SILIGURI INSTITUTE OF TECHNOLOGY

LABPRATORY MANUAL

PROGRAMMING WITH PYTHON

SILIGURI INSTITUTE OF TECHNOLOGY

VISON

Siliguri Institute of Technology is To be a recognized institution offering high quality education, opportunities to students to become globally employable Engineers/Professionals in best ranked industries and research organization.

MISSION

To impart quality technical education for holistic development of students who will full fil the needs of the industry/society and be actively engaged in making a successful career in industry/research/higher education in India & abroad

PROGRAM EDUCATIONAL OBJECTIVES (PEO) :

The graduates will be:

- Competent professionals with knowledge of Computer Science & Engineering to pursue variety of careers/higher education.
- Proficient in successfully designing innovative solutions to real life problems that are technically sound, economically viable and socially acceptable.
- Efficient team leaders, effective communicators and capable of working in multi-disciplinary environment following ethical values.
- Capable of adapting to new technologies and constantly upgrade their skills with an attitude towards lifelong learning.

PROGRAM OUTCOMES (PO)

Engineering Graduates will be able to:

- Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

- Problem analysis: Identify, formulate, review research literature, and analyze complexengineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
- Design/development of solutions: Design solutions for complex engineering problems anddesign system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
- Conduct investigations of complex problems: Use research-based knowledge and researchmethods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
- Modern tool usage: Create, select, and apply appropriate techniques, resources, and modernengineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
- The engineer and society: Apply reasoning informed by the contextual knowledge to assesssocietal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
- Environment and sustainability: Understand the impact of the professional engineering solutionsin societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
- Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms ofthe engineering practice.
- Individual and team work: Function effectively as an individual, and as a member or leader indiverse teams, and in multidisciplinary settings.
- Communication: Communicate effectively on complex engineering activities with the engineeringcommunity and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
- Project management and finance: Demonstrate knowledge and understanding of theengineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
- Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadestcontext of technological change.

Programming with Python :

Course Objective

CO1:Student will able to install, set pat variable of Python 2.7 versions and write, test, and debug simple Python programs.
CO2:To implement Python programs with conditionals and loops.
CO3:Use functions for structuring Python programs.
CO4:Represent compound data using Python lists, tuples, dictionaries.
CO5: Importing module ,Read and write data from/to files in Python.

LABORATORY

Maulana Abul Kalam Azad University of Technology,
(Formerly West Bengal University of Technology) West Bengal

Syllabus for B. Tech in Information Technology
(Applicable from the academic session 2018-2019)
Subject Code : PCC-CS 393
Category: Professional Core course
Subject Name : IT Workshop (Sci Lab/MATLAB/Python/R)
Semester : Third L-T-P : 1-0-3 Credit:3

Pre-Requisites: No-prerequisite

Programming with Python

Introduction :

History, Features ,Setting up path, Working with Python, Basic Syntax, Variable and Data Type, Operator

Conditional Operator :

If , if-else, Nested if-else, looping : For, While, Nested loops

Control Statements:

Break, Continue, pass

String Manipulation:

Accessing String Basic Operations, String slices, Function and Methods.

List :

Introduction, Accessing list, Operations, Working with lists, Function and Methods.

Tuple :

Introduction, Accessing tuples, Operations, Working, Functions and Methods.

Dictionary:

Introduction, Accessing values in dictionaries, Working with dictionaries, Properties

Function :

Defining a function, Calling a function, Types of functions, Function Arguments, Anonymous functions, Global and local variables.

Module :

Importing module, Math module, Random module, Packages, Composition, Input-Output

Printing on screen, Reading data from keyboard, Opening and closing file, Reading and writing files, Functions.

Exception Handling:

Exception, Exception Handling, Except clause, Try ? finally clause, User Defined Exceptions.

Experiment NO.	Topic	Title
1	BASIC	A).Install Python and Set Path variable
		B). Running instructions in Interactive interpreter and a Python Script
		C). Write a program to purposefully raise Indentation Error and Correct it. [Display your name and Department in two separate line]
2	Operator	A). Write a program to compute distance between two points taking input from the user (Pythagorean Theorem)
		B). Write a program add.py that takes 2 numbers as command line arguments and prints its sum.
3	Conditional Statement	A) Write a program to check a given number is even or odd.
		B) Write a program to check a given year is leap year or not.
		C) Write a program to calculate real roots of a quadratic equation.
4	Loop Statement	A) Write a program using a while loop that asks the user for a number, and prints a countdown from that number to zero. [using range() method]
		B) Write a program to calculate the Sum of even Fibonacci numbers below 4 Thousand.
		C) Write a program to calculate GCD of two number.
		D) Write a program to print the following pattern : i) ii) iii)

i)

*		
*	*	
*	*	*

ii) iii)

		*		

	*	*	*	
*	*	*	*	*

*	*	*
	*	*
		*

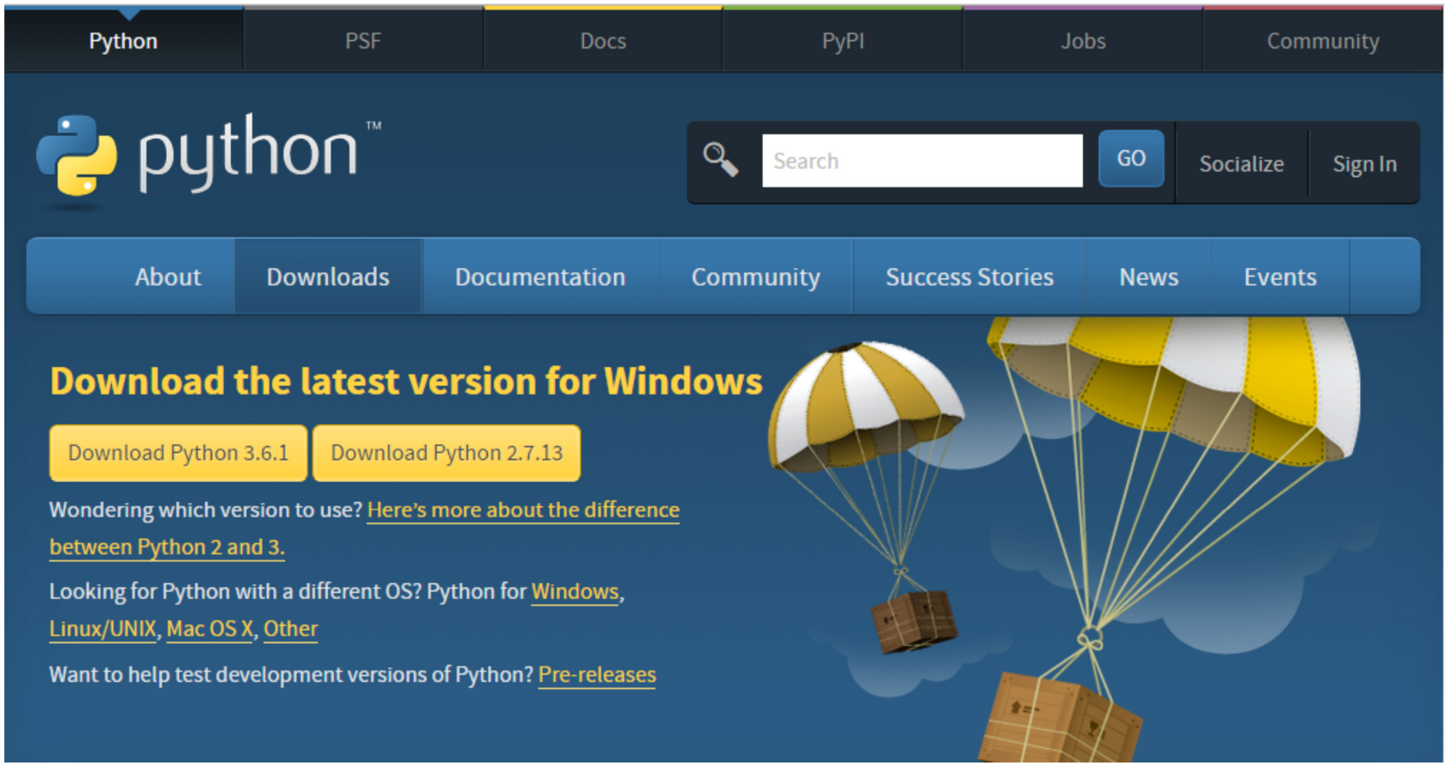
5	String Operation	A) Write a program to count no of vowel in a string(using in operator)
		B) Write a program to perform following operation on strong: i) The total number of characters in the string ii) The last three characters of the string iii) Print The string backwards direction iv) Print The string in all caps
6	Tuple and Set	A)Write a program to initialize and Display tuple data structure. B) Write a program to initialize and Display two Set data structure and do the following operation : i) union ii)intersection ii)difference
7	List	A) Print the total number of items in the list
		B) Print the list in reverse order.
		C) Remove the first and last items from the list, sort the remaining items, and print the result.
		D) Write a program that generates a list of 20 random numbers between 1 and 100. (a) Print the list. (b) Print the average of the elements in the list. (c) Print the largest and smallest values in the list. (d) Print the second largest and second smallest entries in the list (e) Print how many even numbers are in the list.
		E) Write a program that takes any two lists L and M of the same size and adds their elements together to form a new list N whose elements are sums of the corresponding elements in L and M. For instance, if L=[3,1,4] and M=[1,5,9], then N should equal [4,6,13].
		F)Write a program to perform multiplication of two square matrices
8	Function	A) Write a function called rectangle that takes two integers m and n as arguments and prints out an m n box consisting of asterisks. Shown below is the output of rectangle(2,4). **** ****
		B) Write a function called sum_digits() that is given an integer num and returns the sum of the digits of num.
		c)The digital root of a number n is obtained as follows: Add up the digits n to get a new number. Add up the digits of that to get another new number. Keep doing this until you get a number that has only one digit. That number is the digital root. For example, if n = 45893, we add up the digits to get 4 + 5 + 8 + 9 + 3 = 29. We then add up the digits of 29 to get 2 + 9 = 11. We then add up the digits of 11 to get 1 + 1 = 2. Since 2 has only one digit, 2 is our digital root. Write a function that returns the digital root of an integer n. [Note: there is a shortcut, where the digital root is equal to n mod 9, but do not use that here.]
		C)Write a program to multiply two list using lambda function
		D) Write a program to filter out only odd number from a list.
9	Dictionary	A)Write a Python script to store(ascending and descending order) in to a dictionary by value.
		B)Write a Python script to insert a new key in to a dictionary.
		C)Write a program to take a list of student's (name, age, marks) input from key board. Print average marks and details of highest scorer using dictionary data structure.
10	File	A)Write a program to copy the content of one file in to another file.
		B)Write a program to count the frequency of each word from a file.
11	Module	A)Write a program to display i)Current date and time ii)Current year iii)Month of year iv)Week number of year v)Week day of the week vi)Day of year vii)Day of week
		B)Plot the the roll number and average marks of a list of student in a class(import mathplotlib module)

12	Exception Handling	A)Write a program to tame two number as a input and divide theme and show i) value error ii)zero division error

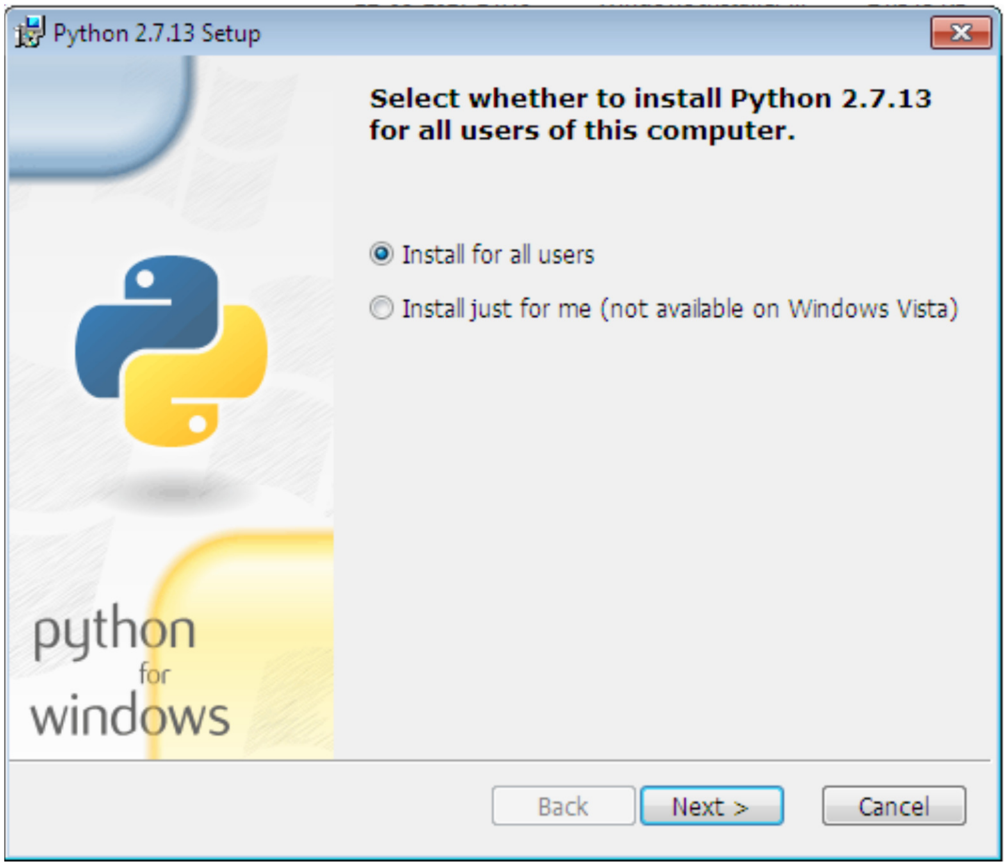
Experiment 1:

Procedure to Install and Run programs in Python:

In order to install python, Visit https://www.python.org. When we visit the Python for Windows download page, we will immediately see the division. Right at the top, square and center, the repository asks if you want the latest release of Python 2 or Python 3 (2.7.13 and 3.6.1, respectively) as shown in below Figure.


