: 209.51.161.238 local addr 74.208.220.231 -> <null>
Nov 14 03:55:47 MyWebUniversity ntpd[16709]: 209.50.63.74 local addr 74.208.220.231 -> <null>
Nov 14 03:55:47 MyWebUniversity ntpd[16709]: 91.189.91.157 local addr 74.208.220.231 -> <null>
Nov 14 03:55:47 MyWebUniversity ntpd[16709]: 207.192.69.118 local addr 74.208.220.231 -> <null>
Nov 14 03:55:47 MyWebUniversity ntpd[16709]: 193.187.181.6 local addr 74.208.220.231 -> <null>
Nov 14 03:55:47 MyWebUniversity ntpd[16709]: 91.189.89.198 local addr 74.208.220.231 -> <null>
Nov 14 03:55:47 MyWebUniversity ntpd[16709]: 52.6.160.3 local addr 74.208.220.231 -> <null>
Nov 14 03:55:47 MyWebUniversity ntpd[16709]: 91.189.94.4 local addr 74.208.220.231 -> <null>
Nov 14 03:55:47 MyWebUniversity systemd[1]: Stopped Network Time Service.

As shown in the above output of the ntp status, the daemon was stopped previously by myself, so I need to start it again.
root@MyWebUniversity:/etc# **systemctl start ntp**

Now, that I started the ntp daemon process again, the status is showing it is up and **running**.
root@MyWebUniversity:/etc# **systemctl status ntp**
● ntp.service - Network Time Service
  Loaded: loaded (/lib/systemd/system/ntp.service; enabled; vendor preset: enabled)
  Active: active **(running)** since Sun 2021-11-14 03:56:00 EST; 4s ago
    Docs: man:ntpd(8)
 Process: 19700 ExecStart=/usr/lib/ntp/ntp-systemd-wrapper (code=exited, status=0/SUCCESS)
 Main PID: 19717 (ntpd)
   Tasks: 2 (limit: 4915)
  CGroup: /system.slice/ntp.service
          └─19717 /usr/sbin/ntpd -p /var/run/ntpd.pid -g -u 113:124

Nov 14 03:56:00 MyWebUniversity ntpd[19717]: Listening on routing socket on fd #22 for interface upd
Nov 14 03:56:01 MyWebUniversity ntpd[19717]: Soliciting pool server 192.189.65.187
Nov 14 03:56:02 MyWebUniversity ntpd[19717]: Soliciting pool server 104.131.155.175
Nov 14 03:56:02 MyWebUniversity ntpd[19717]: Soliciting pool server 206.81.5.45
Nov 14 03:56:03 MyWebUniversity ntpd[19717]: Soliciting pool server 152.67.228.49
Nov 14 03:56:03 MyWebUniversity ntpd[19717]: Soliciting pool server 45.79.1.70
Nov 14 03:56:03 MyWebUniversity ntpd[19717]: Soliciting pool server 162.159.200.1
Nov 14 03:56:04 MyWebUniversity ntpd[19717]: Soliciting pool server 159.203.158.197
Nov 14 03:56:04 MyWebUniversity ntpd[19717]: Soliciting pool server 64.142.54.12
Nov 14 03:56:04 MyWebUniversity ntpd[19717]: Soliciting pool server 23.31.92.161
Nov 14 03:56:04 MyWebUniversity ntpd[19717]: Soliciting pool server 151.236.30.71

We can also use the restart instead of the stop and start command to perform the same outcome. The only difference is that when you use the stop command all processes will go down until you use the start command to bring them back, the

restart command will not shutdown all processes and that are active, but it refreshes the newer connections as they come and may assign new PID to the Main PID as shown in this example.

```
root@MyWebUniversity:/etc# systemctl restart ntp
root@MyWebUniversity:/etc# systemctl status ntp
● ntp.service - Network Time Service
   Loaded: loaded (/lib/systemd/system/ntp.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2021-11-14 03:56:13 EST; 2s ago
     Docs: man:ntpd(8)
  Process: 19948 ExecStart=/usr/lib/ntp/ntp-systemd-wrapper (code=exited, status=0/SUCCESS)
 Main PID: 19966 (ntpd)
    Tasks: 2 (limit: 4915)
   CGroup: /system.slice/ntp.service
           └─19966 /usr/sbin/ntpd -p /var/run/ntpd.pid -g -u 113:124

Nov 14 03:56:13 MyWebUniversity ntpd[19966]: Listen and drop on 0 v6wildcard [::]:123
Nov 14 03:56:13 MyWebUniversity ntpd[19966]: Listen and drop on 1 v4wildcard 0.0.0.0:123
Nov 14 03:56:13 MyWebUniversity ntpd[19966]: Listen normally on 2 lo 127.0.0.1:123
Nov 14 03:56:13 MyWebUniversity ntpd[19966]: Listen normally on 3 eth0 74.208.220.231:123
Nov 14 03:56:13 MyWebUniversity ntpd[19966]: Listen normally on 4 lo [::1]:123
Nov 14 03:56:13 MyWebUniversity ntpd[19966]: Listen normally on 5 eth0 [fe80::225:90ff:fe73:775c%2]:
Nov 14 03:56:13 MyWebUniversity ntpd[19966]: Listening on routing socket on fd #22 for interface upd
Nov 14 03:56:14 MyWebUniversity ntpd[19966]: Soliciting pool server 23.31.92.161
Nov 14 03:56:15 MyWebUniversity ntpd[19966]: Soliciting pool server 45.79.1.70
Nov 14 03:56:15 MyWebUniversity ntpd[19966]: Soliciting pool server 38.229.56.9
```

How to use the date command to change the system date and time?

In this example below, I am showing you some of the ways you can change the system date and time locally . Please understand that in order for the system date to automatically synchronized with accurate date and time, you need to configure your system with Network Time Protocol Daemon (ntpd). The Network Time Protocol Daemon (ntpd), Network Time Protocol Query (ntpq) and other related commands are explained in Chapter 4 under the "Advanced UNIX commands".

Here, I use the date command to find out the date and time.

$ date

Saturday, March 12, 2011 08:10:42 PM PST

In order to change the system date and time, I need to have root access, so I become root to run the date command and change the system date and time.

$ su -

Password:

Oracle Corporation SunOS 5.11 snv_151a November 2010

#

I use the date command as root to know what the current system date and time is.

 # date

Saturday, March 12, 2011 08:10:48 PM PST

Now, I change the time only to 8:16 pm from 8:10 pm.

 # date 0312201611

Saturday, March 12, 2011 08:16:00 PM PST

Here, I change only the year to 2010.

 # date 0312201611

Friday, March 12, 2010 08:16:00 PM PST

Now, I change the month only to January.

 # date 0312201610

Tuesday, January 12, 2010 08:16:00 PM PST

Here, I change the date to January 1, 2011 for same time.

 # date 0101201611

Saturday, January 1, 2011 08:16:00 PM PST

At this time, I want to change the date and time to correct date that I like to set it which is the below date and time.

 # date 0312201711

Saturday, March 12, 2011 08:17:00 PM PST

 # date

Saturday, March 12, 2011 08:17:01 PM PST

Please click on Please click on man date. to see the man pages for this command.

$ whatis echo

echo echo (1) - display a line of text

Examples:

1. echo "Hello, World!"
2. echo "Solaris Express!"

Please click on man echo. to see the man pages for this command.

The output of your command "echo "Hello, World!"" is:

Hello, World!

The output of your command "echo "Solaris, Express!"" is:

Solaris, Express!

More examples of the “echo” commands is shown below:

$ whatis head

head head (1) - output the first part of files

Examples:

1. head -1 /etc/hosts
2. head -2 /etc/hosts
3. head /etc/hosts

Please click on man head. to see the man pages for this command.

The output of your command "head -1 /etc/hosts" is:

127.0.0.1 localhost.localdomain localhost

The output of your command "head -2 /etc/hosts" is:

127.0.0.1 localhost.localdomain localhost
74.208.220.231 mywebuniversity.com u22942380.onlinehome-server.com u22942380

The output of your command "head /etc/hosts" is:

127.0.0.1 localhost.localdomain localhost
74.208.220.231 mywebuniversity.com u22942380.onlinehome-server.com u22942380
#74.208.220.231 mywebuniversity.com

$ whatis hostname

hostname - set or print name of current host system

How to use the hostname command to get or set the hostname?

hostname

MySolaris

hostname NewName

hostname

NewName

hostid

00c6ea02

hostname MySolaris

hostid

00c6ea02

hostname

MySolaris

#

Please click on

The Direct Way to Linux Ubuntu

$ whatis host

host - DNS lookup utility

How to use the host command to get information about other hosts?

$ host

Usage: host [-aCdlriTwv] [-c class] [-N ndots] [-t type] [-W time]

[-R number] [-m flag] hostname [server]

-a is equivalent to -v -t ANY

-c specifies query class for non-IN data

-C compares SOA records on authoritative nameservers

-d is equivalent to -v

-l lists all hosts in a domain, using AXFR

-i IP6.INT reverse lookups

-N changes the number of dots allowed before root lookup is done

-r disables recursive processing

-R specifies number of retries for UDP packets

-s a SERVFAIL response should stop query

-t specifies the query type

-T enables TCP/IP mode

-v enables verbose output

-w specifies to wait forever for a reply

-W specifies how long to wait for a reply

-4 use IPv4 query transport only

-6 use IPv6 query transport only

-m set memory debugging flag (trace|record|usage)

$ host MyWebUniversity.com

MyWebUniversity.com has address 69.65.10.238

MyWebUniversity.com mail is handled by 0 MyWebUniversity.com.

$ host www.google.com

www.google.com is an alias for www.l.google.com.

www.l.google.com has address 74.125.224.180

www.l.google.com has address 74.125.224.179

www.l.google.com has address 74.125.224.176

www.l.google.com has address 74.125.224.178

www.l.google.com has address 74.125.224.177

Please click on

The Direct Way to Linux Ubuntu

$ whatis id

id id (1) - print real and effective user and group IDs

Examples:

1. id
2. id -a
3. id -u
4. id -g
5. id -G

Please click on man id. to see the man pages for this command.

The output of your command "id" is:

uid=33(www-data) gid=33(www-data) groups=33(www-data)

The output of your command "id -a" is:

uid=33(www-data) gid=33(www-data) groups=33(www-data)

The output of your command "id -u" is:

33

The output of your command "id -g" is:

33

39

The output of your command "id -G" is:

33

$ whatis info

info - read Info documents

Examples:

1. info printf
2. info info
3. info man
4. info top
5. info systemctl
6. info systemd

Please click on man info. to see the man pages for this command.

The output of your command "info printf" is:

File: coreutils.info, Node: printf invocation, Next: yes invocation, Prev: echo invocation, Up: Printing text

15.2 'printf': Format and print data
====================================

'printf' does formatted printing of text. Synopsis:

```
printf FORMAT [ARGUMENT]...
```

'printf' prints the FORMAT string, interpreting '%' directives and '\' escapes to format numeric and string arguments in a way that is

mostly similar to the C 'printf' function. *Note 'printf' format directives: (libc)Output Conversion Syntax, for details. The differences are listed below.

Due to shell aliases and built-in 'printf' functions, using an unadorned 'printf' interactively or in a script may get you different functionality than that described here. Invoke it via 'env' (i.e., 'env printf ...') to avoid interference from the shell.

* The FORMAT argument is reused as necessary to convert all the given ARGUMENTs. For example, the command 'printf %s a b' outputs 'ab'.

* Missing ARGUMENTs are treated as null strings or as zeros, depending on whether the context expects a string or a number. For example, the command 'printf %sx%d' prints 'x0'.