Python
PSF
Docs
PyPI
Jobs
Community
python™
Search
GO
Socialize
Sign In
About
Downloads
Documentation
Community
Success Stories
News
Events
Download the latest version for Windows
Download Python 3.6.1
Download Python 2.7.13
Wondering which version to use? Here's more about the difference between Python 2 and 3.
Looking for Python with a different OS? Python for Windows, Linux/UNIX, Mac OS X, Other
Want to help test development versions of Python? Pre-releases

The version we want depends on our end goal. Here we will install Python 2.7.13. Click on Download Python 2.7.13 then python-2.7.13.msi file will be downloaded. Run the installer, then a window will be opened as shown below. Select “Install for all users,” and then click “Next”.



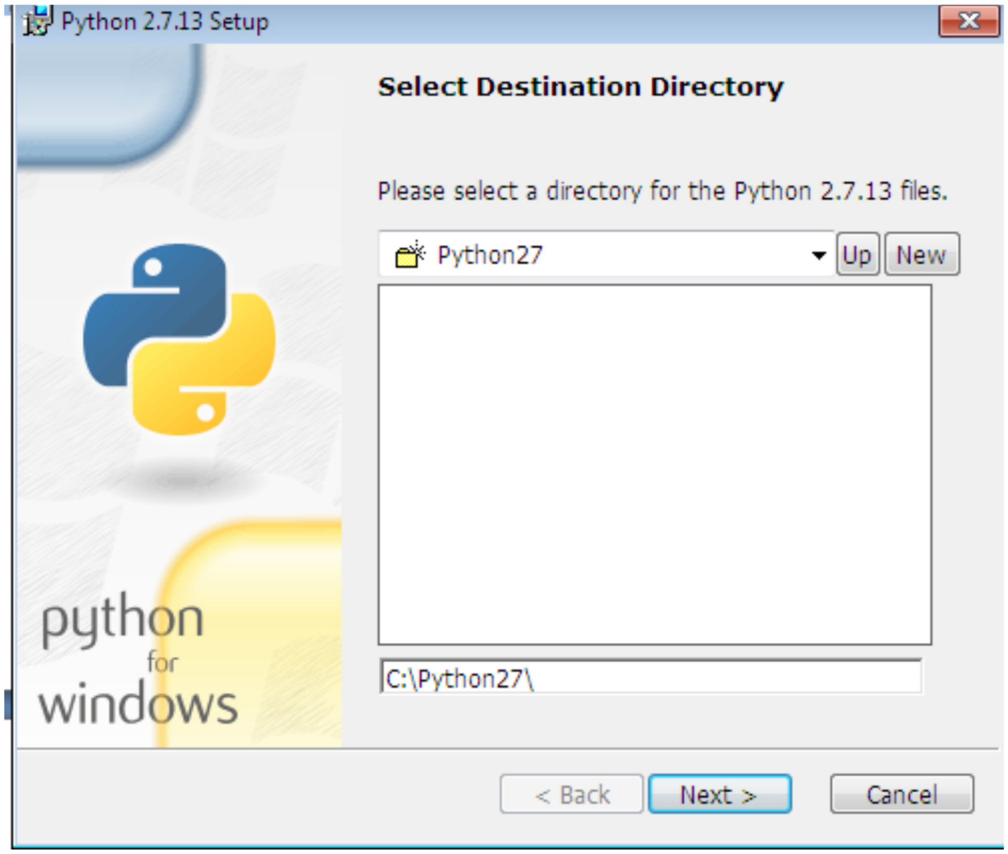
Python 2.7.13 Setup

Select whether to install Python 2.7.13 for all users of this computer.

(●) Install for all users

( ) Install just for me (not available on Windows Vista)

After Clicking on "Next", a window will be opened as shown below. On the directory selection screen, leave the directory as "Python27" and click "Next".



Python 2.7.13 Setup

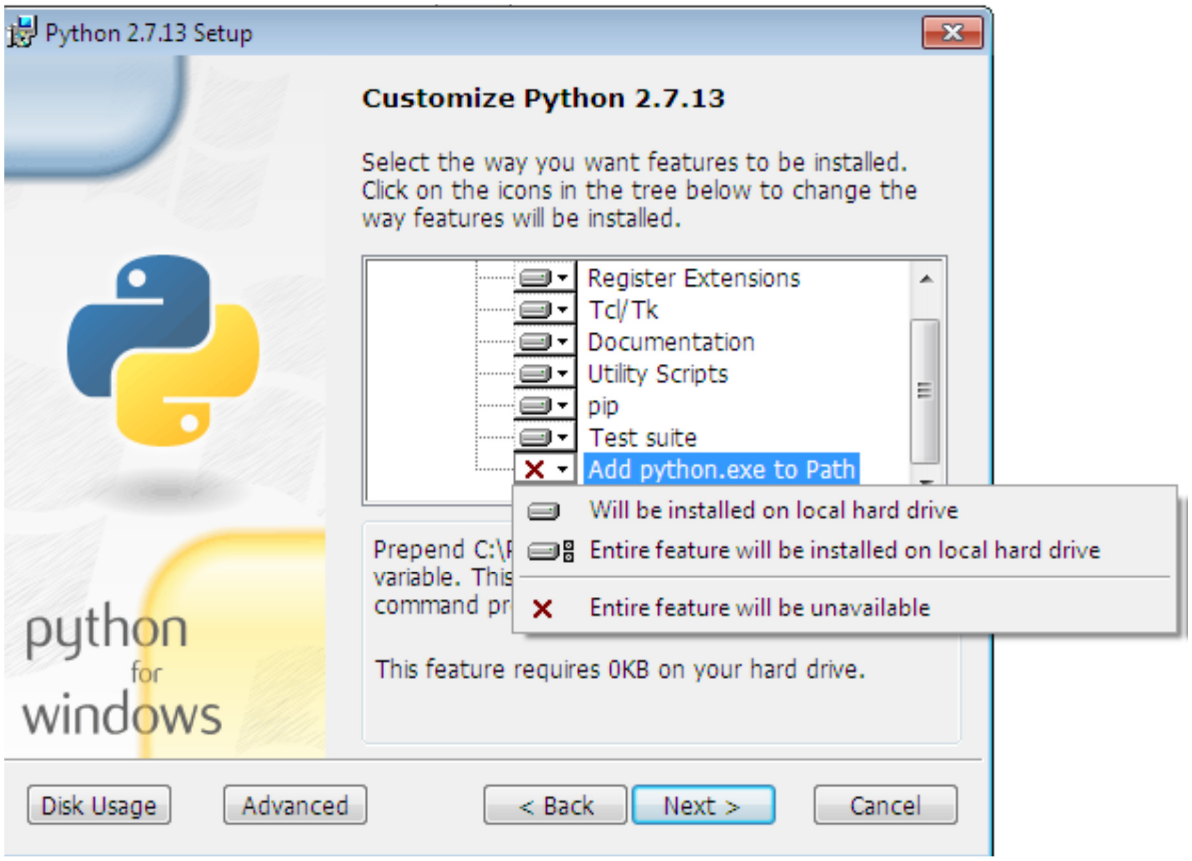
Select Destination Directory

Please select a directory for the Python 2.7.13 files.

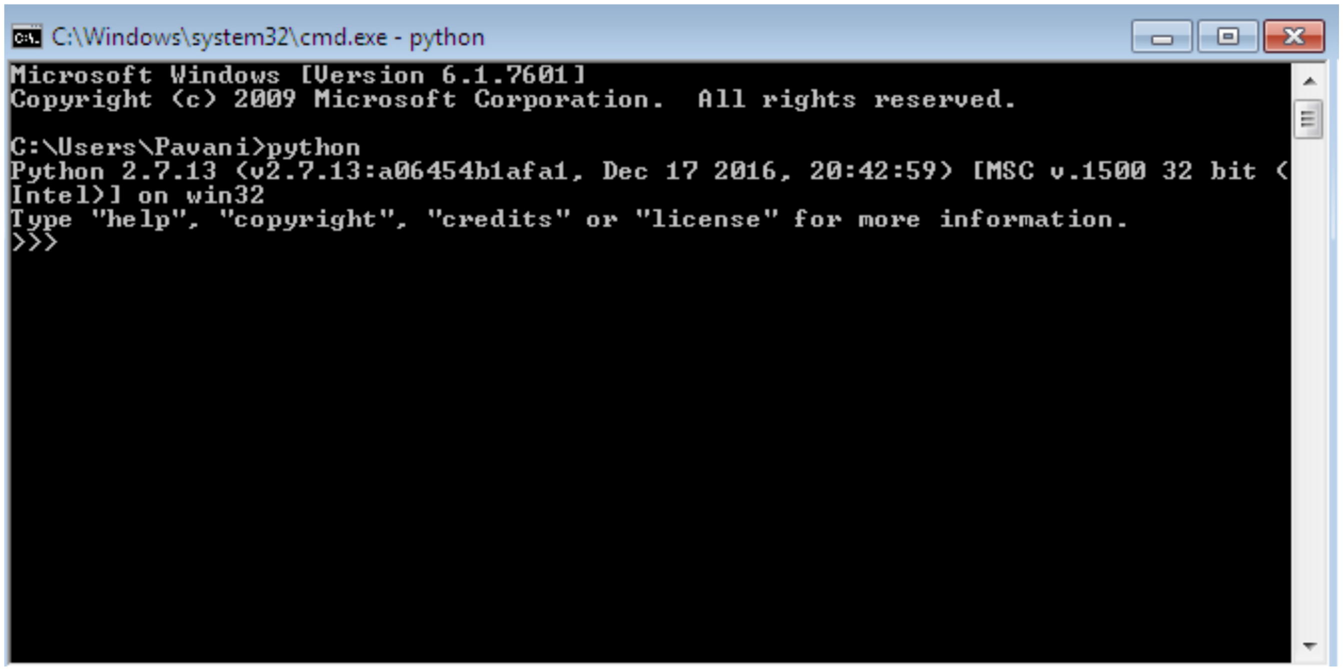
Python27

C:\Python27\

After Clicking on “Next”, a window will be opened as shown below. On the customization screen, scroll down, click “Add python.exe to Path,” and then select “Will be installed on local hard drive.” then click “Next.”


Python 2.7.13 Setup
Customize Python 2.7.13
Select the way you want features to be installed.
Click on the icons in the tree below to change the way features will be installed.
Register Extensions
Tcl/Tk
Documentation
Utility Scripts
pip
Test suite
Add python.exe to Path
Will be installed on local hard drive
Entire feature will be installed on local hard drive
Entire feature will be unavailable
Prepend C:\P
variable. This
command pr
This feature requires 0KB on your hard drive.
python for windows
Disk Usage
Advanced
< Back
Next >
Cancel

We don‟t have to make any more decisions after this point. Just click through the wizard to complete the installation. When the installation is finished, set the variable path. After setting up the path, we can confirm the installation by opening up Command Prompt and type the following command as shown below.



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\Pavani>python
Python 2.7.13 (v2.7.13:a06454b1afa1, Dec 17 2016, 20:42:59) [MSC v.1500 32 bit (
Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

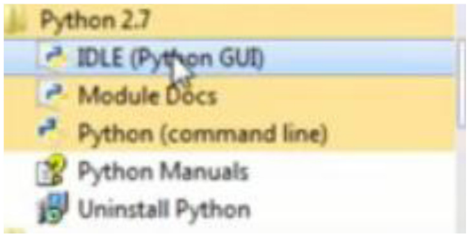
Now, we can say that Python 2.7.13 is installed on our machine.

Different Ways of Invoking Python:

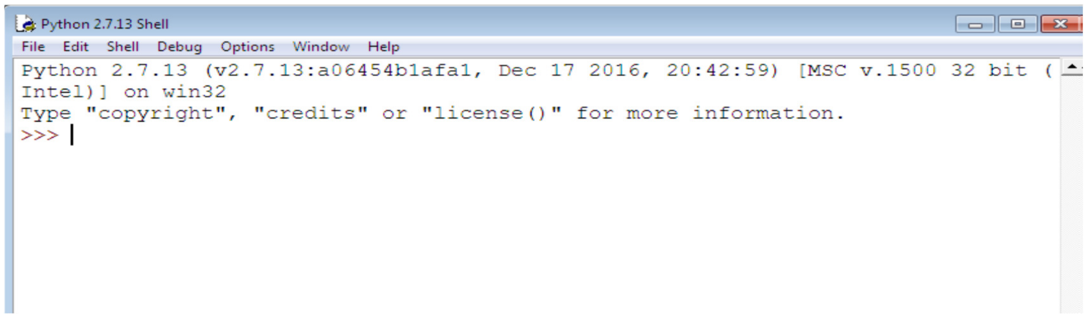
- Python GUI
- Python command line
- Command prompt from windows

Python GUI:

Click on start -> all programs -> python 2.7 -> IDLE(Python GUI).



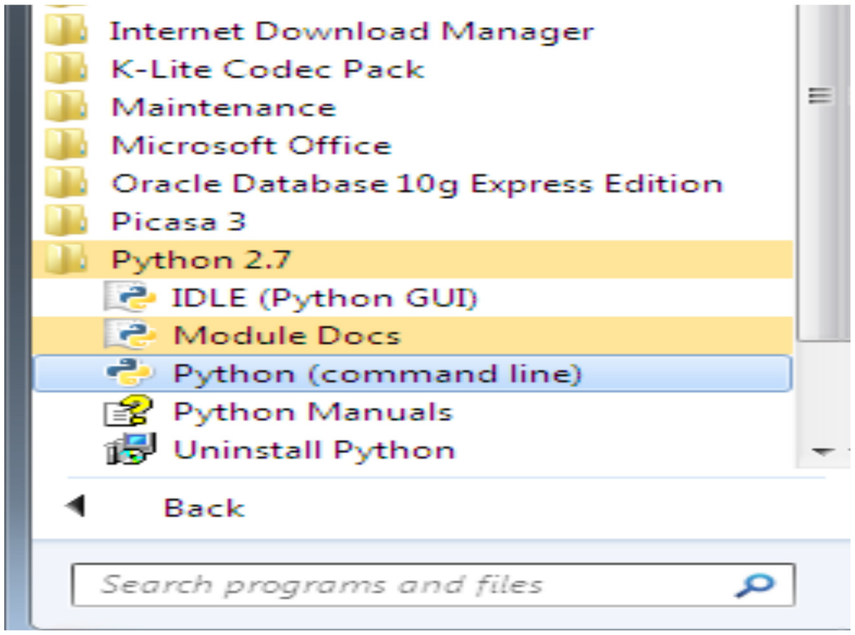
After Clicking on IDLE(Python GUI), a window will be opened as shown below. Python command line: Click on



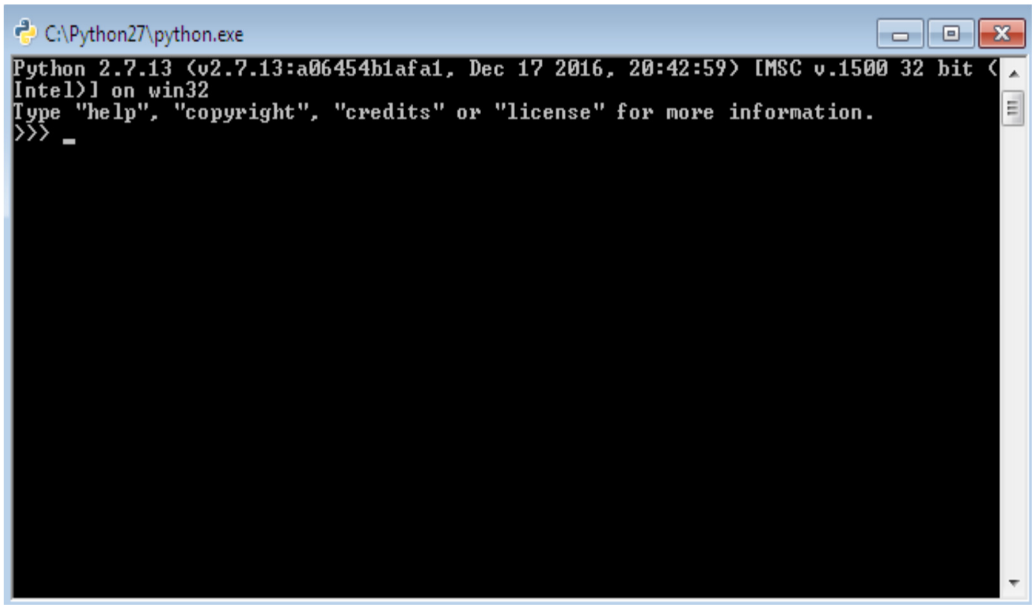
```
Python 2.7.13 (v2.7.13:a06454b1afa1, Dec 17 2016, 20:42:59) [MSC v.1500 32 bit (
Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> |
```

Python command line:

Click on start -> all programs -> python 2.7 -> Python (Command line).



After Clicking on Python (command line), a window will be opened as shown below:

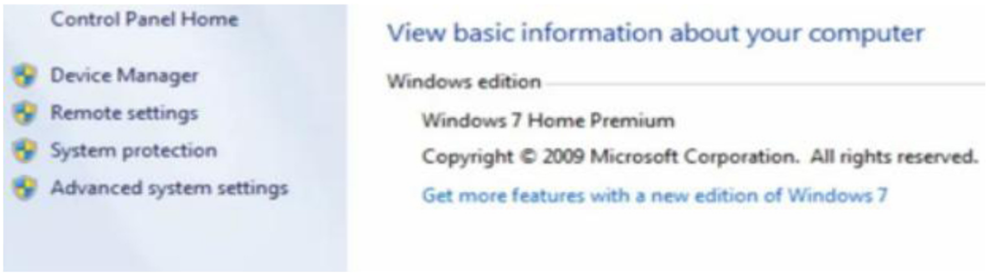


```
Python 2.7.13 (v2.7.13:a06454b1afa1, Dec 17 2016, 20:42:59) [MSC v.1500 32 bit (
Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> _
```

Command prompt from windows:

To open Python from Windows command prompt, We need to **set path**. The procedure to set the path is as follows :

Go to My Computer -> right click and open properties, then a window will be opened as shown below:



View basic information about your computer

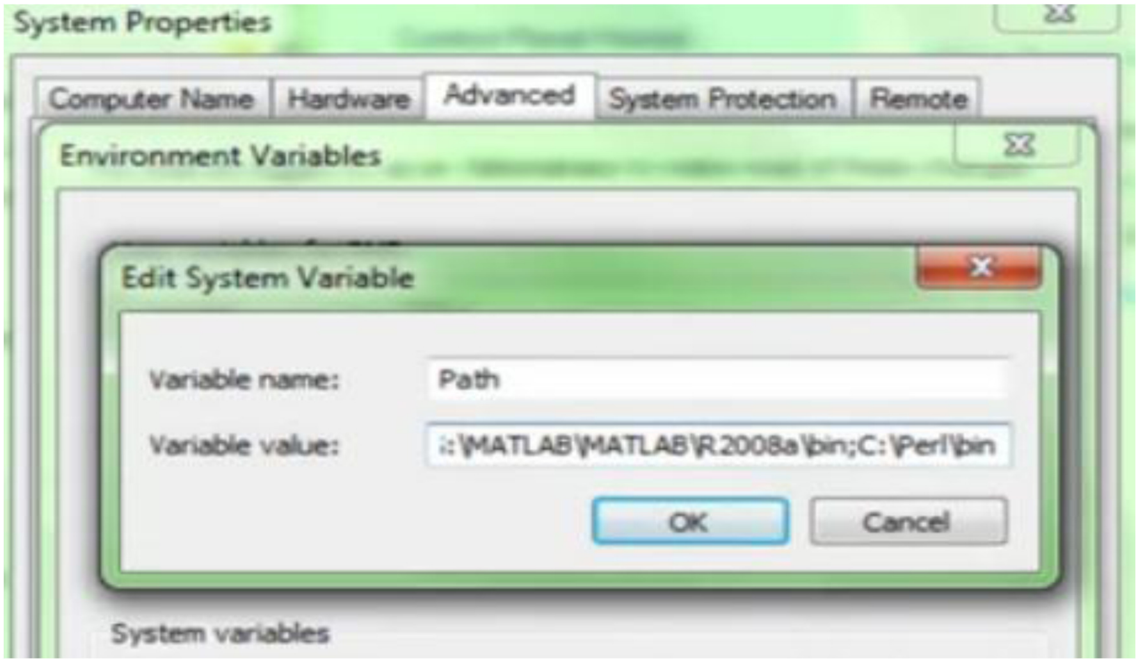
Windows edition

Windows 7 Home Premium

Copyright © 2009 Microsoft Corporation. All rights reserved.

Get more features with a new edition of Windows 7

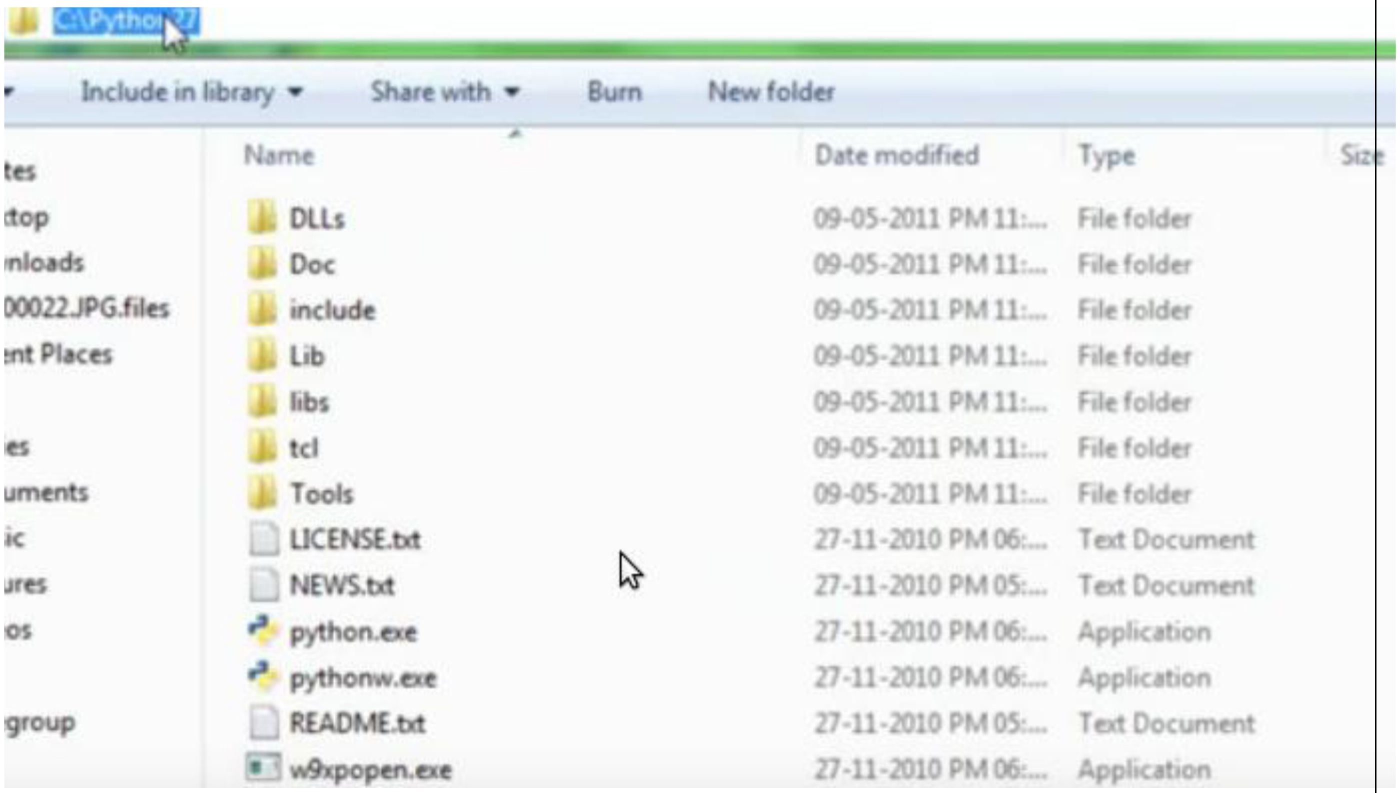
Now, Click on Advanced system settings -> Environmental Variables -> system variables and under system variable, click on Path variable and click on Edit. Then, a window will be opened as follows:



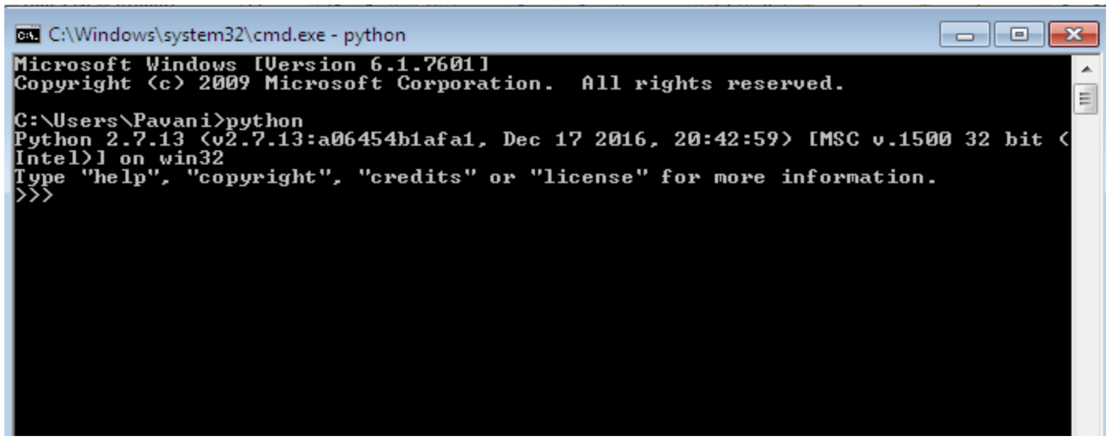
Variable name: Path

Variable value: :\MATLAB\MATLAB\R2008a\bin;C:\Perl\bin

Add python path in variable value and click on **OK** as follows:



Now Open Command prompt from windows (cmd), and type the command "python" as follows:



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\Pavani>python
Python 2.7.13 (v2.7.13:a06454b1afa1, Dec 17 2016, 20:42:59) [MSC v.1500 32 bit (
Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Experiment : 1(C)

Write a program to purposefully raise Indentation Error and Correct it.

Description:

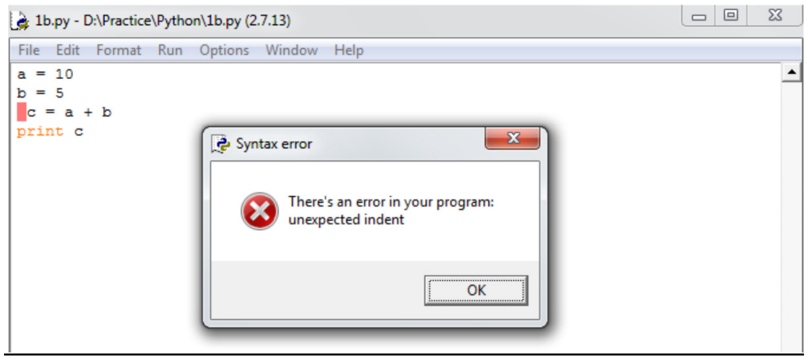
Most of the programming languages like C, C++, Java use braces { } to define a block of code. Python uses indentation.

A code block (body of a function, loop etc.) starts with indentation and ends with the first unintended line. The amount of indentation is depends on our choice, but it must be consistent throughout that block. Generally, Four whitespaces are used for indentation and is preferred over tabs. The enforcement of indentation in Python makes the code look neat and clean. This results into Python programs that look similar and consistent. Incorrect indentation will result into Indentation Error.

Program that shows Indentation Error:

```
a = 10
b = 5
   c = a + b
 print c
```

Output:



1b.py - D:\Practice\Python\1b.py (2.7.13)

```
a = 10
b = 5
 c = a + b
print c
```

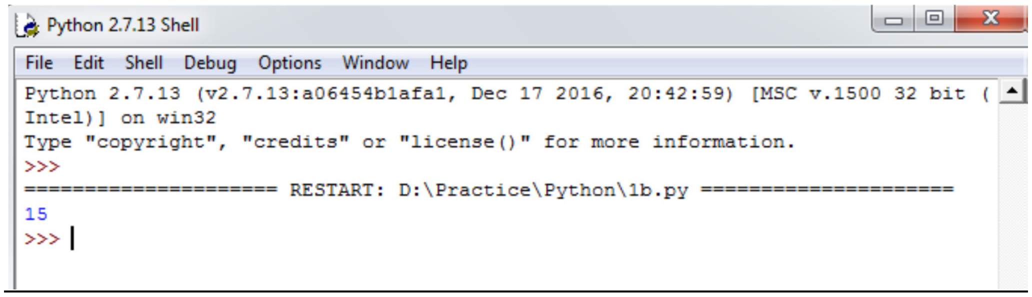
Syntax error

There's an error in your program:
unexpected indent

OK

Program without Indentation Error:

```
a = 10
b = 5
c = a + b
print c
```



Python 2.7.13 Shell

```
Python 2.7.13 (v2.7.13:a06454b1afa1, Dec 17 2016, 20:42:59) [MSC v.1500 32 bit (
Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> 
===================== RESTART: D:\Practice\Python\1b.py =====================
15
>>> |
```

Experiment 2(A) :

Write a program to compute distance between two points taking input from the user (Pythagorean Theorem).