* An additional escape, '\c', causes 'printf' to produce no further output. For example, the command 'printf 'A%sC\cD%sF' B E' prints 'ABC'.

* The hexadecimal escape sequence '\xHH' has at most two digits, as opposed to C where it can have an unlimited number of digits. For example, the command 'printf '\x07e'' prints two bytes, whereas the C statement 'printf ("\x07e")' prints just one.

* An additional directive '%b', prints its argument string with '\' escapes interpreted in the same way as in the FORMAT string, except that octal escapes are of the form '\0OOO' where OOO is 0 to 3 octal digits. If '\OOO' is nine-bit value, ignore the ninth bit. If a precision is also given, it limits the number of bytes printed from the converted string.

* An additional directive '%q', prints its argument string in a format that can be reused as input by most shells. Non-printable characters are escaped with the POSIX proposed '$''' syntax, and shell metacharacters are quoted appropriately. This is an equivalent format to 'ls --quoting=shell-escape' output.

* Numeric arguments must be single C constants, possibly with leading '+' or '-'. For example, 'printf %.4d -3' outputs '-0003'.

* If the leading character of a numeric argument is '"' or ''' then its value is the numeric value of the immediately following character. Any remaining characters are silently ignored if the 'POSIXLY_CORRECT' environment variable is set; otherwise, a warning is printed. For example, 'printf "%d" "'a"' outputs '97' on hosts that use the ASCII character set, since 'a' has the numeric value

97 in ASCII.

A floating-point argument must use a period before any fractional digits, but is printed according to the 'LC_NUMERIC' category of the current locale. For example, in a locale whose radix character is a comma, the command 'printf %g 3.14' outputs '3,14' whereas the command 'printf %g 3,14' is an error. *Note Floating point::.

'printf' interprets '\OOO' in FORMAT as an octal number (if OOO is 1 to 3 octal digits) specifying a byte to print, and '\xHH' as a hexadecimal number (if HH is 1 to 2 hex digits) specifying a character to print. Note however that when '\OOO' specifies a number larger than 255, 'printf' ignores the ninth bit. For example, 'printf '\400'' is equivalent to 'printf '\0''.

'printf' interprets two character syntaxes introduced in ISO C 99: '\u' for 16-bit Unicode (ISO/IEC 10646) characters, specified as four hexadecimal digits HHHH, and '\U' for 32-bit Unicode characters, specified as eight hexadecimal digits HHHHHHHH. 'printf' outputs the Unicode characters according to the 'LC_CTYPE' locale. Unicode characters in the ranges U+0000...U+009F, U+D800...U+DFFF cannot be specified by this syntax, except for U+0024 ($), U+0040 (@), and U+0060 ()??.

The processing of '\u' and '\U' requires a full featured 'iconv' facility. It is activated on systems with glibc 2.2 (or newer), or when 'libiconv' is installed prior to this package. Otherwise '\u' and '\U' will print as-is.

The only options are a lone '--help' or '--version'. *Note Common options::. Options must precede operands.

The Unicode character syntaxes are useful for writing strings in a locale independent way. For example, a string containing the Euro currency symbol

```
$ env printf '\u20AC 14.95'
```

will be output correctly in all locales supporting the Euro symbol (ISO-8859-15, UTF-8, and others). Similarly, a Chinese string

```
$ env printf '\u4e2d\u6587'
```

will be output correctly in all Chinese locales (GB2312, BIG5, UTF-8, etc).

Note that in these examples, the 'printf' command has been invoked

via 'env' to ensure that we run the program found via your shell's search path, and not a shell alias or a built-in function.

For larger strings, you don't need to look up the hexadecimal code values of each character one by one. ASCII characters mixed with \u escape sequences are also known as the JAVA source file encoding. You can use GNU recode 3.5c (or newer) to convert strings to this encoding. Here is how to convert a piece of text into a shell script which will output this text in a locale-independent way:

```
$ LC_CTYPE=zh_CN.big5 /usr/local/bin/printf \
    '\u4e2d\u6587\n' > sample.txt
$ recode BIG5..JAVA < sample.txt \
    | sed -e "s|^|/usr/local/bin/printf '|" -e "s|$|\\\\n'|" \
    > sample.sh
```

An exit status of zero indicates success, and a nonzero value indicates failure.

As another example, let's get the "info" on "pthon3" version by searching on the URL below:
http://www.mywebuniversity.com/cgi-bin/lincmd.cgi?cmd=info+python3

The output of your command "info python3" is:

PYTHON(1) General Commands Manual PYTHON(1)

NAME
python - an interpreted, interactive, object-oriented programming language

SYNOPSIS
python [-B] [-b] [-d] [-E] [-h] [-i] [-I]
[-m module-name] [-q] [-O] [-OO] [-s] [-S] [-u]
[-v] [-V] [-W argument] [-x] [[-X option] -?]
[-c command | script | -] [arguments]

DESCRIPTION
Python is an interpreted, interactive, object-oriented programming language that combines remarkable power with very clear syntax. For an introduction to programming in Python, see the Python Tutorial. The

Python Library Reference documents built-in and standard types, constants, functions and modules. Finally, the Python Reference Manual describes the syntax and semantics of the core language in (perhaps too) much detail. (These documents may be located via the INTERNET RESOURCES below; they may be installed on your system as well.)

Python's basic power can be extended with your own modules written in C or C++. On most systems such modules may be dynamically loaded. Python is also adaptable as an extension language for existing applications. See the internal documentation for hints.

Documentation for installed Python modules and packages can be viewed by running the pydoc program.

COMMAND LINE OPTIONS

-B Don't write .pyc files on import. See also PYTHONDONTWRITEBYTECODE.

-b Issue warnings about str(bytes instance), str(bytearray_instance) and comparing bytes/bytearray with str. (-bb: issue errors)

-c command
Specify the command to execute (see next section). This terminates the option list (following options are passed as arguments to the command).

-d Turn on parser debugging output (for wizards only, depending on compilation options).

-E Ignore environment variables like PYTHONPATH and PYTHONHOME that modify the behavior of the interpreter.

-h , -? , --help
Prints the usage for the interpreter executable and exits.

-i When a script is passed as first argument or the -c option is used, enter interactive mode after executing the script or the command. It does not read the $PYTHONSTARTUP file. This can be useful to inspect global variables or a stack trace when a script raises an exception.

-I Run Python in isolated mode. This also implies -E and -s. In isolated mode sys.path contains neither the script's directory nor the user's site-packages directory. All PYTHON* environment variables are ignored, too. Further restrictions may be imposed to prevent the user from injecting malicious code.

-m module-name
Searches sys.path for the named module and runs the corresponding .py file as a script.

-O Remove assert statements and any code conditional on the value of __debug__; augment the filename for compiled (bytecode) files by adding .opt-1 before the .pyc extension.

-OO Do -O and also discard docstrings; change the filename for compiled (bytecode) files by adding .opt-2 before the .pyc extension.

-q Do not print the version and copyright messages. These messages are also suppressed in non-interactive mode.

-s Don't add user site directory to sys.path.

-S Disable the import of the module site and the site-dependent manipulations of sys.path that it entails. Also disable these manipulations if site is explicitly imported later.

-u Force the binary I/O layers of stdout and stderr to be unbuffered. stdin is always buffered. The text I/O layer will still be line-buffered.

-v Print a message each time a module is initialized, showing the place (filename or built-in module) from which it is loaded. When given twice, print a message for each file that is checked for when searching for a module. Also provides information on module cleanup at exit.

-V , --version
Prints the Python version number of the executable and exits. When given twice, print more information about the build.

-W argument
Warning control. Python sometimes prints warning message to sys.stderr. A typical warning message has the following form: file:line: category: message. By default, each warning is printed once for each source line where it occurs. This option controls how often warnings are printed. Multiple -W options may be given; when a warning matches more than one option, the action for the last matching option is performed. Invalid -W options are ignored (a warning message is printed about invalid options when the first warning is issued). Warnings can also be controlled from within a Python program using the warnings mod-

ule.

The simplest form of argument is one of the following action strings (or a unique abbreviation): ignore to ignore all warnings; default to explicitly request the default behavior (printing each warning once per source line); all to print a warning each time it occurs (this may generate many messages if a warning is triggered repeatedly for the same source line, such as inside a loop); module to print each warning only the first time it occurs in each module; once to print each warning only the first time it occurs in the program; or error to raise an exception instead of printing a warning message.

The full form of argument is action:message:category:module:line. Here, action is as explained above but only applies to messages that match the remaining fields. Empty fields match all values; trailing empty fields may be omitted. The message field matches the start of the warning message printed; this match is case-insensitive. The category field matches the warning category. This must be a class name; the match test whether the actual warning category of the message is a subclass of the specified warning category. The full class name must be given. The module field matches the (fully-qualified) module name; this match is case-sensitive. The line field matches the line number, where zero matches all line numbers and is thus equivalent to an omitted line number.

-X option
Set implementation specific option.

-x Skip the first line of the source. This is intended for a DOS specific hack only. Warning: the line numbers in error messages will be off by one!

INTERPRETER INTERFACE

The interpreter interface resembles that of the UNIX shell: when called with standard input connected to a tty device, it prompts for commands and executes them until an EOF is read; when called with a file name argument or with a file as standard input, it reads and executes a script from that file; when called with -c command, it executes the Python statement(s) given as command. Here command may contain multiple statements separated by newlines. Leading whitespace is significant in Python statements! In non-interactive mode, the entire input is parsed before it is executed.

If available, the script name and additional arguments thereafter are passed to the script in the Python variable sys.argv, which is a list

of strings (you must first import sys to be able to access it). If no script name is given, sys.argv[0] is an empty string; if -c is used, sys.argv[0] contains the string '-c'. Note that options interpreted by the Python interpreter itself are not placed in sys.argv.

In interactive mode, the primary prompt is `>>>'; the second prompt (which appears when a command is not complete) is `...'. The prompts can be changed by assignment to sys.ps1 or sys.ps2. The interpreter quits when it reads an EOF at a prompt. When an unhandled exception occurs, a stack trace is printed and control returns to the primary prompt; in non-interactive mode, the interpreter exits after printing the stack trace. The interrupt signal raises the KeyboardInterrupt exception; other UNIX signals are not caught (except that SIGPIPE is sometimes ignored, in favor of the IOError exception). Error messages are written to stderr.

FILES AND DIRECTORIES

These are subject to difference depending on local installation conventions; ${prefix} and ${exec_prefix} are installation-dependent and should be interpreted as for GNU software; they may be the same. On Debian GNU/{Hurd,Linux} the default for both is /usr.

${exec_prefix}/bin/python
Recommended location of the interpreter.

${prefix}/lib/python
${exec_prefix}/lib/python
Recommended locations of the directories containing the standard modules.

${prefix}/include/python
${exec_prefix}/include/python
Recommended locations of the directories containing the include files needed for developing Python extensions and embedding the interpreter.

ENVIRONMENT VARIABLES

PYTHONHOME
Change the location of the standard Python libraries. By default, the libraries are searched in ${prefix}/lib/python and ${exec_prefix}/lib/python, where ${prefix}
and ${exec_prefix} are installation-dependent directories, both defaulting to /usr/local. When $PYTHONHOME is set to a single directory, its value replaces both ${prefix} and ${exec_prefix}. To specify different values for these, set $PYTHONHOME to ${prefix}:${exec_prefix}.