Description: The Pythagorean theorem is the basis for computing distance between two points. Let (x_1,y_1) and (x_2,y_2) be the co-ordinates of points on xy-plane. From Pythagorean theorem, the distance between two points is calculated using the formulae:

$$\text{Distance D}= \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Distance between two point A(x1,y1) and B(x2,y2)

```
import math as m

print(" Enter the Co ordinate of first point")
x1=int(input())
y1=int(input())
```

```
print("Enter the co ordinate of second point")
x2,y2=[int (x) for x in input("Enter x and y seperated by space").split()]
print(x2,y2)
d=m.sqrt((x2-x1)*(x2-x1) + (y2-y1)**2)
print(m.ceil(d))
print(m.floor(d))
```

# 2_B) Program for Arithmatic operation like addition ,multiplication ,division ....

```
'''
x=int(input("Enter first number"))
y=int(input("enter second number"))
print(" Addition result=",x-y)
print(" Multiplication result=",x*y)
print(" Division result=",x/y)
print(" modular result=",x%y)
print(" integer division result=",x//y)

'''
```

#3_A) Check given number is even or odd...

```
'''

x=int(input("Enter The number"))
if (x%2==0):
 print("The Number",x, "is Even")
else:
 print("The Number",x,"is Odd")

'''
```

#3_B) Check given yuear is leap year or not...

```
'''
x=int(input("Enter The Year"))
if(x%400==0 or x%100!=0 and x%4==0 ):
 print("The year",x,"is leap year")
else:
 print("The Year",x,"Is not leap year")

'''
```

#3_C Check a given charactor is alphabet ,digit or specal char or not

```
ch=input("enter any symbol from keybord")
```

```
print("Ascii value of the Symbol is",ord(ch))
if((ch>='A' and ch<='Z') or (ch>='a' and ch<='z')):
  print("The Symbol",ch,"is alphabet")
elif(ch>='0' and ch<='9'):
  print("The symbol",ch,"is Digit")
else:
  print("The symbol",ch,"Is Special charactor")
```

# 4_A print serasse of number and count down of this number .

```
'''
x=int(input("Enter a number "))
y=x
while(x>0):
   print(x)
   x=x-1
# countdown using range method

for i in range(y,0,-1):
   print(i)
```

# 4_B Sum of even fibo nacci number below 4000

```
a=-1
b=1
s=0
c=a+b
while(c<=4000):
   a=b
   b=c
   c=a+b
   if(c%2==0):
        print(c)
         s=s+c
print("Sum of all even fibo nacci bello 4000 is =",s)
```

# 4_c GCD of two number

```
x=int(input("Enter First number"))
y=int(input("enter second number"))
m=x
n=y
while(x!=y):
  if(x>y):
    x=x-y
  else:
    y=y-x
print(" Gcd of ",m, "and",n,"is=",x)
```

# 4d_patt_i Print pattaen :

```
x=int(input("Enter no of row"))
for i in range(0,x+1):
  for j in range(0,i+1,1):
    if(i>j):
        print("*",end='')
    else:
        print()
```

# 4d_patt_ii Print pattaen :

```
x=int(input("Enter no of row"))
for i in range(0,x):
  for k in range(0,x-i+1):
    print(" ",end="")
  for j in range(0,i+1):
    print(" * ",end='')
  print()
```

# 4d_patt_iii Print pattaen :

```
x=int(input("Enter no of row"))
for i in range(0,x):
  for k in range(0,i+1):
    print(" ",end="")
  for j in range(0,x-i):
    print("*",end='')
  print()
```

'''
# 5 A write a program to count no of vowel in a srting ...
'''

```
s=input("Enter a string")
s=s.lower()
c=0
for item in s:
  if item in ('a','e','i','o','u'):
    c=c+1
print(" Total vowel = ",c)
```

```
'''

# 5_B write a program to perform the following operation in a srting ...
s=input("Enter a string")
print(" Total no of charactor is=",len(s))
print(" last 3 charactor is=",s[-3:])
print(" Reverse String is = ",s[-1::-1])
print(" All capital of string= ",s.upper())


'''
# 6_A_Initialize and display tuple and set

t=(1,2,"Kritt",'a',1,'d',23.7)
# in set all element are unique do not contain duplicate element(automatic delete)
s={ "kritt",3,5,9,23.0,3}

print(t)
print(s)

'''
# 6_B_Initialize and display two  set and do the following operation

# initialize empty set
s1=set()
s2=set()
while (1):

  item=input("Enter set item for set 1: ")
  s1.add(item)
  print(" Press 1  for continue and For quit press 0 ")
  ch=int(input())
  if ch==0:
    break
  else:
    continue

print(" item of first  set = :",s1)
while (1):
  item=input("Enter set item for set 2: ")
  s2.add(item)
  print("Press 1  for continue and For quit press 0")
  ch=int(input())
  if ch==0:
     break
   else:
     continue

print("Total item of First SET = ",s1,"Total item of Second  set = :",s2)

print("Union of SET 1 and SET 2 is=: ",s1.union(s2))
#print(s1|s2)
print("Intersection of SET 1 and SET 2 is =: ",s1.intersection(s2))
#print(s1&s2)
print("Difference of SET 1 and SET 2 is =: ",s1.difference(s2))
#print(s1-s2)

# list Operation ....
# initialize a list
'''
L=[1,2,8,4,3]

#insert a item in to list

item=int(input(" Enter item for insert .."))
L.append(item)
print(L)

#count no of item in the list
c=len(L)
print("Total no of element is =",c)
print(" Print in reverse order",L[-1::-1])
L.remove(L[0])
print(L)
L.pop()
print("After remove last element",L)
#print decending order
L.sort(reverse=True)
print(" Sotred order",L)
```

# Program 7_D generate 20 random number between 1 to 100

```
import random
L=[]
for i in range(5):
   x=random.randint(1,100)
   L.append(x)
print(L)
```

```
m=max(L)
n=min(L)
s=sum(L)
a=s/len(L)
print(" max=",m,"min=",n,"sum=",s,"Average=",a)
L.sort()
print(L)
print("Second Highest =",L[-2])
print("Second lowest=",L[1])
c=[i for i in L if i%2==0]
print("No of Even =",len(c))
```

#Program 7_E Take two list and add them store in to third list

```
import random

L=[]
M=[]

for i in range(5):
  x=random.randint(1,100)
  L.append(x)
print(L)
for i in range(5):
  x=random.randint(1,100)
  M.append(x)
print(M)

N=[ 0 for i in range(5)]
for j in range(5) :
  N[j]=L[j]+M[j]
print("Addition od Two list =",N)
'''
# multiply two matrox...
M=[[1,2,3],[1,1,2],[2,2,1]]
N=[[1,2,1],[1,1,2],[1,2,1]]
R=[]
for i in range(3):
  l=[]
  for j in range(3):
    l.append(0)
    R.append(l)
print(M,N,R)
```

```
for i in range(3):
  for j in range(3):
    for k in range(3):
        R[i][j]=R[i][j]+M[i][k]*N[k][j]
print(" Matrix Result ")
for i in R:
  print(i)

'''
```

# 8_A program print rectangle with * using function..

```
def rect(r,c):
  for i in range(r):
    for j in range(c):
      print("*",end='')
                                                                        print()


print("Enter no of row")
r=int(input())
print("Enter no of collumn")
c=int(input())
rect(r,c)
```

# 8_B program print some of digit using function..

```
def sum_dig(n):
  s=0
  while(n>0):
    r=n%10
    s=s+r
    n=n//10
  return(s)

print(" Enter the number ")
n=int(input())

print("SUM of digit if a number",n," is =",sum_dig(n))

# 8_C program finding digital root  using function.

def sum_dig(n):
  s=0
  while(n>0):
      r=n%10
      s=s+r
      n=n//10
   return(s)

print(" Enter the number ")
n=int(input())
sod=sum_dig(n)
while(sod>9):
   sod=sum_dig(sod)
   print(sod)
print(" Digital root is =",sod)
'''
```

# 8_D program add two list using lambda function (both list must be in same in size.

```
L1=[1,3,2,5]
L2=[3,5,1,2]
res=map(lambda x,y:x*y,L1,L2)

print(list(res))

# lambda with filter function
L=[2,1,4,5,8,9,23.10,3]
res1=list(filter(lambda x :(x%2==1),L))
print("Odd number in list  ")
print(res1)
```

SILIGURI INSTITUTE OF TECHNOLOGY

DEPARTMENT
OF INFORMATION TECHNOLOGY

COMPUTER NETWORK

LABORATORY MANUAL

LM: PCC-CS692

1.0 Network Cabling

1.1 What is Network Cabling?

Cable is the medium through which information usually moves from one network device to another. There are several types of cable which are commonly used with LANs. In some cases, a network will utilize only one type of cable, other networks will use a variety of cable types. The type of cable chosen for a network is related to the network's topology, protocol, and size. Understanding the characteristics of different types of cable and how they relate to other aspects of a network is necessary for the development of a successful network.

The following sections discuss the types of cables used in networks and other related topics.

1. Unshielded Twisted Pair (UTP) Cable
2. Shielded Twisted Pair (STP) Cable
3. Coaxial Cable
4. Fiber Optic Cable
5. Wireless LANs
6. How to wire Ethernet Cables

    - What you need?
    - About the cable
    - About RJ45 plugs & Jacks
    - Ethernet cable pin out
    - How to wire Ethernet patch cable
    - Cable category details

7. Cat 5 cables
8. RJ45 Connectors

1.1.1. Unshielded Twisted Pair (UTP) Cable

Twisted pair cabling comes in two varieties: shielded and unshielded. Unshielded twisted pair (UTP) is the most popular and is generally the best option for school networks (See fig. 1).

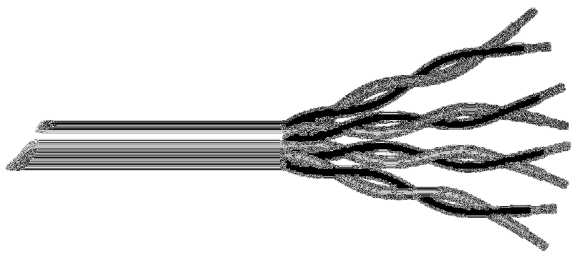


Fig.1. Unshielded twisted pair

The quality of UTP may vary from telephone-grade wire to extremely high-speed cable. The cable has four pairs of wires inside the jacket. Each pair is twisted with a different number of twists per inch to help eliminate interference from adjacent pairs and other electrical devices. The tighter the twisting, the higher the supported transmission rate and the greater the cost per foot. The EIA/TIA (Electronic Industry Association/Telecommunication Industry Association) has established standards of UTP and rated five categories of wire.

Categories of Unshielded Twisted Pair

Type	Use
Category 1	Voice Only (Telephone Wire)
Category 2	Data to 4 Mbps (LocalTalk)
Category 3	Data to 10 Mbps (Ethernet)
Category 4	Data to 20 Mbps (16 Mbps Token Ring)
Category 5	Data to 100 Mbps (Fast Ethernet)

Buy the best cable you can afford; most schools purchase Category 3 or Category 5. If you are designing a 10 Mbps Ethernet network and are considering the cost savings of buying Category 3 wire instead of Category 5, remember that the Category 5 cable will provide more "room to grow" as transmission technologies increase. Both Category 3 and Category 5 UTP have a maximum segment length of 100 meters. In Florida, Category 5 cable is required for retrofit grants. 10BaseT refers to the specifications for unshielded twisted pair cable (Category 3, 4, or 5) carrying Ethernet signals. Category 6 is relatively new and is used for gigabit connections.

Unshielded Twisted Pair Connector

The standard connector for unshielded twisted pair cabling is an RJ-45 connector. This is a plastic connector that looks like a large telephone-style connector (See fig. 2). A slot allows the RJ-45 to be inserted only one way. RJ stands for Registered Jack, implying that the connector follows a standard borrowed from the telephone industry. This standard designates which wire goes with each pin inside the connector.

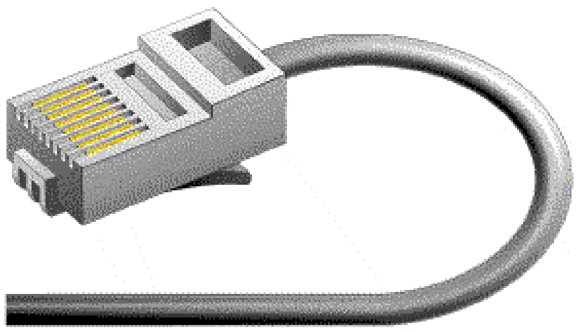


Fig. 2. RJ-45 connector

1.1.2. Shielded Twisted Pair (STP) Cable

A disadvantage of UTP is that it may be susceptible to radio and electrical frequency interference. Shielded twisted pair (STP) is suitable for environments with electrical interference; however, the extra shielding can make the cables quite bulky. Shielded twisted pair is often used on networks using Token Ring topology.

1.1.3. Coaxial Cable

Coaxial cabling has a single copper conductor at its center. A plastic layer provides insulation between the center conductor and a braided metal shield (See fig. 3). The metal shield helps to block any outside interference from fluorescent lights, motors, and other computers.

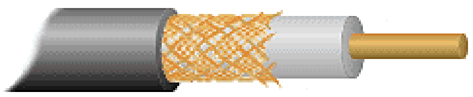


Fig. 3. Coaxial cable

Although coaxial cabling is difficult to install, it is highly resistant to signal interference. In addition, it can support greater cable lengths between network devices than twisted pair cable. The two types of coaxial cabling are thick coaxial and thin coaxial.

Thin coaxial cable is also referred to as thinnet. 10Base2 refers to the specifications for thin coaxial cable carrying Ethernet signals. The 2 refers to the approximate maximum segment length being 200 meters. In actual fact the maximum segment length is 185 meters. Thin coaxial cable is popular in school networks, especially linear bus networks.

Thick coaxial cable is also referred to as thicknet. 10Base5 refers to the specifications for thick coaxial cable carrying Ethernet signals. The 5 refers to the maximum segment length being 500 meters. Thick coaxial cable has an extra protective plastic cover that helps keep moisture away from the center conductor. This makes thick coaxial a great choice when running longer lengths in a linear bus network. One disadvantage of thick coaxial is that it does not bend easily and is difficult to install.

Coaxial Cable Connectors

The most common type of connector used with coaxial cables is the Bayone-Neill-Concelman (BNC) connector (See fig. 4). Different types of adapters are available for BNC connectors, including a T-connector, barrel connector, and terminator. Connectors on the cable are the weakest points in any network. To help avoid problems with your network, always use the BNC connectors that crimp, rather than screw, onto the cable.

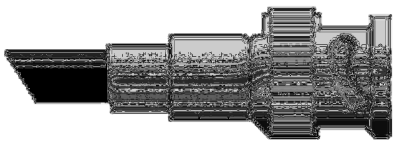


Fig. 4. BNC connector

1.1.4. Fiber Optic Cable

Fiber optic cabling consists of a center glass core surrounded by several layers of protective materials (See fig. 5). It transmits light rather than electronic signals eliminating the problem of electrical interference. This makes it ideal for certain environments that contain a large amount of electrical interference. It has also made it the standard for connecting networks between buildings, due to its immunity to the effects of moisture and lighting.

Fiber optic cable has the ability to transmit signals over much longer distances than coaxial and twisted pair. It also has the capability to carry information at vastly greater speeds. This capacity broadens communication possibilities to include services such as video conferencing and interactive services. The cost of fiber optic cabling is comparable to copper cabling; however, it is more difficult to install and modify. 10BaseF refers to the specifications for fiber optic cable carrying Ethernet signals.

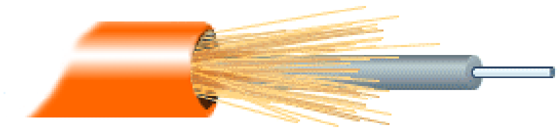


Fig.5. Fiber optic cable

Facts about fiber optic cables:

- Outer insulating jacket is made of Teflon or PVC.
- Kevlar fiber helps to strengthen the cable and prevent breakage.
- A plastic coating is used to cushion the fiber center.
- Center (core) is made of glass or plastic fibers.

Fiber Optic Connector

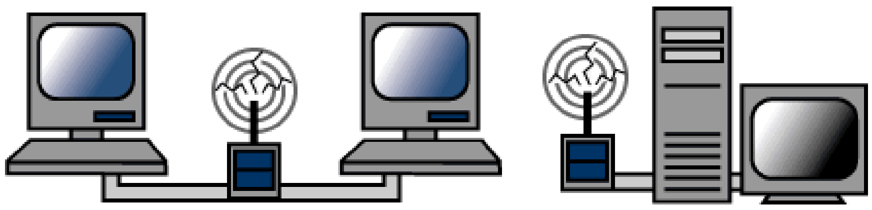
The most common connector used with fiber optic cable is an ST connector. It is barrel shaped, similar to a BNC connector. A newer connector, the SC, is becoming more popular. It has a squared face and is easier to connect in a confined space.

Ethernet Cable Summary

Specification	Cable Type	Maximum length
10BaseT	Unshielded Twisted Pair	100 meters
10Base2	Thin Coaxial	185 meters
10Base5	Thick Coaxial	500 meters
10BaseF	Fiber Optic	2000 meters
100BaseT	Unshielded Twisted	100 meters

	Pair	
100BaseTX	Unshielded Twisted Pair	220 meters

1.1.5. Wireless LANs



Not all networks are connected with cabling; some networks are wireless. Wireless LANs use high frequency radio signals, infrared light beams, or lasers to communicate between the workstations and the file server or hubs. Each workstation and file server on a wireless network has some sort of transceiver/antenna to send and receive the data. Information is relayed between transceivers as if they were physically connected. For longer distance, wireless communications can also take place through cellular telephone technology, microwave transmission, or by satellite.

Wireless networks are great for allowing laptop computers or remote computers to connect to the LAN. Wireless networks are also beneficial in older buildings where it may be difficult or impossible to install cables.

The two most common types of infrared communications used in schools are line-of-sight and scattered broadcast. Line-of-sight communication means that there must be an unblocked direct line between the workstation and the transceiver. If a person walks within the line-of-sight while there is a transmission, the information would need to be sent again. This kind of obstruction can slow down the wireless network.

Scattered infrared communication is a broadcast of infrared transmissions sent out in multiple directions that bounces off walls and ceilings until it eventually hits the receiver. Networking communications with laser are virtually the same as line-of-sight infrared networks.

Wireless LANs have several disadvantages. They provide poor security, and are susceptible to interference from lights and electronic devices. They are also slower than LANs using cabling.

1.1.6. How to wire Ethernet Cables

What You Need:

Required:

- Cable - bulk Category 5, 5e, 6 or 7 cable
- Wire Cutters - to cut and strip the cable if necessary

For Patch Cables:

- RJ45 Plugs
- RJ45 Crimper

For Fixed Wiring:

- RJ45 Jacks
- 110 Punch Down Tool

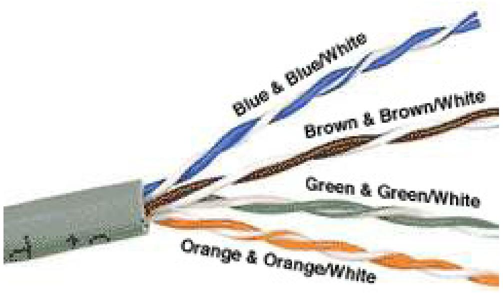
Recommended:

- Wire Stripper
- Cable Tester

About the Cable:

You can find bulk supplies of the cable at many computer stores or most electrical or home centers. You want UTP (Unshielded Twisted Pair) Category 5 cable for basic 10/100 functionality. You want CAT 5e for gigabit (1000BaseT) operation and CAT 6 or 7 gives you a measure of future proofing. Bulk cable comes in many types, there are 2 basic categories, solid and braided cable. Braided cable tends to work better in patch applications for desktop use. It is more flexible and resilient than solid cable and easier to work with, but really meant for shorter lengths. Solid cable is meant for longer runs in a fixed position. Plenum rated cable must be used whenever the cable travels through an air circulation space. For example, above a false ceiling or below a raised floor. It may be difficult or impossible to tell from the package what type of cable it is, so peal out an end and investigate.

Here is what the internals of the cable look like:



Internal Cable Structure and Color Coding

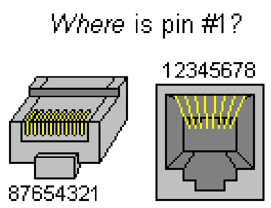
Inside the cable, there are 8 color coded wires. These wires are twisted into 4 pairs of wires, each pair has a common color theme. One wire in the pair being a solid or primarily solid colored wire and the other being a primarily white wire with a colored stripe (Sometimes cables won't have any color on the striped wire, the only way to tell which is which is to check which wire it is twisted around). Examples of the naming schemes used are: Orange (alternatively Orange/White) for the solid colored wire and White/Orange for the striped cable. The twists are extremely important. They are there to counteract noise and interference. It is important to wire according to a standard to get proper performance from the cable. The TIA/EIA-568-A specifies two wiring standards for an 8-position modular connector such as RJ45. The two wiring standards, T568A and T568B vary only in the arrangement of the colored pairs. Tom writes to say "...sources suggest using T568A cabling since T568B is the AT&T standard, but the US Government specifies T568A since it matches USOC cabling for pairs 1 & 2, which allows it to work for 1/2 line phones...". Your choice might be determined by the need to match existing wiring, jacks or personal preference, but you should maintain consistency. I've shown both below for straight through cabling and just T568B for cross over cabling.

About RJ45 Plugs and Jacks:

The RJ45 plug is an 8-position modular connector that looks like a large phone plug. There are a couple variations available. The primary variation you need to pay attention to is whether the connector is intended for braided or solid wire. For braided/stranded wires, the connector has sharp pointed contacts that actually pierce the wire. For solid wires, the connector has fingers which cut through the insulation and make contact with the wire by grasping it from both sides. The connector is the weak point in an ethernet cable, choosing the wrong one will often cause grief later. If you just walk into a computer store, it's nearly impossible to tell what type of plug it is. You may be able to determine what type it is by crimping one without a cable.

RJ45 jacks come in a variety styles intended for several different mounting options. The choice is one of requirements and preference. RJ45 jacks are designed to work only with solid cable. Most jacks come labeled with color codes for either T568A, T568B or both. Make sure you end up with the correct one.

Here is a diagram and pin out:



RJ45 Plug and Jack Pin Out

Ethernet Cable Pin Outs:

There are two basic cable pin outs. A straight through cable, which is used to connect to a hub or switch, and a cross over cable used to operate in a peer-to-peer fashion without a hub/switch. Generally all fixed wiring should be run as straight through. Some ethernet interfaces can cross and un-cross a cable automatically as needed, a handy feature.

Standard, Straight-Through Wiring (both ends are the same):

RJ45 Pin #	Wire Color (T568A)	Wire Diagram (T568A)	10Base-T Signal 100Base-TX Signal	1000Base-T Signal
1	White/Green	▨	Transmit+	BI_DA+
2	Green	🟩	Transmit-	BI_DA-
3	White/Orange	▨	Receive+	BI_DB+
4	Blue	🟩	Unused	BI_DC+
5	White/Blue	▨	Unused	BI_DC-
6	Orange	🟩	Receive-	BI_DB-
7	White/Brown	▨	Unused	BI_DD+
8	Brown	🟩	Unused	BI_DD-

Straight-Through Cable Pin Out for T568A

RJ45 Pin #	Wire Color (T568B)	Wire Diagram (T568B)	10Base-T Signal 100Base-TX Signal	1000Base-T Signal
1	White/Orange	▨	Transmit+	BI_DA+

RJ45 Pin #	Wire Color (T568B)	Wire Diagram (T568B)	10Base-T Signal 100Base-TX Signal	1000Base-T Signal
2	Orange	🟩	Transmit-	BI_DA-
3	White/Green	▨	Receive+	BI_DB+
4	Blue	🟩	Unused	BI_DC+
5	White/Blue	▨	Unused	BI_DC-
6	Green	🟩	Receive-	BI_DB-
7	White/Brown	▨	Unused	BI_DD+
8	Brown	🟩	Unused	BI_DD-

Straight-Through Cable Pin Out for T568B

Cross Over Cable (T568B):

RJ45 Pin # (END 1)	Wire Color	Diagram End #1	RJ45 Pin # (END 2)	Wire Color	Diagram End #2
1	White/Orange	▨	1	White/Green	▨
2	Orange	🟩	2	Green	🟩
3	White/Green	▨	3	White/Orange	▨
4	Blue	🟩	4	White/Brown	▨
5	White/Blue	▨	5	Brown	🟩
6	Green	🟩	6	Orange	🟩
7	White/Brown	▨	7	Blue	🟩
8	Brown	🟩	8	White/Blue	▨

Cross Over Cable Pin Outs

+Note: The cross over cable layout is suitable for 1000Base-T operation, all 4 pairs are crossed.

How to wire Ethernet Patch Cables:

1. Strip off about 2 inches of the cable sheath.
2. Untwist the pairs - don't untwist them beyond what you have exposed, the more untwisted cable you have the worse the problems you can run into.
3. Align the colored wires according to the diagrams above.
4. Trim all the wires to the same length, about 1/2" to 3/4" left exposed from the sheath.
5. Insert the wires into the RJ45 plug - make sure each wire is fully inserted to the front of the RJ45 plug and in the correct order. The sheath of the cable should extend into the RJ45 plug by about 1/2" and will be held in place by the crimp.
6. Crimp the RJ45 plug with the crimper tool.
7. Verify the wires ended up the right order and that the wires extend to the front of the RJ45 plug and make good contact with the metal contacts in the RJ45 plug
8. Cut the cable to length - make sure it is more than long enough for your needs.
9. Repeat the above steps for the second RJ45 plug.

How to wire fixed Ethernet Cables:

1. Run the full length of cable in place, from endpoint to endpoint, making sure to leave excess.
2. At one end, cut the wire to length leaving enough length to work, but not too much excess.
3. Strip off about 2 inches of the cable sheath.

4. Align each of the colored wires according to the layout of the jack.
5. Use the punch down tool to insert each wire into the jack.
6. Repeat the above steps for the second RJ45 jack.

If a cable tester is available, use it to verify the proper connectivity of the cable. That should be it, if your cable doesn't turn out, look closely at each end and see if you can find the problem. Often a wire ended up in the wrong place or one of the wires is making no contact or poor contact. Also double check the color coding to verify it is correct. If you see a mistake or problem, cut the end off and start again. A cable tester is invaluable at identifying and highlighting these issues.

When sizing cables remember that an end to end connection should not extend more than 100m (~328ft). Try to minimize the cable length, the longer the cable becomes, the more it may affect performance. This is usually noticeable as a gradual decrease in speed and increase in latency.

Notes:

Power over Ethernet (PoE):

Power over Ethernet has been implemented in many variations before IEEE standardized 802.3af. 802.3af specifies the ability to supply an endpoint with 48V DC at up 350mA or 16.8W. The endpoint must be capable of receiving power on either the data pairs [Mode A] (often called phantom power) or the unused pairs [Mode B] in 100Base-TX. PoE can be used with any ethernet configuration, including 10Base-T, 100Base-TX and 1000Base-T. Power is only supplied when a valid PoE endpoint is detected by using a low voltage probe to look for the PoE signature on the endpoint. PoE power is typically supplied in one of two ways, either the host ethernet switch provides the power, or a "midspan" device is plugged in between the switch and endpoints and supplies the power. No special cabling is required.