PYTHONPATH

Augments the default search path for module files. The format is the same as the shell's $PATH: one or more directory pathnames separated by colons. Non-existent directories are silently ignored. The default search path is installation dependent, but generally begins with ${prefix}/lib/python (see PYTHONHOME above). The default search path is always appended to $PYTHONPATH. If a script argument is given, the directory containing the script is inserted in the path in front of $PYTHONPATH. The search path can be manipulated from within a Python program as the variable sys.path.

PYTHONSTARTUP

If this is the name of a readable file, the Python commands in that file are executed before the first prompt is displayed in interactive mode. The file is executed in the same name space where interactive commands are executed so that objects defined or imported in it can be used without qualification in the interactive session. You can also change the prompts sys.ps1 and sys.ps2 in this file.

PYTHONOPTIMIZE

If this is set to a non-empty string it is equivalent to specifying the -O option. If set to an integer, it is equivalent to specifying -O multiple times.

PYTHONDEBUG

If this is set to a non-empty string it is equivalent to specifying the -d option. If set to an integer, it is equivalent to specifying -d multiple times.

PYTHONDONTWRITEBYTECODE

If this is set to a non-empty string it is equivalent to specifying the -B option (don't try to write .pyc files).

PYTHONINSPECT

If this is set to a non-empty string it is equivalent to specifying the -i option.

PYTHONIOENCODING

If this is set before running the interpreter, it overrides the encoding used for stdin/stdout/stderr, in the syntax encodingname:errorhandler The errorhandler part is optional and has the same meaning as in str.encode. For stderr, the errorhandler part is ignored; the handler will always be 'backslashreplace'.

PYTHONNOUSERSITE

If this is set to a non-empty string it is equivalent to specifying the -s option (Don't add the user site directory to sys.path).

PYTHONUNBUFFERED
If this is set to a non-empty string it is equivalent to specifying the -u option.

PYTHONVERBOSE
If this is set to a non-empty string it is equivalent to specifying the -v option. If set to an integer, it is equivalent to specifying -v multiple times.

PYTHONWARNINGS
If this is set to a comma-separated string it is equivalent to specifying the -W option for each separate value.

PYTHONHASHSEED
If this variable is set to "random", a random value is used to seed the hashes of str, bytes and datetime objects.

If PYTHONHASHSEED is set to an integer value, it is used as a fixed seed for generating the hash() of the types covered by the hash randomization. Its purpose is to allow repeatable hashing, such as for selftests for the interpreter itself, or to allow a cluster of python processes to share hash values.

The integer must be a decimal number in the range [0,4294967295]. Specifying the value 0 will disable hash randomization.

AUTHOR
The Python Software Foundation: https://www.python.org/psf/

INTERNET RESOURCES
Main website: https://www.python.org/
Documentation: https://docs.python.org/
Developer resources: https://devguide.python.org/
Downloads: https://www.python.org/downloads/
Module repository: https://pypi.org/
Newsgroups: comp.lang.python, comp.lang.python.announce

LICENSING
Python is distributed under an Open Source license. See the file "LICENSE" in the Python source distribution for information on terms & conditions for accessing and otherwise using Python and for a DISCLAIMER OF ALL WARRANTIES.

In the above website, I have also provided you with searching mechanisms that would work provided the argument is passed properly to the above "**lincmd.cgi"** which I have also written it in **"Python"** version 3 in this case. For instance, one can simply change the string "python3" to any other valid Linux commands such as **"pwd, pwdx, pstree, strace, top, free, vmstat, iostat, gdb, cat, cal, who, date, man, df, systemd, systemctl**" to name some. In addition to compilers name like **"gcc",** interpreters name like "**python or python3**", **Python Package Index "pip or pip3**".

For instance, you can get the info on Process Tree command using the "info pstree" at the Linux command prompt or use the manual pages by running the command "man pstree" at a Linux terminal. However, you can get the output below by simply visiting my website and get the below output as well.

http://www.mywebuniversity.com/cgi-bin/lincmd.cgi?cmd=info+pstree

The Direct Way to Linux Ubuntu

The output of your command "info pstree" is:

PSTREE(1) User Commands PSTREE(1)

NAME
pstree – display a tree of processes

SYNOPSIS
pstree [-a, --arguments] [-c, --compact] [-h, --high-
light-all, -Hpid, --highlight-pid pid] [-g] –show-pgids] [-l, --long]
[-n, --numeric-sort] [-N, --ns-sortns [-p, --show-pids]
[-s, --show-parents] [-S, --ns-changes] [-t, --thread-names]
[-T, --hide-threads] [-u, --uid-changes] [-Z, --security-context]
[-A, --ascii, -G, --vt100, -U, --unicode] [pid, user]
pstree -V, --version

DESCRIPTION

pstree shows running processes as a tree. The tree is rooted at either pid or init if pid is omitted. If a user name is specified, all process trees rooted at processes owned by that user are shown.

Pstree visually merges identical branches by putting them in square brackets and prefixing them with the repetition count, e.g.

```
init-+-getty
     |-getty
     |-getty
     `-getty
```

becomes

```
init---4*[getty]
```

Child threads of a process are found under the parent process and are shown with the process name in curly braces, e.g.

```
icecast2---13*[{icecast2}]
```

If pstree is called as pstree.x11 then it will prompt the user at the end of the line to press return and will not return until that has happened. This is useful for when pstree is run in a xterminal.

Certain kernel or mount parameters, such as the hidepid option for procfs, will hide information for some processes. In these situations pstree will attempt to build the tree without this information, showing process names as question marks.

OPTIONS

-a Show command line arguments. If the command line of a process is swapped out, that process is shown in parentheses. -a implicitly disables compaction for processes but not threads.

-A Use ASCII characters to draw the tree.

-c Disable compaction of identical subtrees. By default, subtrees are compacted whenever possible.

-G Use VT100 line drawing characters.

-h Highlight the current process and its ancestors. This is a no-op if the terminal doesn't support highlighting or if neither the current process nor any of its ancestors are in the subtree being shown.

-H Like -h, but highlight the specified process instead. Unlike with -h, pstree fails when using -H if highlighting is not available.

-g Show PGIDs. Process Group IDs are shown as decimal numbers in parentheses after each process name. -g implicitly disables compaction. If both PIDs and PGIDs are displayed then PIDs are shown first.

-l Display long lines. By default, lines are truncated to either the COLUMNS environment variable or the display width. If neither of these methods work, the default of 132 columns is used.

-n Sort processes with the same ancestor by PID instead of by name. (Numeric sort.)

-N Show individual trees for each namespace of the type specified. The available types are: ipc, mnt, net, pid, user, uts. Regular users don't have access to other users' processes information, so the output will be limited.

-p Show PIDs. PIDs are shown as decimal numbers in parentheses after each process name. -p implicitly disables compaction.

-s Show parent processes of the specified process.

-S Show namespaces transitions. Like -N, the output is limited when running as a regular user.

-t Show full names for threads when available.

-T Hide threads and only show processes.

-u Show uid transitions. Whenever the uid of a process differs from the uid of its parent, the new uid is shown in parentheses after the process name.

-U Use UTF-8 (Unicode) line drawing characters. Under Linux 1.1-54 and above, UTF-8 mode is entered on the console with echo -e ' 33%8' and left with echo -e ' 33%@'

-V Display version information.

-Z (SELinux) Show security context for each process. This flag will only work if pstree is compiled with SELinux support.

FILES
/proc location of the proc file system

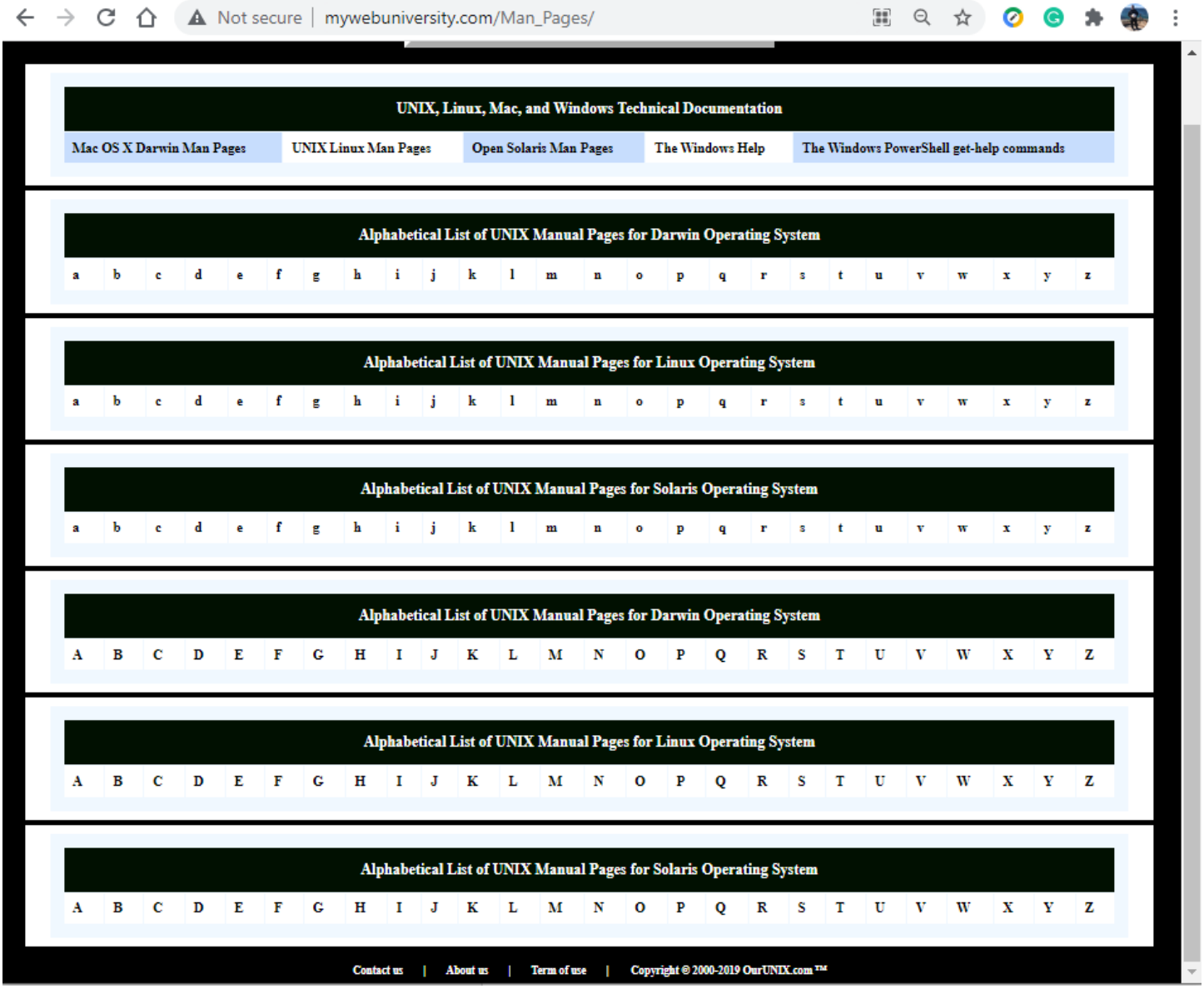
BUGS
Some character sets may be incompatible with the VT100 characters.

SEE ALSO
ps(1), top(1).