RJ45 Pin #	Wire Color (T568A)	Wire Diagram (T568A)	10Base-T Signal 100Base-TX Signal	PoE
1	White/Green	▨	Transmit+	Mode A +
2	Green	🟩	Transmit-	Mode A +
3	White/Orange	▨	Receive+	Mode A -
4	Blue	🟩	Unused	Mode B +
5	White/Blue	▨	Unused	Mode B +
6	Orange	🟩	Receive-	Mode A -
7	White/Brown	▨	Unused	Mode B -
8	Brown	🟩	Unused	Mode B -

Power over Ethernet Power Delivery

Protocol Details:

	Frequency (MHz)	Symbol Encoding	Signal Rate (Mbaud)	Symbol Rate	Data Encoding	Data Bits per Symbol	Pairs per Channel	Pairs Used	Minimum Cable Category
10BaseT	10	Manchester	10	10	None	1	1	2	3
100BaseT4	12.5	Multi-level,	25	25	8B6T	8/6	3	4	3

	Frequency (MHz)	Symbol Encoding	Signal Rate (Mbaud)	Symbol Rate	Data Encoding	Data Bits per Symbol	Pairs per Channel	Pairs Used	Minimum Cable Category
100BaseTX	31.25	2T/Hz MLT-3	125	125	4B5B	4/5	1	2	5
100BaseT2	12.5	PAM5x5 (2D-PAM5)	25	12.5	None	4 (2x2)	2	2	3
1000BaseT	31.25	4D-PAM5	125	31.25	None	8 (4x2)	4	4	5*

*Designed to work on MOST category 5 cable, category 5e specifications ensure 1000Base-T operation

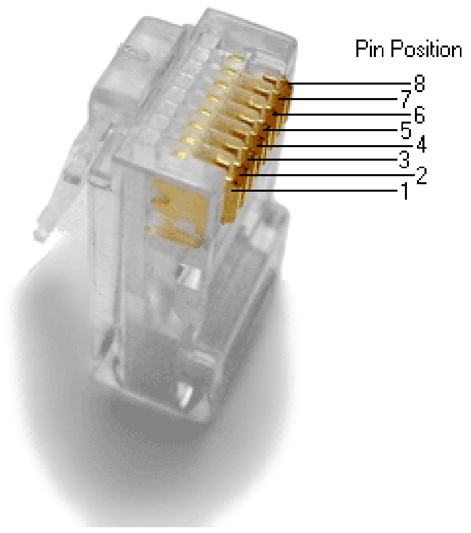
Cable Category Details:

Cable Category	Rated Frequency Bandwidth (MHz)	Common Uses
1	None	Common Use
2	1	Telephone Wiring
3	16	Telephone Wiring, 10Base-T
4	20	Token-Ring, 10Base-T
5	100	100Base-TX, 10Base-T
5e	100	1000Base-T, 100Base-TX
6	250	1000Base-T, 100Base-TX
6a*	500	10GBase-T
7	600	

Increasing category levels are backward compatible.
Manufacturers will often test and certify their cable well beyond the standards.
*10GBase-T should work on Cat6, but to get the full 100m range, Cat 6a is required.

Category 5 cable

Category 5 cable, commonly known as **Cat 5**, is a twisted pair (4 pairs) cable type designed for high signal integrity. Many such cables are unshielded but some are shielded. This type of cable is often used in structured cabling for computer networks such as Ethernet, and is also used to carry many other signals such as basic voice services, token ring, and ATM (at up to 155 Mbit/s, over short distances).



Cat-5 cable, sometimes called Ethernet cable, is short for Category 5 cable, a current industry standard for network and telephone wiring. Cat-5 cable is unshielded wire containing four pairs of 24-gauge twisted copper pairs, terminating in an RJ-45 jack. If a wire is certified as Cat-5 and not just a twisted pair wire, it will have "Cat-5" printed on the shielding.

The outer sheath of Cat-5 cable can come in many colors, with bright blue being quite common. Inside, the twisted pairs are also sheathed in plastic with a standard color scheme: Solid orange, blue, green and brown wires twisted around mates that are white and striped with a solid color. The twisted pairs inside a Cat-5 cable reduce interference and crosstalk, and should be left twisted except at the termination point. Some experts recommend untwisting only ½ inch (12.7 mm) of the pairs to strip and make connections. Cat-5 cable can be purchased off a spool in varying lengths, or bought pre-cut to standard lengths with RJ-45 jacks already attached.

Cat-5 cable replaces Cat-3 cable, which could only carry data at speeds up to 10 megabits per second (mbps), while Cat-5 cable supports data speeds of 100 mbps or more. A standard Cat-5 cable can also reach 300 feet (100 meters), and aside from networks and telephones it can be used for many other purposes. Cat-5e is enhanced Cat-5 cable that supports 1000 mbps or *gigabit Ethernet*, or it can be used with 100 Base-T networks for long-distance runs of 1150 feet (350 meters). This type of Cat-5 cable meets a specific standard referred to as "EIA/TIA 568A-5," which should be stamped on the outer sheath.

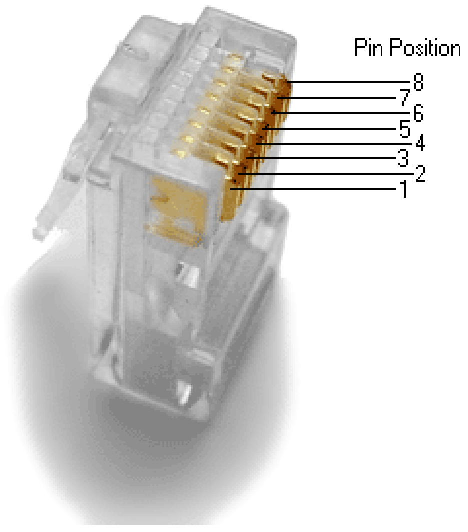
Among Cat-5 cables, there are three different configurations for pinouts, or wiring of the RJ-45 connectors. Various network devices utilize one of the three types of pinouts. The three pinouts are referred to as *straight through*, *crossover* and *roll-over*.

The Cat-5 cable that runs from a computer to a switch will be a straight through cable, for example. If two PCs or two switches are connected, the Cat-5 crossover cable would be used. Finally, a Cat-5 roll-over cable will connect a PC to a router. More recent devices, however, can detect the type of Cat-5 cable being used and route signals accordingly.

Cat-5 cable is available everywhere wiring is sold, including electronics shops, home improvement centers, and computer outlets. It is also widely available online. Note that a newer Cat-6 cable will eventually replace Cat-5, having twice the bandwidth.

RJ-45 Connectors

Short for *Registered Jack-45,* an eight-wire connector used commonly to connect computers onto a local-area networks (LAN), especially Ethernets. RJ-45 connectors look similar to the ubiquitous RJ-11 connectors used for connecting telephone equipment, but they are somewhat wider.



Pinout for Ethernet

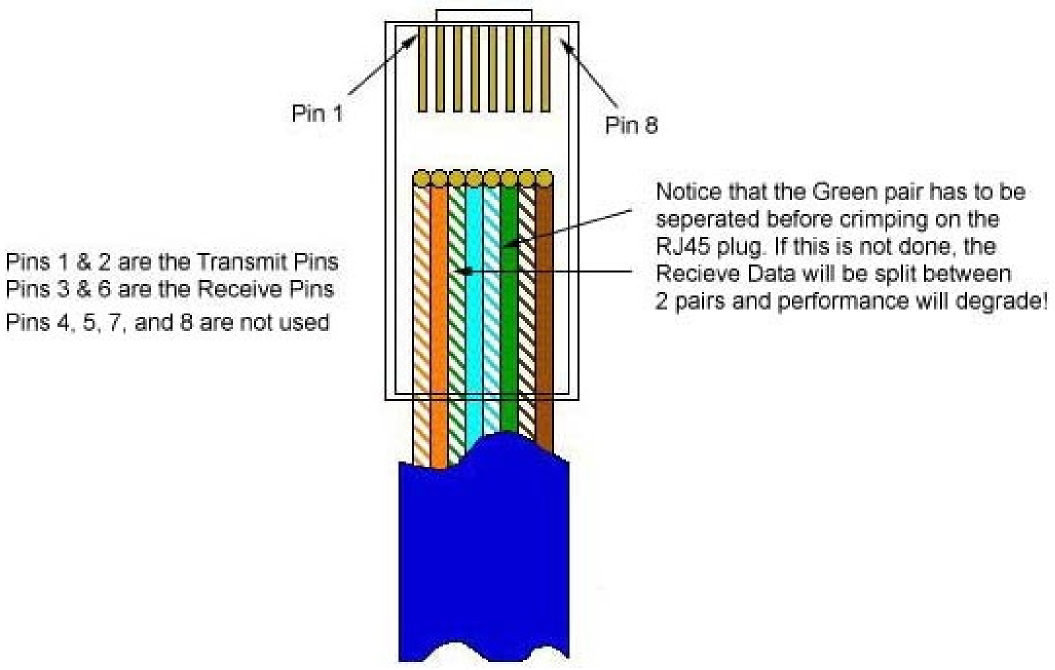
Although used for a variety of purposes, the RJ-45 connector is probably most commonly used for 10Base-T and 100Base-TX Ethernet connections.

Pin #	Ethernet 10BASE-T 100BASE-TX	EIA/TIA 568A	EIA/TIA 568B or AT&T 258A
1	Transmit +	White with green strip	White with orange stripe

2	Transmit -	Green with white stripe or solid green	Orange with white stripe or solid orange
3	Receive +	White with orange stripe	White with green stripe
4	N/A	Blue with white stripe or solid blue	Blue with white stripe or solid blue
5	N/A	White with blue stripe	White with blue stripe
6	Receive -	Orange with white stripe or solid orange	Green with white stripe or solid
7	N/A	White with brown strip or solid brown	White with brown strip or solid brown
8	N/A	Brown with white stripe or solid brown.	Brown with white stripe or solid brown.

Because only two pairs of wires in the eight-pin RJ-45 connector are used to carry Ethernet signals, and both 10BASE-T and 100BASE-TX use the same pins, a crossover cable made for one will also work with the other.

Also, please note that it is very important that a single pair be used for pins 3 and 6. If one conductor from one pair is used for pin 3 and a conductor from another pair is used for pin 6, performance will degrade. See the following figure.


Pin 1
Pin 8
Pins 1 & 2 are the Transmit Pins
Pins 3 & 6 are the Receive Pins
Pins 4, 5, 7, and 8 are not used
Notice that the Green pair has to be seperated before crimping on the RJ45 plug. If this is not done, the Recieve Data will be split between 2 pairs and performance will degrade!

To learn more about Ethernet, check out Charles Spurgeon's Ethernet Reference.

RJ-45 Pinout for RocketPort

The following chart shows the pinout for RJ-45 connectors used on certain RocketPort serial interface cards (manufactured by Comtrol).

Pin	Name/Description
1	Request To Send
2	Data Terminal Ready
3	Ground
4	Transmit Data
5	Receive Data
6	Data Carrier Detect
7	Data Set Ready
8	Clear To Send

Pinouts for ISDN

Here's an ISDN BRI U port pinout for a Cisco 750 series router:

Pin	Function
1	Not used
2	Not used
3	Not used
4	U interface network connection (tip)
5	U interface network connection (ring)
6	Not used
7	Power (pass-through to S connector)
8	Ground (pass-through to S connector)

The following chart shows the pinout for RJ-45 connectors used on certain ISDN S/T interfaces. For more info, see ANSI T1.605.

Pin	Color	Name/Description
1	White/Orange	N/A
2	Orange	N/A
3	White/Green	Receive+
4	Blue	Transmit +
5	White/Blue	Transmit -
6	Green	Receive -
7	White/Brown	-48VDC (optional)
8	Brown	-48VDC Return (optional)

2.0 NIC Installation & Configuration

2.1 Network Hardware Installation

The purpose of this documentation is to aid first-time users in the installation of an ISA/PCI/PCMCIA network interface card such as D-Link, 3COM,SMC 10/100 PCI Fast
Ethernet Card. This tutorial will cover both Desktop PC and Macintosh desktop based machines as well as Notebook computers.

Desktop PC

What you will need:

1. A screwdriver (preferably a Philips) of moderate size.
2. An anti-static strip bracelet.
3. Your computer's user manual.
4. Your new Network card, driver disks, and user's manual.

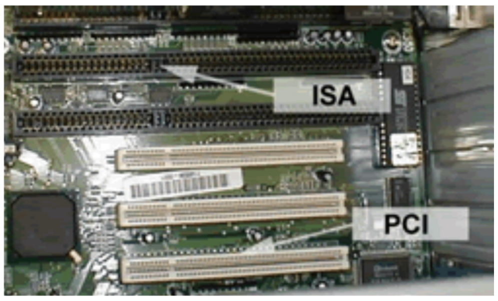
Safety Precautions

1. To prevent static electricity from damaging vital components of your computer, remember to always attach an antistatic strip bracelet from your wrist to your computer case.
2. Computer cases were not meant to be opened by the everyday user and thus are not made with the safety of the user in mind. Be careful for sharp edges in the casing that can cut your fingers and/or hands.
3. Never remove a component or open a computer case while the power is on and the power cable attached. Always remove all connecting cables before opening your case.

Step 1: Opening the Case

1. Shut off the system if it is on.
2. Remove all cables connecting to the computer.
3. Locate the screws holding the case cover in place on the frame.
4. Remove the screws attaching the cover to the frame.
5. Many new systems have tight cases and/or special cases. Removing the casing might require some prying. Use a flat-head screwdriver to push the case open against the front panel. Seek assistance if you cannot open the case alone. If the case seems really peculiar. Check your computer's user manual first to see if they instruct you on how to open your computer.

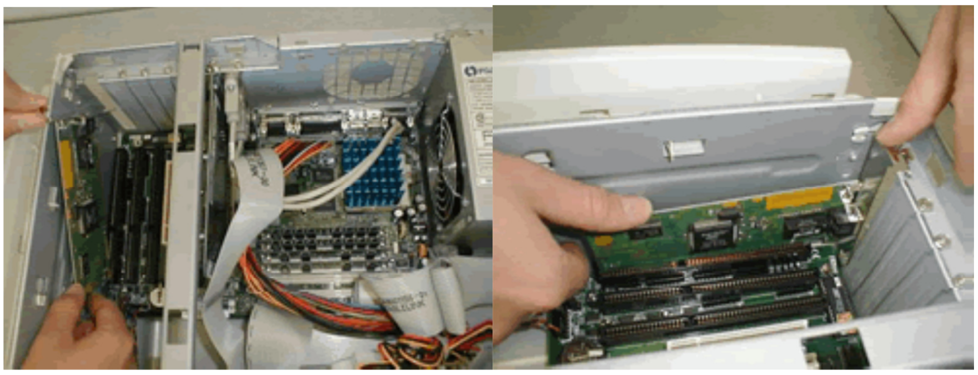
Step 2: Locating the Expansion Slots



1. Place the open computer frame on its side with the motherboard facing up. This means you can see the motherboard from a bird's eye view.
The motherboard is the biggest board you can see within the frame. It usually covers an entire side and has other smaller boards sticking up from it.