Psmisc 2016-06-18 PSTREE(1)

The Linux Manual Pages can also be obtained by visiting my website Linux and UNIX Manual Pages, where you can get the manual Pages for All Linux Ubuntu or CentOS, Oracle Sun Solaris Express or Sun Solaris, Mac OS X or Darwin BSD version, as well as Windows Help and Windows PowerShell command let. Here is the URL where you will find more than sixty thousands of technical documentation sorted by Operating System and alphabetically in ascending order that is available at your fingertip 24/7, where you can learn and practice them as well.

http://www.mywebuniversity.com/Man_Pages/



The Direct Way to Linux Ubuntu

By visiting the URL: http://www.mywebuniversity.com/cgi-bin/CH2.cgi?cmd=Paste-h.html you will learn about the usage of the command **"paste"** that merges lines between files as shown below:

$ whatis paste

paste paste (1) - merge lines of files

paste paste (1) - merge corresponding or subsequent lines of files

Examples:

In the examples below, I am creating the file1.txt, file2.txt and file3.txt and then I paste the contents of them to the screen, or a second file1 and file2.txt, or file1-3.txt and then display the results for verification.

$ echo "content of file one" > file1.txt

$ echo "content of file two" > file2.txt

$ ls -ld file?.txt

-rw-r--r-- 1 wlutfy staff 20 2010-12-26 22:54 file1.txt

-rw-r--r-- 1 wlutfy staff 20 2010-12-26 22:54 file2.txt

$ cat file1.txt

content of file one

$ cat file2.txt

content of file two

$ paste file1.txt file2.txt

content of file one content of file two

$ cat file1.txt

content of file one

$ cat file2.txt

content of file two

$ paste file1.txt file2.txt > file1-and-file2.txt

$ cat file1-and-file2.txt

content of file one content of file two

$ echo "content of file three" > file3.txt

$ cat file3.txt

content of file three

$ paste file1.txt file2.txt file3.txt > file1-3.txt

$ cat file1-3.txt

content of file one content of file two content of file three

Please click on man paste. to see the man pages for this command.

By visiting the URL: http://www.mywebuniversity.com/cgi-bin/CH2.cgi?cmd=Ps-h.html you will learn about the usage of the command **"ps"** which reports about the status of the running processes on the system as shown below:

$ whatis ps

ps ps (1) - report process status

ps ps (1b) - display the status of current processes

Examples:

1. ps -u root
2. ps -p 1,2,3
3. ps -efa | grep -i http
4. ps -efa | grep apache

Please click on man ps. to see the man pages for this command.

Above are four examples that you can run by simply clicking on them and see the output of them at my website. For instance, let's simply click on option **3**.which is "**3. ps -efa | grep -I http**" as shown above.

http://www.mywebuniversity.com/cgi-bin/lincmd.cgi?cmd=ps+-efa+|+grep+-i+http

When we click on the ""**3. ps -efa | grep -I http**" the string is passed via the **get method** of the **lincmd.cgi** with the below parameters and after executing the above command in our Linux Server, the output below displays.

The Direct Way to Linux Ubuntu

The output of your command "ps -efa | grep -i http" is:

```
www-data 30278 30277  0 23:53 ?        00:00:00 /bin/sh -c ps -efa | grep -i http
www-data 30280 30278  0 23:53 ?        00:00:00 grep -i http
```

Similarly, when I click on number **"4. Ps -efa | grep apache"** the output below shows up.

The Direct Way to Linux Ubuntu

The output of your command "ps -efa | grep apache" is:

```
root      7962     1  0 May16 ?        00:06:30 /usr/sbin/apache2 -k start
www-data 20208  7962  0 06:25 ?        00:00:00 /usr/sbin/apache2 -k start
www-data 20209  7962  0 06:25 ?        00:00:00 /usr/sbin/apache2 -k start
www-data 20210  7962  0 06:25 ?        00:00:00 /usr/sbin/apache2 -k start
www-data 30297 30296  0 23:57 ?        00:00:00 /bin/sh -c ps -efa | grep apache
www-data 30299 30297  0 23:57 ?        00:00:00 grep apache
```

The first and second examples are left as an exercise for you, please click on them to see the result yourself.

By visiting the URL: http://www.mywebuniversity.com/cgi-bin/CH2.cgi?cmd=Pwd-h.html you will learn about the usage of the command **"pwd"** that stands for "**p**rint **w**orking **d**irectory" as shown below:

$ whatis pwd

pwd pwd (1) - print name of current/working directory

Examples:

1. pwd

Please click on man pwd. to see the man pages for this command.

In addition to the command ", there is also an environment variable that is called **"PWD"**, which is all in upper-case letter that is pointing to the current working directory. As shown below, when I echo **$PWD**, I see that my current working directory is **"/home/wlutfy**", and after changing my directory to the subdirectory **"Programs**" by running the command **"cd Programs**", and "**echo $PWD"** shows my current working directory changed to **"/home/wlutfy/Programs"**.

wlutfy@MyWebUniversity:~$ **echo $PWD**

/home/wlutfy

wlutfy@MyWebUniversity:~$ **cd Programs**

wlutfy@MyWebUniversity:~/Programs$ **echo $PWD**

/home/wlutfy/Programs

wlutfy@MyWebUniversity:~/Programs$

Below is an example of using the command **"pwd"** and getting the output as shown**:**

The output of your command "pwd" is:

/usr/lib/cgi-bin

In addition to the command **"pwd**", there is also another command named **"pwdx**" which shows the executable path where the process started. In the example below, I am running the command "ps" to simply get my running process ids **"PIDs**" and the executables. Here, I am interested in finding the current working directory for my "**bash" PID** which is the process "**29509"** that is highlighted in **bold**.

As you can see from the output the **"/home/wlutfy/Programs**" is the current working directory for it.

wlutfy@MyWebUniversity:~/Programs$ **ps**

PID TTY TIME CMD

29509 pts/0 00:00:00 bash

30444 pts/0 00:00:00 ps

wlutfy@MyWebUniversity:~/Programs$ **pwdx 29509**

29509: /home/wlutfy/Programs

As a matter of fact, the **"pwdx**" is one of the **Linux and UNIX (MAC OS X, and Oracle Sun Solaris) "/proc File System**" which is also known as Pseudo File System that is based on memory, thus the file size is zero by long since the content of it is generated dynamically from memory. In chapter 4 Advanced UNIX and Linux commands, we will cover some more examples of the /proc file system such as **"pwdx", "pmap", "pstree", "pkill", "pgrep",** and **"pldd".** In UNIX Oracle Sun Solaris, we also have the command "**ptree**" that generate the process tree for a process. In Linux the equivalent command is called "**pstre**e" which stands for "process stream tree", but it does exactly what the Sun Solaris version of the "**ptree**" command do, not sure why the developer chose to rename the Solaris "ptree" command to "pstree, when both commands are working on stream of I/O. I am not too worried about it, but for consistency purposes, the other commands are named the same on both platforms such as **"pmap", "pwdx**", **"pldd",** etc. Since", here I am comparing both Operating systems in terms of providing rich features of the /proc file systems, and list of available commands, I would also like to mention in Oracle Sun Solaris there is additional processes such as "**pstop", "prun", "pstack", "preap**" "truss" commands that are very powerful and helpful when you need them. For instance, the command "**preap**" will take care of any zombie or defunct processes. On the other hand, in Linux when you have a lot of the zombie processes processes, there is no "preap" equivalent command to take care of the zombie processes and sometimes the system goes into degradation status and the only option is to reboot the system. However, sometimes using the List of Open Files "**lsof"** command, I have been able to identify the culprit defunct processes before they become problematic and have using a combination of the **"kill"** which a while read to identify all the parent process ids "**PID**s" before they turn into **zombies.** Sometimes, it is too late to clean up on the orphan or defunct process, due to the fact the parent process has been terminated or killed and the **"init**" or **"systemd**" which is the mother of all processes adapts the orphan defunct process and try to give it the resources it needs and then it will clear the kernel symbol table. In Sun Solaris I have never faced into a situation that the "init", fails to take care of the defunct process, or I could have not take care of the orphan process using the **"preap"** or "**lsof**", or combination of multiple commands as root by identifying the culprit processes and taken care of them all.

Now, we are looking into the command "uname" which provide us the system information based on the argument or options we provide. Below I am showing multiple of these commands, Please feel free to simply click on each one of the examples below to see the output and memorize the command so you can do the same at any UNIX or Linux terminal session.

$ whatis uname

uname uname (1) - print system information

Examples:

1. uname
2. uname -a
3. uname -s
4. uname -n
5. uname -r
6. uname -m
7. uname -p
8. uname -i
9. uname -sr

Please click on man uname. to see the man pages for this command.

The output of your command "uname" is:

Linux

The **“-a”** option of the **“uname”** command provides all the information about the system as shown below:

The output of your command "uname -a" is:

Linux MyWebUniversity 5.4.0-67-generic #75~18.04.1-Ubuntu SMP Tue Feb 23 19:17:50 UTC 2021 x86_64 x86_64 x86_64 GNU/Linux

The **“-s”** option of the **“uname”** command provides the operating system name in this case **“Linux”**. This is equivalent to the **“uname”** command which by default provides the system name.

The output of your command "uname -s" is:

Linux

The **“-n”** option of the **“uname”** command provides node or hostname as shown below:

The output of your command "uname -n" is:

MyWebUniversity

The **“-r”** option of the **“uname”** command provides revision or build for the OS version as shown below:

The output of your command "uname -r" is:

5.4.0-67-generic

The **“-m”** option of the **“uname”** command provides machine type in this case **“x86”** architecture “**64 bits**” as shown below:

The output of your command "uname -m" is:

x86_64

The **“-p”** option of the **“uname”** command provides processor type in this case also **“x86”** architecture “**64 bits**” as shown below:

The output of your command "uname -p" is:

x86_64

The **“-i”** option of the **“uname”** command provides hardware platform information in this case also **“x86”** architecture “**64 bits**” as shown below:

The output of your command "uname -i" is:

x86_64

You can also pass multiple **“-sr”** options, in the below example I am using “s” to the system in this case “**Linux**” and the option **“r”** to get the revision or build in this case “**5.4.0-67-generic**” as shown below:

The output of your command "uname -sr" is:

Linux 5.4.0-67-generic
In the below example of the “less” command, I am showing multiple ways you can use this command since on the web, the interactivity is almost does not make much sense and you have to use a terminal to practice this command.

The Direct Way to Linux Ubuntu

$ whatis less

less - opposite of more

Examples:

1. less /etc/hosts

whatis less less less (1) - opposite of more Examples: The command 'less' is similar to command 'more' , but it allows backward movement in the file as well as forward movement. The syntax is simmilar to the 'vi' editor. A list of commands can be displayed by simply pressing 'h or H' for help while using the less command.

$ less -V

Copyright (C) 1984-2007 Mark Nudelman

less comes with NO WARRANTY, to the extent permitted by law. p

For information about the terms of redistribution,

see the file named README in the less distribution.

Homepage: http://www.greenwoodsoftware.com/less

You can get a the list of commands at least four ways, by performaning the command 'man less', or running the command 'less -?', or 'less help', or 'less [FILENAME] and then type 'h or H' while reading the file [FILENAME]'. The list of commands to use are provided to you below, please practice some these commands to learn how to naviagate within less commands. What you learn here could also benefit you with the list of commands that you could use within the ' vi and vim' editor.

$ less --help

SUMMARY OF LESS COMMANDS

Commands marked with * may be preceded by a number, N.

Notes in parentheses indicate the behavior if N is given.

h H Display this help.

q :q Q :Q ZZ Exit.

MOVING

e ^E j ^N CR * Forward one line (or N lines).

y ^Y k ^K ^P * Backward one line (or N lines).

f ^F ^V SPACE * Forward one window (or N lines).

b ^B ESC-v * Backward one window (or N lines).

z * Forward one window (and set window to N).

w * Backward one window (and set window to N).

ESC-SPACE * Forward one window, but don't stop at end-of-file.

d ^D * Forward one half-window (and set half-window to N).

u ^U * Backward one half-window (and set half-window to N).

ESC-) RightArrow * Left one half screen width (or N positions).

ESC-(LeftArrow * Right one half screen width (or N positions).

F Forward forever; like "tail -f".

r ^R ^L Repaint screen.

R Repaint screen, discarding buffered input.

Default "window" is the screen height.

Default "half-window" is half of the screen height.

SEARCHING

/pattern * Search forward for (N-th) matching line.

?pattern * Search backward for (N-th) matching line.

n * Repeat previous search (for N-th occurrence).

N * Repeat previous search in reverse direction.

ESC-n * Repeat previous search, spanning files.

ESC-N * Repeat previous search, reverse dir. & spanning files.

ESC-u Undo (toggle) search highlighting.

Search patterns may be modified by one or more of:

^N or ! Search for NON-matching lines.

^E or * Search multiple files (pass thru END OF FILE).

^F or @ Start search at FIRST file (for /) or last file (for ?).

^K Highlight matches, but don't move (KEEP position).

^R Don't use REGULAR EXPRESSIONS.

JUMPING

g < ESC-< * Go to first line in file (or line N).

G > ESC-> * Go to last line in file (or line N).

p % * Go to beginning of file (or N percent into file).

t * Go to the (N-th) next tag.

T * Go to the (N-th) previous tag.

{ ([* Find close bracket })].

})] * Find open bracket { ([.

ESC-^F * Find close bracket .

ESC-^B * Find open bracket

Each "find close bracket" command goes forward to the close bracket

matching the (N-th) open bracket in the top line.

Each "find open bracket" command goes backward to the open bracket

matching the (N-th) close bracket in the bottom line.

m Mark the current position with .

' Go to a previously marked position.

'' Go to the previous position.

^X^X Same as '.

A mark is any upper-case or lower-case letter.

Certain marks are predefined:

^ means beginning of the file

$ means end of the file

CHANGING FILES

:e [file] Examine a new file.

^X^V Same as :e.

:n * Examine the (N-th) next file from the command line.

:p * Examine the (N-th) previous file from the command line.

:x * Examine the first (or N-th) file from the command line.

:d Delete the current file from the command line list.

= ^G :f Print current file name.

MISCELLANEOUS COMMANDS

- Toggle a command line option [see OPTIONS below].

-- Toggle a command line option, by name.

_ Display the setting of a command line option.

__ Display the setting of an option, by name.

+cmd Execute the less cmd each time a new file is examined.

!command Execute the shell command with $SHELL.

|Xcommand Pipe file between current pos & mark X to shell command.

v Edit the current file with $VISUAL or $EDITOR.

V Print version number of "less".

OPTIONS

Most options may be changed either on the command line,

or from within less by using the - or -- command.

Options may be given in one of two forms: either a single

character preceded by a -, or a name preceeded by --.

-? ..... --help

Display help (from command line).

-a ..... --search-skip-screen

Forward search skips current screen.

-b [N] .... --buffers=[N]

Number of buffers.

-B ..... --auto-buffers

Don't automatically allocate buffers for pipes.

-c ..... --clear-screen

Repaint by clearing rather than scrolling.

-d ..... --dumb

Dumb terminal.

-D [xn.n] . --color=xn.n

Set screen colors. (MS-DOS only)

-e -E .... --quit-at-eof --QUIT-AT-EOF

Quit at end of file.

-f ..... --force

Force open non-regular files.

-F ..... --quit-if-one-screen Quit if entire file fits on first screen.

-g ..... --hilite-search

Highlight only last match for searches.

-G ..... --HILITE-SEARCH

Don't highlight any matches for searches.

-h [N] .... --max-back-scroll=[N]

Backward scroll limit.

-i ..... --ignore-case

Ignore case in searches that do not contain uppercase.

-I ..... --IGNORE-CASE

Ignore case in all searches.

-j [N] .... --jump-target=[N]

Screen position of target lines.

-J ..... --status-column

Display a status column at left edge of screen.

-k [file] . --lesskey-file=[file]

Use a lesskey file.

-L ..... --no-lessopen

Ignore the LESSOPEN environment variable.

-m -M .... --long-prompt --LONG-PROMPT

Set prompt style.

-n -N .... --line-numbers --LINE-NUMBERS

Don't use line numbers.

-o [file] . --log-file=[file]

Copy to log file (standard input only).

-O [file] . --LOG-FILE=[file]

Copy to log file (unconditionally overwrite).

-p [pattern] --pattern=[pattern]

Start at pattern (from command line).

-P [prompt] --prompt=[prompt] Define new prompt.

-q -Q .... --quiet --QUIET --silent --SILENT

Quiet the terminal bell.

-r -R .... --raw-control-chars --RAW-CONTROL-CHARS

Output "raw" control characters.

-s ..... --squeeze-blank-lines

Squeeze multiple blank lines.

-S ..... --chop-long-lines

Chop long lines.

-t [tag] .. --tag=[tag]

Find a tag.

-T [tagsfile] --tag-file=[tagsfile]

Use an alternate tags file.

-u -U .... --underline-special --UNDERLINE-SPECIAL

Change handling of backspaces.

-V ..... --version

Display the version number of "less".

-w ..... --hilite-unread

Highlight first new line after forward-screen.

-W ..... --HILITE-UNREAD

Highlight first new line after any forward movement.

-x [N[,...]] --tabs=[N[,...]] Set tab stops.

-X ..... --no-init

Don't use termcap init/deinit strings.

--no-ke ypad

Don't use termcap keypad init/deinit strings.

-y [N] .... --max-forw-scroll=[N]

Forward scroll limit.

-z [N] .... --window=[N]

Set size of window.

-" [c[c]] . --quotes=[c[c]]

Set shell quote characters.

-~ ..... --tilde

Don't d isplay tildes after end of file.

-# [N] .... --shift=[N]

Horizontal scroll amount (0 = one half screen width)

LINE EDITING

These keys can be used to edit text being entered on the "command line" at the bottom of the screen.

RightArrow ESC-l Move cursor right one character.

LeftArrow ESC-h Move cursor left one character.

CNTL-RightArrow ESC-RightArrow ESC-w Move cursor right one word.

CNTL-LeftArrow ESC-LeftArrow ESC-b Move cursor left one word.

HOME ESC-0 Move cursor to start of line.

END ESC-$ Move cursor to end of line.

BACKSPACE Delete char to left of cursor.

DELETE ESC-x Delete char under cursor.

CNTL-BACKSPACE ESC-BACKSPACE Delete word to left of cursor.

CNTL-DELETE ESC-DELETE ESC-X Delete word under cursor.

CNTL-U ESC (MS-DOS only) Delete entire line.

UpArrow ESC-k Retrieve previous command line.

DownArrow ESC-j Retrieve next command line.

TAB Complete filename & cycle.

SHIFT-TAB ESC-TAB Complete filename & reverse cycle. -la

CNTL-L Complete filename, list all.

Please click on man less. to see the man pages for this command.

Below, is an example of the **“less”** command from our website.

The output of your command "less /etc/hosts" is:

127.0.0.1 localhost.localdomain localhost
74.208.220.231 mywebuniversity.com u22942380.onlinehome-server.com u22942380
#74.208.220.231 mywebuniversity.com

The Direct Way to Linux Ubuntu

$ whatis ls

ls ls (1) - list directory contents

Examples:

1. ls
2. ls -l
3. ls -L
4. ls -ltr
5. ls -lCF
6. ls -li
7. ls -li *.html | awk '{print $1,$10}'

Please click on man cal. to see the man pages for this command.

You are welcome to click on any of the above examples to see the output of the list "**ls**" command and the option given for each situation, I am showing the result of the command below which is example number 7 from the above. In the below example, I am using the **"-li**" option of the list **"ls**" command on all files **"*"** wildcard that are having the extension **".html**" and then **pipe "|"** (redirect the standard input to the commands followed after the symbol "**|**" pipe) it to the command **"awk"** and want to print only the first column shows the i**node**-number and the 10th column the file name with the output show below:.

The **inode** in Linux and UNIX is the data structure that holds all the information about the file system, and files since everything in UNIX and Linux is considered a file. A directory is a file, a device is a file, a process is a file in memory and finally everything is in the **inode** data structure

The output of your command "ls -li *.html | awk '{print $1,$10}'" is:

30264 Solaris.html
30233 head.html
30235 index.html
30279 tail.html

The command **"man"** which stands for "**manual"** pages, provides us with all UNIX and Linux Manual and Technical documentation available. You can get the "Linux", "Mac OS X", and "Oracle Sun Solaris" manual pages from our website or you can get it from the Vendor documentation for the particular OS. In our website, we are referring to multiple links to search engine like google.com, or companies like Oracle, Apple, RedHat, Ubuntu, CentOS and many more. Here is the UNIX and Linux Manual Pages that we generated by writing a Python program and parse and clean the documentation, so it is presentable on the web for you to learn from it.

You can also visit our website at the URL: http://www.mywebuniversity.com/Man_Pages/index.html to view and read the UNIX and Linux technical documentation online at any time you like. You need access to the internet, but you can

choose your browser and any device that you like, Yes, even your portable cell phone. No, you do not need to have the Operating Systems running on your device it is all running in our website.

$ whatis man

man - find and display reference manual pages

Examples:

1. man man
2. man systemctl
3. man systemd
4. man ufw
5. man curl
6. man wget
7. man -k ext4
8. man -k xfs
9. man -k bfs
10. man xfs
11. man banner

Please click on man man. to see the man pages for this command.

You are welcome to use any of those or simply go to the URL above and see alphabetical list of the UNIX and Linux commands in our website. Here I am showing the example eleven which is to run the “man banner” to find out what the “banner” command do.

The Direct Way to Linux Ubuntu

The output of your command "man banner" is:

BANNER(1) User's Reference Manual BANNER(1)

NAME
banner - print large banner

SYNOPSIS
banner text

DESCRIPTION
banner prints out the first 10 characters of text in large letters.

SEE ALSO
banner(6).

Debian February 4, 1997 BANNER(1)

In the two examples below, I am using the "**banner**" command to say **"Hello**" first and then on the second command I am using the "**banner"** command to say **"Hello, Ali"** as shown below:

The Direct Way to Linux Ubuntu

$ whatis mkdir

mkdir mkdir (1) - make directories

Examples:

To make the directory newdir you run the command below:

$ mkdir new

To make the three directories dir1/subdir-A/subdir-B all in one command, you run the command below:

$ mkdir -p dir1/subidr-A/subdir-B

Please click on man mkdir. to see the man pages for this command.

Here are some example of using **"mkdir**", **"rmdir"** and **"rm"** with **"-rf"** very dangerous if you are not careful, please don't use the **"rm -rf"** command since it forces **"f"** to remove recursively **"r"** the top directory and any subdirectories and files. You may inadvertently remove files or directories that you did not want to, therefore be extra careful to not make such a mistake especially when you are working with higher access privileges such as "sudo" or "root" access to the system. Even , if you are working on non-production system, please make sure to read your command very carefully before you executed.