2. Looking at the motherboard, try to locate the expansion slots. Expansion slots are either long black strips or short white strips that look like Lego blocks standing up. ISA slots are black. PCI slots are white. Open slots are those that do not have other boards inserted in them.

Step 3: Installing Your New Card



1. Determine which interface (ISA or PCI) your card uses. ISA is long and the gold contacts are large. PCI is much shorter and smaller.

2. Next, check to see if the expansion slot opening next to the slot is covered. If it is, remove the cover by unscrewing it from the frame or popping it out. (IMPORTANT: Keep the screw and the slot cover.) If you have a new case that has slot covers built in you will have to remove them manually with a screwdriver. Please refer to your user manual for details.

3. When the slot cover has been removed, insert your card into the expansion slot on the motherboard. Press firmly so the entire part of the card that has the gold contacts goes completely into the expansion slot on the motherboard and will go no further. Do not use any tools to try to hammer the card in if it does not fit.

4. Make sure the side of the card resembling the expansion slot cover you just removed is covering most of the open slot.

5. Screw the card into place with the screw you removed from the expansion
slot cover or a new screw.

Step 4: Replacing the Case

1. After confirming the proper placement of the card, make sure you did
not leave any tools or screws within the computer. Replace the case and screw it back in place.

2. Reconnect all the cables to their proper places.

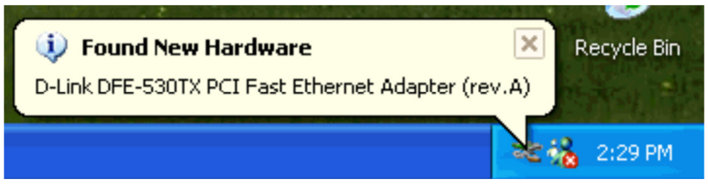
Step 5: Booting Up

1. Turn on the power.
2. Refer to your user's manual to install the proper drivers from the disk(s) that came with the card.
3. Refer to our guide Installing a Network Interface Card for a specific Operating System.

Installing Network Drivers on Windows XP

After the physical installation of the card in your PC you will have to install the
drivers for your Windows Operating System.You must be logged-in as Administrator
or your login must have Administrator rights.

1. Power on your computer.


Found New Hardware
D-Link DFE-530TX PCI Fast Ethernet Adapter (rev.A)

2. As the computer is loading Windows XP it will automatically find your new
Ethernet card and configure it to work properly. A small dialogue box should appear
at the bottom right corner of your desktop to inform you that Windows XP has found
your card and has installed it.
If the card doesn't install properly refer to the Troubleshooting guide. Otherwise
continue with Configuring Network Settings for Windows XP.

3.0 IP Address configuration

Q. Write down the steps to configuring IP address of a machine (windows'98/2000/XP, Linux).

3.1 IP – address configuration on Linux machine.

<u>Procedure:-</u>

This procedure is for a Linux environment.

```
Step I : - Go to the Terminal.
Step II : - # ifconfig eth0 192.168.0.3
Step III : -  # Service network restart.            // ( For restart your computer)
Step IV : - # ifconfig
Step V : -  # ping 192.168.0.27             // (connectivity testing)
```

<u>Report:-</u>

Output:-

```
Syntax  #ifconfig
eth0 :   flags=4043<UP, BROADCAST, RUNNING, MULTICAST > mtu 1500
         inet 192.168.0.3 netmask ffffff00 broadcast 192.168.0.255
         perf.Params : recv size : 4096; send 8192; full-size frams :1
         ether 00:20:18:62:47:e0

Syntax  #ping 192.168.0.27
         PING 192.168.0.3 (192.168.0.27) 56(84) bytes of data
                  64 bytes from 192.168.0.27:icmp_seq ttl=64 time=0.053ms
                  64 bytes from 192.168.0.27:icmp_seq ttl=64 time=0.048ms
                  64 bytes from 192.168.0.27:icmp_seq ttl=64 time=0.051ms
                  64 bytes from 192.168.0.27:icmp_seq ttl=64 time=0.047ms
[Ctrl + c]
                  ----earth PING statistics----
                  4 packet transmitted, 4 packet received, 0% packet loss
                  Round-trip(ms) min/avg/max=0/0
```

Remarks:-

The command ifconfig is used for configuring the network interface. Since TCP/IP is independent of the network hardware, the IP addresses are not built to the kernel, but rather reside in the networking software. I have to use the ifconfig command to set the IP address of your interface. The command is used like .. Step II.

Syntax # ifconfig eth0 192.168.0.3

► The syntax requires the command to be followed by the *interface name* (here, eth0) and the *IP address*.

► ifconfig not only tells you the IP address of my machine, but also its *status*. The system is UP, supports BROADCASTs and is currently RUNNING.

► Note that it also shows you the *MAC address* of the interface card (00:20:18:62:47:e0)
Syntax # ping 192.168.0.27

► *ping (Packet Internet Grouper)* is used for *checking the network*.

3.2 IP – address configuration of Windows'98/2000/XP

For Windows'98 Network Neighborhood (Properties) Configuration
TCP/IP (NIC Adapter) Properties IP Address (specify an IP address)

For Windows'XP Start Control Panel Network and internet connection
Network Connection Local area connection (properties) Internet protocols (TCP/IP) Properties

For Windows'2000 My computer (properties) Network Identifier properties
IP address

3.3. Some Network Commands

3.3.1 Ping

Ping is a computer network tool used to test whether a particular host is reachable across an IP network; it is also used to self test the network interface card of the computer, or as a speed test. It works by sending ICMP "echo request" packets to the target host and listening for ICMP "echo response" replies.

e.g.
admin@localhost# ping en.wikipedia.org

```
PING rr.pmtpa.wikimedia.org (66.230.200.100) 56(84) bytes of data.
64 bytes from rr.pmtpa.wikimedia.org (66.230.200.100): icmp_seq=1 ttl=52 time=87.7 ms
64 bytes from rr.pmtpa.wikimedia.org (66.230.200.100): icmp_seq=2 ttl=52 time=95.6 ms
```

```
64 bytes from rr.pmtpa.wikimedia.org (66.230.200.100): icmp_seq=3 ttl=52 time=85.4 ms
64 bytes from rr.pmtpa.wikimedia.org (66.230.200.100): icmp_seq=4 ttl=52 time=95.8 ms
64 bytes from rr.pmtpa.wikimedia.org (66.230.200.100): icmp_seq=5 ttl=52 time=87.0 ms
64 bytes from rr.pmtpa.wikimedia.org (66.230.200.100): icmp_seq=6 ttl=52 time=97.6 ms
--- rr.pmtpa.wikimedia.org ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 8998ms
rtt min/avg/max/mdev = 78.162/89.213/97.695/6.836 ms
```

3.3.2 ifconfig

Configure a network interface
There are many more parameters to **ifconfig** than we have described so far. Its normal invocation is this:

```
ifconfig interface [address [parameters]]
```

interface is the interface name, and *address* is the IP address to be assigned to the interface. This may be either an IP address in dotted quad notation or a name that **ifconfig** will look up in /etc/hosts.

If **ifconfig** is invoked with only the interface name, it displays that interface's configuration. When invoked without any parameters, it displays all interfaces you have configured so far; a –a option forces it to show the inactive ones as well. A sample invocation for the Ethernet interface eth0 may look like this:

```
# ifconfig eth0
eth0      Link encap 10Mbps Ethernet  HWaddr 00:00:C0:90:B3:42
inet addr 172.16.1.2 Bcast 172.16.1.255 Mask 255.255.255.0
UP BROADCAST RUNNING  MTU 1500  Metric 0
RX packets 3136 errors 217 dropped 7 overrun 26
TX packets 1752 errors 25 dropped 0 overrun 0
```

The MTU and Metric fields show the current MTU and metric value for that interface. The metric value is traditionally used by some operating systems to compute the cost of a route. Linux doesn't use this value yet, but defines it for compatibility, nevertheless.

The RX and TX lines show how many packets have been received or transmitted error free, how many errors occurred, how many packets were dropped (probably because of low memory), and how many were lost because of an overrun. Receiver overruns usually occur when packets come in faster than the kernel can service the last interrupt. The flag values printed by **ifconfig** roughly correspond to the names of its command-line options; they will be explained later.

The following is a list of parameters recognized by **ifconfig** with the corresponding flag names. Options that simply turn on a feature also allow it to be turned off again by preceding the option name by a dash (−).

up

This option makes an interface accessible to the IP layer. This option is implied when an *address* is given on the command line. It may also be used to reenable an interface that has been taken down temporarily using the down option.

This option corresponds to the flags UP and RUNNING.

down

This option marks an interface inaccessible to the IP layer. This effectively disables any IP traffic through the interface. Note that this option will also automatically delete all routing entries that use this interface.

netmask*mask*

This option assigns a subnet mask to be used by the interface. It may be given as either a 32-bit hexadecimal number preceded by 0x, or as a dotted quad of decimal numbers. While the dotted quad format is more common, the hexadecimal representation is often easier to work with. Netmasks are essentially binary, and it is easier to do binary-to-hexadecimal than binary-to-decimal conversion.

pointopoint*address*

This option is used for point-to-point IP links that involve only two hosts. This option is needed to configure SLIP or PLIP interfaces, for example. If a point-to-point address has been set, **ifconfig** displays the POINTOPOINT flag.

broadcast*address*

The broadcast address is usually made up from the network number by setting all bits of the host part. Some IP implementations (systems derived from BSD 4.2, for instance) use a different scheme in which all host part bits are cleared instead. The broadcast option adapts to these strange environments. If a broadcast address has been set, **ifconfig** displays the BROADCAST flag.

irq

This option allows you to set the IRQ line used by certain devices. This is especially useful for PLIP, but may also be useful for certain Ethernet cards.

metric*number*

This option may be used to assign a metric value to the routing table entry created for the interface. This metric is used by the Routing Information Protocol (RIP) to build routing tables for the network.[1] The default metric used by **ifconfig** is zero. If you don't run a RIP daemon, you don't need this option at all; if you do, you will rarely need to change the metric value.

mtu*bytes*

This sets the Maximum Transmission Unit, which is the maximum number of octets the interface is able to handle in one transaction. For Ethernets, the MTU defaults to 1,500 (the largest allowable size of an Ethernet packet); for SLIP interfaces, it is 296. (There is no constraint on the MTU of SLIP links; this value is a good compromise.)

arp

This is an option specific to broadcast networks such as Ethernets or packet radio. It enables the use of the Address Resolution Protocol (ARP) to detect the physical addresses of hosts attached to the network. For broadcast networks, it is on by default. If ARP is disabled, **ifconfig** displays the NOARP flag.

–arp

This option disables the use of ARP on this interface.

promisc

This option puts the interface in promiscuous mode. On a broadcast network, this makes the interface receive all packets, regardless of whether they were destined for this host or not. This allows network traffic analysis using packet filters and such, also called *Ethernet snooping*. Usually, this is a good technique for hunting down network problems that are otherwise hard to detect. Tools such as **tcpdump** rely on this.

On the other hand, this option allows attackers to do nasty things, such as skim the traffic of your network for passwords. You can protect against this type of attack by prohibiting just anyone from plugging their computers into your Ethernet. You could also use secure authentication protocols, such as Kerberos or the secure shell login suite.[2] This option corresponds to the PROMISC flag.

–promisc

This option turns promiscuous mode off.

allmulti

Multicast addresses are like Ethernet broadcast addresses, except that instead of automatically including everybody, the only people who receive packets sent to a multicast address are those programmed to listen to it. This is useful for applications like Ethernet-based videoconferencing or network audio, to which only those interested can listen. Multicast addressing is supported by most, but not all, Ethernet drivers. When this option is enabled, the interface receives and passes multicast packets for processing. This option corresponds to the ALLMULTI flag.

–allmulti

This option turns multicast addresses off.

3.3.3 traceroute

Print the route packet takes to network hosts.

The traceroute command traces the network path of Internet *routers* that *packets* take as they are forwarded from your computer to a destination address. The "length" of the network connection is indicated by the number of Internet routers in the traceroute path.

Traceroutes can be useful to diagnose slow network connections. For example, if you can usually reach an Internet site but it is slow today, then a traceroute to that site should show you one or more hops with either long times or marked with "*" indicating the time was *really* long.

Example

Estonia to the United States. 195.80.96.219 (XYZ.co.in) to 130.94.122.199 (abcd.org).
Windows command : tracert 130.94.122.199
Linux or Mac OS X command : traceroute 130.94.122.199

Uses

Traceroute is often used for network troubleshooting. By showing a list of routers traversed, it allows the user to identify the path taken to reach a particular destination on the network. This can help identify routing problems or firewalls that may be blocking access to a site. Traceroute is also used by penetration testers to gather information about network infrastructure and IP ranges around a given host. It can also be used when downloading data, and if there are multiple mirrors available for the same piece of data, one can trace each mirror to get a good idea of which mirror would be the fastest to use.

3.3.4 telnet

User interface to the TELNET protocol.

Description – The telnet command is used to communicate with another host using the TELNET protocol. If telnet is invoked without the host argument, it enters command modes, indicated by its prompt (telnet >). In this mode, it accepts and executes the commands. If it is invoked with arguments, it performs an open command with those arguments.

Telnet commands

The Telnet Client command prompt accepts the following commands:

Command	**Description**
open	Use **open hostname portnumber** to establish a Telnet connection to a host.
close	Use the **close** command to close an existing Telnet connection. Use the **display** command to view the current settings for the Telnet client.
display	The **display** command lists the current operating parameters. If you are in a Telnet session (connected to a Telnet server), to modify the parameters, press CTRL+]. This escapes from the Telnet session. (To return to the Telnet

session, press ENTER.) The following operating parameters are available:

- **WILL AUTH** (NTLM Authentication)
- **WONT AUTH**
- **WILL TERM TYPE**
- **WONT TERM TYPE**
- **LOCALECHO off**
- **LOCALECHO on**

quit Use the **quit** command to exit from Telnet.

set Use the **set** command to set the terminal type for the connection, turn on local echo, set authentication to NTLM, set the escape character, and set up logging.