I always check which hostname I am logged on, what is my current working directory and who I am logged in as before executing those commands. In the example, below I know for sure I am creating these directories for demonstration purposes, and it is Okay for me to remove them recursively provided that I do not make a typo 😊.

wlutfy@MyWebUniversity:~$ **hostname**
MyWebUniversity

wlutfy@MyWebUniversity:~$ **pwd**
/home/wlutfy

wlutfy@MyWebUniversity:~$ **id**
uid=1000(wlutfy) gid=1000(wlutfy) groups=1000(wlutfy)

I use the two command below to make the "newdir" and get a long list of the directory I just created.

wlutfy@MyWebUniversity:~$ **mkdir newdir**
wlutfy@MyWebUniversity:~$ **ls -ld newdir**
drwxrwxr-x 2 wlutfy wlutfy 4096 Nov 20 02:08 newdir

I try to make three directories but did not provide the "-p" option so the top directories does not exist at this time, and the output below will say no such file or directory.

wlutfy@MyWebUniversity:~$ **mkdir dir1/dir2/dir3**
mkdir: cannot create directory 'dir1/dir2/dir3': No such file or directory

This time, I provide the "-p" option and no matter how many subdirectories I provide, all of them are created successfully.

wlutfy@MyWebUniversity**:~$ mkdir -p dir1/dir2/dir3**
wlutfy@MyWebUniversity**:~$ ls -l dir1/dir2/dir3**
total 0
wlutfy@MyWebUniversity:~$ **ls -ld dir1/dir2/dir3**
drwxrwxr-x 2 wlutfy wlutfy 4096 Nov 20 02:08 dir1/dir2/dir3
wlutfy@MyWebUniversity:~$ **ls -l dir1/dir2**
total 4
drwxrwxr-x 2 wlutfy wlutfy 4096 Nov 20 02:08 dir3
wlutfy@MyWebUniversity**:~$ ls -l dir1/dir2**
total 4
drwxrwxr-x 2 wlutfy wlutfy 4096 Nov 20 02:08 dir3
wlutfy@MyWebUniversity:~$ **ls -l dir1/**
total 4
drwxrwxr-x 3 wlutfy wlutfy 4096 Nov 20 02:08 dir2

Now, I use the rmdir to remove the last directory "dir3" in this case.

wlutfy@MyWebUniversity:~$ **rmdir dir1/dir2/dir3**

I use the "ls -l" followed by the arguments I want to see the content of the directories I like to see.

wlutfy@MyWebUniversity:~$ **ls -l dir1/dir2**
total 0
wlutfy@MyWebUniversity:~$ **ls -l dir1/**
total 4
drwxrwxr-x 2 wlutfy wlutfy 4096 Nov 20 02:09 dir2

I want to recursively remove the "dir1" and all the sub-directories and files under the "dir1", so I am extra carefully that is what I want as I explained above. Please make sure you are also extra careful whenever you are removing files or directories that are no longer needed so that you remove what you don't need not more than that.

wlutfy@MyWebUniversity:~$ **rm -rf dir1**
wlutfy@MyWebUniversity:~$ **ls -ld newdir**
drwxrwxr-x 2 wlutfy wlutfy 4096 Nov 20 02:08 newdir
wlutfy@MyWebUniversity:~$ **ls -ld dir1**
ls: cannot access 'dir1': No such file or directory
wlutfy@MyWebUniversity:~$ **rmdir newdir**
wlutfy@MyWebUniversity:~$

As shown above I removed both the newdir and the dir1/dir2/dir3 which I no longer needed.

The Direct Way to Linux Ubuntu

$ whatis more

more, page - browse or page through a text file

Examples:

1. more /etc/os-release
2. more /etc/hosts

Please click on man more. to see the man pages for this command.

To see the content of the **"/etc/os-release**" file and if it is larger than one page, the **"more"** command will display one page at a time.

$ **more /etc/os-release**

NAME="Ubuntu"

VERSION="18.04.5 LTS (Bionic Beaver)"

ID=ubuntu

ID_LIKE=debian

PRETTY_NAME="Ubuntu 18.04.5 LTS"

VERSION_ID="18.04"

HOME_URL="https://www.ubuntu.com/"

SUPPORT_URL="https://help.ubuntu.com/"

BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"

PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"

VERSION_CODENAME=bionic

UBUNTU_CODENAME=bionic

The command “pg” like more will display the content of the file one page at a time, and if your Linux distribution does not come with it, you can either install the

The output of your command "pg /etc/os-release" is:

::::::::::::::
/etc/os-release
::::::::::::::
NAME="Ubuntu"
VERSION="18.04.5 LTS (Bionic Beaver)"
ID=ubuntu
ID_LIKE=debian
PRETTY_NAME="Ubuntu 18.04.5 LTS"
VERSION_ID="18.04"
HOME_URL="https://www.ubuntu.com/"
SUPPORT_URL="https://help.ubuntu.com/"
BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"
PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"
VERSION_CODENAME=bionic
UBUNTU_CODENAME=bionic

Below, I am showing the command “pg” that will display the larger file /etc/ntp.conf file one page at a time showing 48% of the content on the first page and then 97% and then the entire file. I pressed enter after each pause so I could continue to read the rest of the page, you can also type “**q**” for quit if you do not want to continue reading the rest of the pages.

$ pg /etc/ntp.conf
/etc/ntp.conf, configuration for ntpd; see ntp.conf(5) for help

driftfile /var/lib/ntp/ntp.drift

Leap seconds definition provided by tzdata
leapfile /usr/share/zoneinfo/leap-seconds.list

Enable this if you want statistics to be logged.
#statsdir /var/log/ntpstats/

statistics loopstats peerstats clockstats
filegen loopstats file loopstats type day enable
filegen peerstats file peerstats type day enable
filegen clockstats file clockstats type day enable

Specify one or more NTP servers.

```
# Use servers from the NTP Pool Project. Approved by Ubuntu Technical Board
# on 2011-02-08 (LP: #104525). See http://www.pool.ntp.org/join.html for
# more information.
pool 0.ubuntu.pool.ntp.org iburst
pool 1.ubuntu.pool.ntp.org iburst
pool 2.ubuntu.pool.ntp.org iburst
pool 3.ubuntu.pool.ntp.org iburst

# Use Ubuntu's ntp server as a fallback.
pool ntp.ubuntu.com

# Access control configuration; see /usr/share/doc/ntp-doc/html/accopt.html for
# details.  The web page <http://support.ntp.org/bin/view/Support/AccessRestrictions>
# might also be helpful.
#
# Note that "restrict" applies to both servers and clients, so a configuration
# that might be intended to block requests from certain clients could also end
# up blocking replies from your own upstream servers.

# By default, exchange time with everybody, but don't allow configuration.
restrict -4 default kod notrap nomodify nopeer noquery limited
restrict -6 default kod notrap nomodify nopeer noquery limited

# Local users may interrogate the ntp server more closely.
restrict 127.0.0.1
restrict ::1

# Needed for adding pool entries
restrict source notrap nomodify noquery

# Clients from this (example!) subnet have unlimited access, but only if
# cryptographically authenticated.
#restrict 192.168.123.0 mask 255.255.255.0 notrust


# If you want to provide time to your local subnet, change the next line.
# (Again, the address is an example only.)
#broadcast 192.168.123.255

# If you want to listen to time broadcasts on your local subnet, de-comment the
# next lines.  Please do this only if you trust everybody on the network!
#disable auth
#broadcastclient

#Changes recquired to use pps synchonisation as explained in documentation:
#http://www.ntp.org/ntpfaq/NTP-s-config-adv.htm#AEN3918
```

#server 127.127.8.1 mode 135 prefer   # Meinberg GPS167 with PPS
#fudge 127.127.8.1 time1 0.0042   # relative to PPS for my hardware

#server 127.127.22.1   # ATOM(PPS)
#fudge 127.127.22.1 flag3 1   # enable PPS API

The command "ping" will send "ICMP" echo request port 7 packets to the host or IP address provided. For instance, below we are ping google.com and mywebuniverisyt.com to see if they are alive and responding toe "ICMP" ECHO_REQUEST or not.

$ whatis ping

ping - send ICMP (ICMP6) ECHO_REQUEST packets to network hosts

Examples:

1. ping www.www.mywebuniversity.com
2. ping+ourunix.com
3. ping+sun.com
4. ping -s ourunix.com 56 4

Please click on man ping. to see the man pages for this command.

Now, let's send a count of 4 packages to google.com using the "ping" command.

wlutfy@MyWebUniversity:~$ **ping -c 4 google.com**

PING google.com (142.250.191.238) 56(84) bytes of data.

64 bytes from ord38s32-in-f14.1e100.net (142.250.191.238): icmp_seq=1 ttl=119 time=12.9 ms

64 bytes from ord38s32-in-f14.1e100.net (142.250.191.238): icmp_seq=2 ttl=119 time=13.0 ms

64 bytes from ord38s32-in-f14.1e100.net (142.250.191.238): icmp_seq=3 ttl=119 time=13.1 ms

64 bytes from ord38s32-in-f14.1e100.net (142.250.191.238): icmp_seq=4 ttl=119 time=12.9 ms

--- google.com ping statistics ---

4 packets transmitted, 4 received, 0% packet loss, time 3004ms

rtt min/avg/max/mdev = 12.941/13.027/13.154/0.140 ms

Now, let's send a count of 4 packages to www.mywebuniversity.com using the "ping" command.

wlutfy@MyWebUniversity:~$ **ping -c 4 www.mywebuniversity.com**

PING www.mywebuniversity.com (74.208.220.231) 56(84) bytes of data.

64 bytes from mywebuniversity.com (74.208.220.231): icmp_seq=1 ttl=64 time=0.041 ms

64 bytes from mywebuniversity.com (74.208.220.231): icmp_seq=2 ttl=64 time=0.049 ms

64 bytes from mywebuniversity.com (74.208.220.231): icmp_seq=3 ttl=64 time=0.048 ms

64 bytes from mywebuniversity.com (74.208.220.231): icmp_seq=4 ttl=64 time=0.049 ms

--- www.mywebuniversity.com ping statistics ---

4 packets transmitted, 4 received, 0% packet loss, time 3069ms

rtt min/avg/max/mdev = 0.041/0.046/0.049/0.009 ms

Once again, using the "ping" command is Okay on occasion to see if the host or IP address is reachable, but it should never be abused for any other purposes.

Yes, you can also use the IP address to ping an IP address if it is not in Domain Name Services "DNS" yet and no host assigned to the IP address, or for any other purpose you with to use the IP address instead

The Direct Way to Linux Ubuntu

$ whatis dig

dig dig (1m) - DNS lookup utility

Examples:

1. dig www.mywebuniversity.com
2. dig+ourunix.com
3. dig -h

Please click on man dig. to see the man pages for this command.

Example 3 above will produce the “help” or “-h” for the command “dig” as shown below:

The output of your command "dig -h" is:

```
Usage:  dig [@global-server] [domain] [q-type] [q-class] {q-opt}
        {global-d-opt} host [@local-server] {local-d-opt}
        [ host [@local-server] {local-d-opt} [...]]
Where:  domain     is in the Domain Name System
      q-class  is one of (in,hs,ch,...) [default: in]
      q-type   is one of (a,any,mx,ns,soa,hinfo,axfr,txt,...) [default:a]
            (Use ixfr=version for type ixfr)
      q-opt    is one of:
            -4                (use IPv4 query transport only)
            -6                (use IPv6 query transport only)
            -b address[#port]   (bind to source address/port)
            -c class          (specify query class)
            -f filename         (batch mode)
            -i                (use IP6.INT for IPv6 reverse lookups)
            -k keyfile          (specify tsig key file)
            -m                (enable memory usage debugging)
            -p port           (specify port number)
            -q name           (specify query name)
            -t type           (specify query type)
            -u                (display times in usec instead of msec)
            -x dot-notation     (shortcut for reverse lookups)
            -y [hmac:]name:key  (specify named base64 tsig key)
      d-opt    is of the form +keyword[=value], where keyword is:
            +[no]aaflag         (Set AA flag in query (+[no]aaflag))
            +[no]aaonly         (Set AA flag in query (+[no]aaflag))
            +[no]additional     (Control display of additional section)
            +[no]adflag         (Set AD flag in query (default on))
            +[no]all          (Set or clear all display flags)
            +[no]answer         (Control display of answer section)
            +[no]authority      (Control display of authority section)
            +[no]badcookie      (Retry BADCOOKIE responses)
            +[no]besteffort     (Try to parse even illegal messages)
            +bufsize=###        (Set EDNS0 Max UDP packet size)
            +[no]cdflag         (Set checking disabled flag in query)
            +[no]class          (Control display of class in records)
            +[no]cmd            (Control display of command line)
```

```
+[no]comments      (Control display of comment lines)
+[no]cookie        (Add a COOKIE option to the request)
+[no]crypto        (Control display of cryptographic fields in records)
+[no]defname       (Use search list (+[no]search))
+[no]dnssec        (Request DNSSEC records)
+domain=###        (Set default domainname)
+[no]dscp[=###]    (Set the DSCP value to ### [0..63])
+[no]edns[=###]    (Set EDNS version) [0]
+ednsflags=###     (Set EDNS flag bits)
+[no]ednsnegotiation (Set EDNS version negotiation)
+ednsopt=###[:value] (Send specified EDNS option)
+noednsopt         (Clear list of +ednsopt options)
+[no]expire        (Request time to expire)
+[no]fail          (Don't try next server on SERVFAIL)
+[no]header-only   (Send query without a question section)
+[no]identify      (ID responders in short answers)
+[no]idnout        (convert IDN response)
+[no]ignore        (Don't revert to TCP for TC responses.)
+[no]keepopen      (Keep the TCP socket open between queries)
+[no]mapped        (Allow mapped IPv4 over IPv6)
+[no]multiline     (Print records in an expanded format)
+ndots=###         (Set search NDOTS value)
+[no]nsid          (Request Name Server ID)
+[no]nssearch      (Search all authoritative nameservers)
+[no]onesoa        (AXFR prints only one soa record)
+[no]opcode=###    (Set the opcode of the request)
+[no]qr            (Print question before sending)
+[no]question      (Control display of question section)
+[no]rdflag        (Recursive mode (+[no]recurse))
+[no]recurse       (Recursive mode (+[no]rdflag))
+retry=###         (Set number of UDP retries) [2]
+[no]rrcomments    (Control display of per-record comments)
+[no]search        (Set whether to use searchlist)
+[no]short         (Display nothing except short
                   form of answer)
+[no]showsearch    (Search with intermediate results)
+[no]sigchase      (Chase DNSSEC signatures)
+[no]split=##      (Split hex/base64 fields into chunks)
+[no]stats         (Control display of statistics)
+subnet=addr       (Set edns-client-subnet option)
+[no]tcp           (TCP mode (+[no]vc))
+timeout=###       (Set query timeout) [5]
+[no]topdown       (Do +sigchase in top-down mode)
+[no]trace         (Trace delegation down from root [+dnssec])
+trusted-key=####  (Trusted Key to use with +sigchase)
+tries=###         (Set number of UDP attempts) [3]
+[no]ttlid         (Control display of ttls in records)
```

```
        +[no]ttlunits      (Display TTLs in human-readable units)
        +[no]unknownformat  (Print RDATA in RFC 3597 "unknown" format)
        +[no]vc            (TCP mode (+[no]tcp))
        +[no]zflag         (Set Z flag in query)
    global d-opts and servers (before host name) affect all queries.
    local d-opts and servers (after host name) affect only that lookup.
    -h                     (print help and exit)
    -v                     (print version and exit)
```

Here is an example for the hostname "**mywebuniversity.com**" using the default gateway to resolve the **A record** getting the "**IP address 74.208.220.231"** that is assign to the hostname "**mywebuniversity.com**" in this case.

$ dig 10.255.255.1 mywebuniversity.com

```
; <<>> DiG 9.11.3-1ubuntu1.14-Ubuntu <<>> 10.255.255.1 mywebuniversity.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 58331
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;10.255.255.1.                IN      A

;; Query time: 3 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: Sat Nov 20 02:56:10 EST 2021
;; MSG SIZE  rcvd: 41

;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 54486
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;mywebuniversity.com.          IN      A

;; ANSWER SECTION:
mywebuniversity.com.    0       IN      A       74.208.220.231

;; Query time: 0 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: Sat Nov 20 02:56:10 EST 2021
```

;; MSG SIZE rcvd: 64

Also, I can get the same information using the nameserver by finding out the nameserver first from the /etc/resolv.conf file and then using the "dig" command below to get the A record or IP address and hostname for "MyWebUniversity.com".

$ grep -i nameserver /etc/resolv.conf

nameserver 127.0.0.53

wlutfy@MyWebUniversity:~$ **dig 127.0.0.53 mywebuniversity.com**

; <<>> DiG 9.11.3-1ubuntu1.14-Ubuntu <<>> 127.0.0.53 mywebuniversity.com

;; global options: +cmd

;; Got answer:

;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 46017

;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:

; EDNS: version: 0, flags:; udp: 65494

;; QUESTION SECTION:

;127.0.0.53. IN A

;; Query time: 4 msec

;; SERVER: 127.0.0.53#53(127.0.0.53)

;; WHEN: Sat Nov 20 03:00:04 EST 2021

;; MSG SIZE rcvd: 39

;; Got answer:

;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 34891

;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:

; EDNS: version: 0, flags:; udp: 65494

;; QUESTION SECTION:

;mywebuniversity.com. IN A

;; ANSWER SECTION:

mywebuniversity.com. 0 IN A 74.208.220.231

;; Query time: 0 msec

;; SERVER: 127.0.0.53#53(127.0.0.53)

;; WHEN: Sat Nov 20 03:00:04 EST 2021

;; MSG SIZE rcvd: 64

wlutfy@MyWebUniversity:~$

$ whatis nslookup

nslookup - query Internet name servers interactively

Examples:

1. nslookup www.mywebuniversity.com
2. nslookup+ourunix.com

3. nslookup -q=mx www.mywebuniversity.com
4. nslookup -q=A www.mywebuniversity.com

Please click on man nslookup. to see the man pages for this command.

The output of your command "nslookup www.mywebuniversity.com" is:

Server: 127.0.0.53
Address:127.0.0.53#53

Non-authoritative answer:
Name: www.mywebuniversity.com
Address: 74.208.220.231

Below, we are using the Name Server Lookup command **"nslookup**" to get the query **"-q"** the A record so in this case we get the same IP address that was display at the output of the command "**dig"** and **"ping"** command.

The output of your command "nslookup -q=A www.mywebuniversity.com" is:

Server: 127.0.0.53
Address:127.0.0.53#53

Non-authoritative answer:
Name: www.mywebuniversity.com
Address: 74.208.220.231

In the examples below, I am using "nslookup" to query the "A record", the "Mail Exchanger mx" record as well as the default hostname and IP address when we do not provide any parameter than hostname.

wlutfy@MyWebUniversity:~$ **nslookup google.com**

Server: 127.0.0.53

Address: 127.0.0.53#53

Non-authoritative answer:

Name: google.com

Address: 142.250.191.238

Name: google.com

Address: 2607:f8b0:4009:81b::200e

wlutfy@MyWebUniversity:~$ **nslookup -q=mx google.com**

Server: 127.0.0.53

Address: 127.0.0.53#53

Non-authoritative answer:

google.com mail exchanger = 40 alt3.aspmx.l.google.com.

google.com mail exchanger = 10 aspmx.l.google.com.

google.com mail exchanger = 50 alt4.aspmx.l.google.com.

google.com mail exchanger = 30 alt2.aspmx.l.google.com.

google.com mail exchanger = 20 alt1.aspmx.l.google.com.

Authoritative answers can be found from:

wlutfy@MyWebUniversity:~$ **nslookup -q=A google.com**

Server: 127.0.0.53

Address: 127.0.0.53#53

Non-authoritative answer:

Name: google.com

Address: 142.250.191.238

$ whatis atrm

at(1), batch(1), atq(1), atrm(1) - queue, examine, or delete jobs for later execution

Examples:

First I use the command **' at'** to submit a job to run one minutes from now. In the example below the common two simple commands are '**pwd** and **ls -l'** just for demonstration purpose. You can run any valid UNIX commands or scripts instead of the two simple commands I used here. The syntax I used to run more than one commands from command line is the **HERE** syntax which in this case I used the end of file **'EOF**' after the characters '**<<**' to start my **HER**E, and '**EOF'** again to end my **HERE** syntax/documentation.

$ at now + 1 minute <<="" h3="">

>pwd

> ls -l

> EOF

commands will be executed using /bin/bash

job 1294552482.a at Sat Jan 8 21:54:42 2011

I was given the output above to tell me the job number for my at command and I run the command 'atq' to check the queue for my 'at' commands.

$ atq

Rank Execution Date Owner Job Queue Job Name

1st Jan 8, 2011 21:54 wlutfy 1294552482.a a stdin

I run the command 'atrm' to remove this new at command that I just submitted since I don't need to run

any commands and I just wanted to demo the 'atrm' command.

$ atrm 1294552482.a

1294552482.a: removed

$ atq

no files in queue.

Please click on man atrm. to see the man pages for this command.

The Direct Way to Linux Ubuntu

$ whatis atq

at(1), batch(1), atq(1), atrm(1) - queue, examine, or delete jobs for later execution

Examples:

First I use the command ' at' to submit a job to run one minutes from now. In the example below the common two simple commands are 'pwd and ls -l' just for demonstration purpose. You can run any valid UNIX commands or scripts instead of the two simple commands I used here. The syntax I used to run more than one commands from command line is the HERE syntax which in this case I used the end of file 'EOF' after the characters '<<' to start my HERE, and 'EOF' again to end my HERE syntax/documentation.

$ at now + 1 minute <<="" h3="">

>pwd

> ls -l

> EOF

commands will be executed using /bin/bash

job 1294552482.a at Sat Jan 8 21:54:42 2011

I was given the output above to tell me the job number for my at command and I run the command 'atq' to check the queue for my 'at' commands.

$ atq

Rank Execution Date Owner Job Queue Job Name

1st Jan 8, 2011 21:54 wlutfy 1294552482.a a stdin

I run the command 'atrm' to remove this new at command that I just submited since I don't need to run

any commands and I just wanted to demo the 'atrm' command.

$ atrm 1294552482.a

1294552482.a: removed

$ atq

no files in queue.

Please click on man atq. to see the man pages for this command.

The Direct Way to Linux Ubuntu

$ whatis sort

sort sort (1) - sort lines of text files

Examples:

1. sort -rn sortme.txt
2. cat sortme.txt | awk '{print $2,$1}' | sort -n
3. ls -l *.html | sort
4. ls -l *.html | sort -rn
5. ls -l *.html | sort

Please click on man sort. to see the man pages for this command.

Please feel free to practice any of the examples above. For instance, I am using the example number two and see the output of the command shown below. In this case, I am concatenate or “cat” the content of the file “sortme.txt” first, then I pipe “|” it to the command “awk” by printing only the second column which are the strings followed by the first column which are numbers and do sort them numerically by the string numeric values.

The output of your command "cat sortme.txt | awk '{print $2,$1}' | sort -n" is:

Japan 2011
91

Mortaza 10
MyWebuniversity.com 2000
Oracle 2010
OurUNIX.com 2009
Sadaf 7
Solaris 2010
Sun 1978
UNIX 1969
Wahid 5

The Direct Way to Linux Ubuntu

The output of your command "sort -rn sortme.txt" is:

2011 Japan EQ
2010 Solaris Express
2010 Oracle and Sun Merged
2009 OurUNIX.com
2000 MyWebuniversity.com
1978 Sun Microsystems
1969 UNIX AT&T
10 Mortaza
7 Sadaf

In the above example, I am using both "-rn" so that I do a numeric comparison with the "-r" option for the strings and sort them in reverse order with the "-r" option.

The Direct Way to Linux Ubuntu

$ whatis touch

touch(1) - change file access and modification times

Examples:

Before I demonstrate the touch command, let's identify what is my User ID, 'UID' and Group ID, 'GID'.

$ id

uid=101(wlutfy) gid=10(staff) groups=10(staff)

Here I am running the command below to verify that I am logged in as 'wlutfy'.

$ whoami

wlutfy

Then, I run the command below to find out the terinal (TTY), and the date and time I have logged in.

$ who am i

wlutfy pts/3 2011-01-09 17:44 (:0.0)

In the next two commands I am in the 'Test' directory and create an empty file 'newfile' with the timestamp of the current date and time of the system.

$ pwd

/export/home/wlutfy/Public/Test

$ touch newfile

I verify that the 'newfile' was zero byte in size and has the current system timestamp as it's creating date and time.

$ ls -ld newfile

-rw-r--r-- 1 wlutfy staff 0 2011-01-09 17:55 newfile

Using the '-t TIMESTAMP' option of the touch command, I change the timestamp of the file to 12:00

noon of January 1st, 2011.

$ touch -t 1101011200 newfile

I verify the change as shown below:

$ ls -ld newfile

-rw-r--r-- 1 wlutfy staff 0 2011-01-01 12:00 newfile

Now, I run the command 'date' to show today's date and time.

$ date

Sun Jan 9 17:56:10 PST 2011

Here, I change the timestamp of the 'ne wfile' to the current date and time timestamp.

$ touch newfile

I verfiy again the change and know the touch command worked as it was expected.

$ ls -ld newfile

-rw-r--r-- 1 wlutfy staff 0 2011-01-09 17:56 newfile

Please click on man touch. to see the man pages for this command.

$ whatis w

w - display information about currently logged-in users

Examples:

1. w
2. w -h
3. w -s
4. w -i

Please click on man w. to see the man pages for this command.

Please use any of the above examples on the our website to get the output. I also used the below commands to see the output of each at the Linux terminal.

$ **w --help**

Usage:

w [options]

Options:

-h, --no-header do not print header

-u, --no-current ignore current process username

-s, --short short format

-f, --from show remote hostname field

-o, --old-style old style output

-i, --ip-addr display IP address instead of hostname (if possible)

--help display this help and exit

-V, --version output version information and exit

For more details see w(1).

wlutfy@MyWebUniversity:~$ **w -s**

03:31:04 up 209 days, 23:56, 1 user, load average: 0.00, 0.01, 0.00

USER TTY FROM IDLE WHAT

wlutfy pts/0 47.150.234.108 0.00s w -s

wlutfy@MyWebUniversity:~$ **w -h**

wlutfy pts/0 47.150.234.108 22:44 4.00s 0.38s 0.01s w -h

wlutfy@MyWebUniversity:~$ **w -i**

03:31:11 up 209 days, 23:56, 1 user, load average: 0.00, 0.01, 0.00

USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT

wlutfy pts/0 47.150.234.108 22:44 7.00s 0.38s 0.01s w -i

wlutfy@MyWebUniversity:~$ **w**

03:31:13 up 209 days, 23:56, 1 user, load average: 0.00, 0.01, 0.00

USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT

wlutfy pts/0 47.150.234.108 22:44 0.00s 0.38s 0.01s w

The Direct Way to Linux Ubuntu

$ whatis who

who who (1) - show who is logged on

Examples:

1. who
2. who -r

Please click on man who. to see the man pages for this command.

The output of your command "who" is:

wlutfy pts/0 Nov 19 22:44 (47.150.234.108)

The output of your command "who -r" is:

run-level 5 Apr 24 04:35

The Direct Way to Linux Ubuntu

$ whatis chmod

chmod chmod (1) - change file mode bits

chmod chmod (1) - change the permissions mode of a file

chmod chmod (2) - change access permission mode of file

Examples:

The command 'chmod' is used to change the access permission for one or more files and directories. To better understand the change mode, 'chmod' command one better understand how the 'umask' comamnd and it is default settings works. As you can see from the commands below, I am logged in as user 'wlutfy' and the 'umask' value is '0022'. This means that any directories I create will have the permession of 'rwx' for owner, 'r-x' for group and 'r-x' for others. In addition, any files I create will have the permession of 'rw-' for owner 'wlutfy', 'r--' for group 'staff' and 'r--' for others. To better understande this, please see the tables below.

In the table below, I am showing the permession in both text format and octal values for a given permission. Please undertand that this permission settings can be used for owner, group and others. For instance, if one likes to make the owner 'rwx' octal value of '7' and 'rwx' for group octal value of '7', and 'rwx' for others octal value of '7', the the octal value equivalent to 'rwxrwxrwx', would be '777'. Bingo, with these three jackpot numbers '777' in UNIX you have full access permission.

rwx	**7**	**All permission are set for read, write and execute**
rw-	**6**	**Can read and write, but no executable accessr**
r-x	**5**	**Can read and excute but no wirte access.**
r--	**4**	**Can only read, but no write and excute access.**
-wx	**3**	**Can wirte and execute but have no read access. (?)**
-w-	**2**	**Can write, but have no read and execute access (?)**
--x	**1**	**Can only execute, but no read and write access.**
---	**0**	**No read, write and execute acesss at all**

Now, I am going to explain how these ocatal numbers are derived from the binary numbers. Since at the lowest level everything is represented as binaries numbers or bits of zeros '0' and ones '1' in the CPU registers. As you can see the three characters '---' have three positions from right to left, that is position zero, postion one and position two. Since binary numbers are base 2, we can easily find out the ocatal value fo each of the characters permission.

$2^2 = 4$ (read)

$2^1 = 2$ (write)

$2^0 = 1$(execute)

Let's follow some examples:

99

$ whoami

wlutfy

$ id

uid=101(wlutfy) gid=10(staff) groups=10(staff)

$ umask

0022

$ mkdir newdir

$ touch newfile

$ ls -ld newdir newfile

drwxr-xr-x 2 wlutfy staff 2 2011-01-15 21:09 newdir

-rw-r--r-- 1 wlutfy staff 0 2011-01-15 21:10 newfile

With the absense of '-' means the permission is set, otherwise the permission is not set. If al l permission is set, then we have rwx, which means 4 + 2 + 1 = 7, if we have rw-, then we have 4 + 2 + 0 = 6, and if permission is set to 'r-x', then we have 4 + 0 + 1 = 5. At this point you have a clear idea, but want to give one example of the 'rwxr-xr-x' which is eaquivalent to ocatal value of 755 or (4 + 2 + 1 = 7,4 + 0 + 1= 5, 4 + 0 + 1 = 5).

wnerShip	Group	Others	Octal Value
	rwx	**rwx**	**777**
	r--	**r--**	**644**
	r-x	**r-x**	**755**

	---	**---**	**400**
	rwx	**rwx**	**777**

Now we are ready to show some examples of the 'chmod' commands. I run 'pwd' , and 'ls -l' to verify the current permission of files and directory under the 'pkgdir' directory.

$ pwd

/export/home/wlutfy/training/OpenSolaris/pkgdir

As you can see below, when I created the directory 'newdir', the permission was created as 'drwxr-xr-x'. The 'd' character represents it is a directory, and the next nine character is equivalent to octal value of '755'. That is derived from the umask. The same formula is used for calculating the file permission, except the umask is subtracted from the octal value of '666' for new created files. For instance the new file I created using 'touch this', the permission is '-rw-r--r--', which is a regular text file with read, write permission for owner 'wlutfy' and read only permission for group 'staff' and others.

777 - 022 = 755 directory permission.

666 - 022 = 644 file permission.

$ ls -l

total 16

drwxr-xr-x 2 wlutfy staff 2 2011-01-15 21:09 newdir

-rw-r--r-- 1 wlutfy staff 0 2011-01-15 21:10 newfile

-rw-r--r-- 1 wlutfy staff 11258 2011-01-03 22:09 pkg-publisher

drwxr-xr-x 2 wlutfy staff 8 2011-01-15 20:02 text

-rw-r--r-- 1 wlutfy staff 0 2011-01-15 21:29 this

Now, I give full access (read, write, and excute) for the directory 'newdir' to everyone.

$ chmod 777 newdir

$ ls -ld newdir

drwxrwxrwx 2 wlutfy staff 2 2011-01-15 21:09 newdir

I change the directory permission for 'newdir' back to '755' which is read, write, execute for owner, read, and execute for group and others.

$ chmod 755 newdir

$ ls -ld newdir

drwxr-xr-x 2 wlutfy staff 2 2011-01-15 21:09 newdir

Now, I changed the permission mode of the three files below to read and write for owner and group and readable for others.

$ chmod 664 newfile pkg-publisher this

I list to verify the change I just made.

$ ls -l

total 16

drwxr-xr-x 2 wlutfy staff 2 2011-01-15 21:09 newdir

-rw-rw-r-- 1 wlutfy staff 0 2011-01-15 21:10 newfile

-rw-rw-r-- 1 wlutfy staff 11258 2011-01-03 22:09 pkg-publisher

drwxr-xr-x 2 wlutfy staff 8 2011-01-15 20:02 text

-rw-rw-r-- 1 wlutfy staff 0 2011-01-15 21:29 this

Now, I am going to make a simple script to show the permission

$ echo "df -hF zfs" > df-zfs.sh

$ ls -l df-zfs.sh

-rw-r--r-- 1 wlutfy staff 11 2011-01-17 13:01 df-zfs.sh

As you can see the permission of the script 'df-zfs.sh' is not executable. Therefore, when I tried to execute this script, it generated the error 'Permission denied' as shown below:

$ ./df-zfs.sh

bash: ./df-zfs.sh: Permission denied

Now, I use the born shell '-x' option to execute this script without changing the permission of the file. The '-x' script works like an interpreter, it displays every command with the plus '+' sign infront of it, and then if is is valid command or expression it display the result after it.

$ sh -x df-zfs.sh

$ df -hF zfs

Filesystem Size Used Avail Use% Mounted on

rpool/ROOT/opensolaris-1

20G 7.2G 12G 38% /

rpool/export 12G 22K 12G 1% /export

rpool/export/home 12G 22K 12G 1% /export/home

rpool/export/home/wlutfy

13G 771M 12G 6% /export/home/wlutfy

rpool 12G 81K 12G 1% /rpool

As shown below, I am changing the permission of this 'df-zfs.sh' script to executable for everyone so that I could run it as an executable script.

$ chmod 755 df-zfs.sh

$ ls -l df-zfs.sh

-rwxr-xr-x 1 wlutfy staff 11 2011-01-17 13:01 df-zfs.sh

Now, I am run it as an executable shell script.

$ ./df-zfs.sh

Filesystem Size Used Avail Use% Mounted on

rpool/ROOT/opensolaris-1

20G 7.2G 12G 38% /

rpool/export 12G 22K 12G 1% /export

rpool/export/home 12G 22K 12G 1% /export/home

rpool/export/home/wlutfy

13G 771M 12G 6% /export/home/wlutfy

rpool 12G 81K 12G 1% /rpool

Please click on man chmod. to see the man pages for this command.