- **SET NTLM** turns on NTLM.

  While you are using NTLM Authentication, you are not prompted for a logon name and password when connecting from a remote computer.

- **SET LOCALECHO** turns on local echoing.
- **SET TERM {ANSI|VT100|VT52|VTNT}** sets the terminal type to the appropriate terminal type.

  Use the VT100 terminal type if you are running normal command-line applications. Use the VTNT terminal type if you are running advanced command-line applications, such as **edit**.

- **ESCAPE** *Character* sets the key sequence to use for switching from session to command mode. For example, to set CTRL+P as your escape character, type **set escape**, press CTRL+P, and then press ENTER.
- **LOGFILE** *FileName* sets the file to be used for logging Telnet activity. The log file must be on your local computer.

  Logging begins automatically when you set this option.

- **LOGGING** turns on logging.

  If no log file is set, an error message is displayed.

unset Use **unset** to turn off local echo or to set authentication to logon/password prompt.

- **UNSET NLM** turns off NLM.
- **UNSET LOCALECHO** turns off local echoing.

status	Use the **status** command to determine whether the Telnet client is connected.
CTRL+]	Press CTRL+] to move to the Telnet command prompt from a connected session.
enter	Use the **enter** command from the command prompt to go to the connected session (if it exists).
?/help	Prints Help information.

3.3.4 ftp

Internet file transfer program.

Description – ftp is the user interface to the internet standard. File transfer protocol. The program allows a user to transfer files to and from a remote network site.

What is FTP?

The FTP (**F**ile **T**ransfer **P**rotocol) utility program is commonly used for copying files to and from other computers. These computers may be at the same site or at different sites thousands of miles apart. FTP is a general protocol that works on UNIX systems as well as a variety of other (non-UNIX) systems.

For the purposes of this Web page, the *local* machine refers to the machine you are initially logged into, the one on which you type the **ftp** command. The *remote* machine is the other one, the one that is the argument of the **ftp** command.

A user interface for the standard File Transfer Protocol for ARPANET, FTP acts as an interpreter on the remote machine. The user may type a number of UNIX-like commands under this interpreter to perform desired actions on the remote machine.

Most operating systems and communication programs now include some form of an FTP utility program, but the commands differ slightly between them. The following explanations and alphabetical list of commands refers to the common FTP utility program as provided on a UNIX machine. Check the documentation for your own machine to determine the comparable commands.

Most computers today include a windows-based type FTP program that is more PC-oriented and does not require full knowledge of these commands.
You can also perform FTP through a browser. For example, bring up Internet Explorer and type in

ftp://*yourLoginName@IPaddress*

instead of a normal web page URL.

Getting Started

To connect your local machine to the remote machine, type

```
ftpmachinename
```

where *machinename* is the full machine name of the remote machine, e.g., purcell.cs.colostate.edu. If the name of the machine is unknown, you may type

```
ftpmachinennumber
```

where *machinennumber* is the net address of the remote machine, e.g., 129.82.45.181. In either case, this command is similar to logging onto the remote machine. If the remote machine has been reached successfully, FTP responds by asking for a *loginname* and *password*.

When you enter your own *loginname* and *password* for the remote machine, it returns the prompt

```
ftp>
```

and permits you access to your own home directory on the remote machine. You should be able to move around in your own directory and to copy files to and from your local machine using the FTP interface commands given on the following page.

Anonymous FTP

At times you may wish to copy files from a remote machine on which you do not have a *loginname*. This can be done using *anonymous* FTP.
When the remote machine asks for your *loginname*, you should type in the word anonymous. Instead of a *password*, you should enter your own electronic mail address. This allows the remote site to keep records of the anonymous FTP requests.
Once you have been logged in, you are in the anonymous directory for the remote machine. This usually contains a number of public files and directories. Again you should be able to move around in these directories. However, you are only able to copy the files from the remote machine to your own local machine; you are not able to write on the remote machine or to delete any files there.

4.0 Configure Telnet and Ftp in Linux.

4.1 Telnet Configuration on Redhat Linux 9.

Step 1: go to /etc/xinetd.d (cd /etc/xinetd.d)

Step 2: edit telnet (vi telnet)

```
1       # default: on
2       # description: The telnet server serves telnet sessions; it uses
3       #     unencrypted username/password pairs for authentication.
4       service telnet
5       {
6       disable = yes
7       flags          = REUSE
8       socket_type    = stream
9       wait           = no
10      user           = root
11      only_from      = 192.168.12.117
12      server         = /usr/sbin/in.telnetd
13      log_on_failure  = USERID
14      }
```

Step 3: At line number 6 change the value of disable to no.

Step 4: go to /etc/ (cd /etc)

Step 5: edit securetty (vi telnet)

```
console
vc/1
vc/2
vc/3
vc/4
vc/5
vc/6
vc/7
vc/8
vc/9
vc/10
vc/11
tty1
tty2
```

```
        tty3
        tty4
        tty5
        tty6
        tty7
        tty8
        tty9
        tty10
        tty11
```

Step 5: add pts/0 to pts/n for concurrent telnet session.

Step 6: restart xinetd (service xinetd restart)

Step 7: type “telnet ip_address” from remote machine.

```
Red Hat Linux release 9 (Shrike)
Kernel 2.4.20-8 on an i686
login: username
password: password
```

Step 8: enter username and password of remote machine to start telnet session.

4.2 FTP Configuration

Introduction

The File Transfer Protocol (FTP) is used as one of the most common means of copying files between servers over the Internet. Most web based download sites use the built in FTP capabilities of web browsers and therefore most server oriented operating systems usually include an FTP server application as part of the software suite. Linux is no exception.

This chapter will show you how to convert your Linux box into an FTP server using the default Very Secure FTP Daemon (VSFTPD) package included in Fedora.

FTP Overview

FTP relies on a pair of TCP ports to get the job done. It operates in two connection channels as I'll explain:

FTP Control Channel, TCP Port 21: All commands you send and the ftp server's responses to those commands will go over the control connection, but any data sent back (such as "ls" directory lists or actual file data in either direction) will go over the data connection.

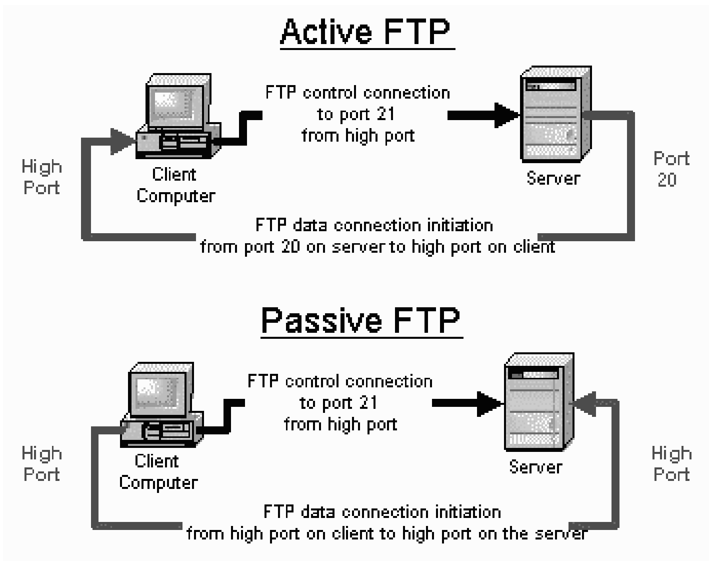
FTP Data Channel, TCP Port 20: This port is used for all subsequent data transfers between the client and server.

In addition to these channels, there are several varieties of FTP.

Types of FTP

From a networking perspective, the two main types of FTP are active and passive. In active FTP, the FTP server initiates a data transfer connection back to the client. For passive FTP, the connection is initiated from the FTP client. These are illustrated in Figure 15-1.

Figure Active And Passive FTP Illustrated


Active FTP
FTP control connection to port 21 from high port
High Port
Client Computer
Server
Port 20
FTP data connection initiation from port 20 on server to high port on client
Passive FTP
FTP control connection to port 21 from high port
High Port
Client Computer
Server
High Port
FTP data connection initiation from high port on client to high port on the server

From a user management perspective there are also two types of FTP: regular FTP in which files are transferred using the username and password of a regular user FTP server, and anonymous FTP in which general access is provided to the FTP server using a well known universal login method.

Take a closer look at each type.

Active FTP

The sequence of events for active FTP is:

1. Your client connects to the FTP server by establishing an FTP control connection to port 21 of the server. Your commands such as 'ls' and 'get' are sent over this connection.
2. Whenever the client requests data over the control connection, the server initiates data transfer connections back to the client. The source port of these data transfer connections is always port 20 on the server, and the destination port is a high port (greater than 1024) on the client.
3. Thus the ls listing that you asked for comes back over the port 20 to high port connection, not the port 21 control connection.

FTP active mode therefore transfers data in a counter intuitive way to the TCP standard, as it selects port 20 as it's source port (not a random high port that's greater than 1024) and connects back to the client on a random high port that has been pre-negotiated on the port 21 control connection.

Active FTP may fail in cases where the client is protected from the Internet via many to one NAT (masquerading). This is because the firewall will not know which of the many servers behind it should receive the return connection.

Passive FTP

Passive FTP works differently:

1. Your client connects to the FTP server by establishing an FTP control connection to port 21 of the server. Your commands such as ls and get are sent over that connection.
2. Whenever the client requests data over the control connection, the client initiates the data transfer connections to the server. The source port of these data transfer connections is always a high port on the client with a destination port of a high port on the server.

Passive FTP should be viewed as the server never making an active attempt to connect to the client for FTP data transfers. Because client always initiates the required connections, passive FTP works better for clients protected by a firewall.

As Windows defaults to active FTP, and Linux defaults to passive, you'll probably have to accommodate both forms when deciding upon a security policy for your FTP server.

Regular FTP

By default, the VSFTPD package allows regular Linux users to copy files to and from their home directories with an FTP client using their Linux usernames and passwords as their login credentials.

VSFTPD also has the option of allowing this type of access to only a group of Linux users, enabling you to restrict the addition of new files to your system to authorized personnel.

The disadvantage of regular FTP is that it isn't suitable for general download distribution of software as everyone either has to get a unique Linux user account or has to use a shared username and password. Anonymous FTP allows you to avoid this difficulty.

Anonymous FTP

Anonymous FTP is the choice of Web sites that need to exchange files with numerous unknown remote users. Common uses include downloading software updates and MP3s and uploading diagnostic information for a technical support engineers' attention. Unlike regular FTP where you login with a preconfigured Linux username and password, anonymous FTP requires only a username of anonymous and your email address for the password. Once logged in to a VSFTPD server, you automatically have access to only the default anonymous FTP directory (/var/ftp in the case of VSFTPD) and all its subdirectories.

As seen in Chapter 6, "Installing Linux Software", using anonymous FTP as a remote user is fairly straight forward. VSFTPD can be configured to support user-based and or anonymous FTP in its configuration file which you'll see later.

How To Download And Install VSFTPD

Most Linux software products are available in a precompiled package format. Downloading and installing packages isn't hard. If you need a refresher, Chapter 6, "Installing Linux Software", covers how to do this in detail. It is best to use the latest version of VSFTPD.

When searching for the file, remember that the VSFTPD packages' filename usually starts with the word vsftpd followed by a version number, as in `vsftpd-1.2.1-5.i386.rpm` for Redhat/Fedora.

How To Get VSFTPD Started

With Fedora, Redhat, Ubunbtu and Debian You can start, stop, or restart VSFTPD after booting by using these commands:

```
[root@bigboy tmp]# /etc/init.d/vsftpd start
[root@bigboy tmp]# /etc/init.d/vsftpd stop
[root@bigboy tmp]# /etc/init.d/vsftpd restart
```

With Redhat / Fedora you can configure VSFTPD to start at boot you can use the chkconfig command.

```
[root@bigboy tmp]# chkconfig vsftpd on
```

Note: In RedHat Linux version 8.0 and earlier, VSFTPD operation is controlled by the xinetd process, which is covered in Chapter 16, "Telnet, TFTP, and xinetd". You can find a full description of how to configure these versions of Linux for VSFTPD in Appendix III, "Fedora Version Differences."

Testing the Status of VSFTPD

You can always test whether the VSFTPD process is running by using the netstat -a command which lists all the TCP and UDP ports on which the server is listening for traffic. This example shows the expected output.

```
[root@bigboy root]# netstat -a | grep ftp
tcp     0     0     *:ftp     *:*     LISTEN
[root@bigboy root]#
```

If VSFTPD wasn't running, there would be no output at all.

The vsftpd.conf File

VSFTPD only reads the contents of its vsftpd.conf configuration file only when it starts, so you'll have to restart VSFTPD each time you edit the file in order for the changes to take effect. The file may be located in either the `/etc` or the `/etc/vsftpd` directories depending on your Linux distribution.

This file uses a number of default settings you need to know about.

- VSFTPD runs as an anonymous FTP server. Unless you want any remote user to log into to your default FTP directory using a username of anonymous and a password that's the same as their email address, I would suggest turning this off. The configuration file's anonymous_enable directive can be set to no to disable this feature. You'll also need to simultaneously enable local users to be able to log in by removing the comment symbol (#) before the local_enable instruction.
- If you enable anonymous FTP with VSFTPD, remember to define the root directory that visitors will visit. This is done with the anon_root directive.

```
anon_root=/data/directory
```

- VSFTPD allows only anonymous FTP downloads to remote users, not uploads from them. This can be changed by modifying the anon_upload_enable directive shown later.
- VSFTPD doesn't allow anonymous users to create directories on your FTP server. You can change this by modifying the anon_mkdir_write_enable directive.
- VSFTPD logs FTP access to the /var/log/vsftpd.log log file. You can change this by modifying the xferlog_file directive.
- By default VSFTPD expects files for anonymous FTP to be placed in the /var/ftp directory. You can change this by modifying the anon_root directive. There is always the risk with anonymous FTP that users will discover a way to write files to your anonymous FTP directory. You run the risk of filling up your /var partition if you use the default setting. It is best to make the anonymous FTP directory reside in its own dedicated partition.

The configuration file is fairly straight forward as you can see in the snippet below where we enable anonymous FTP and individual accounts simultaneously.

```
# Allow anonymous FTP?
anonymous_enable=YES
...
# The directory which vsftpd will try to change
# into after an anonymous login. (Default = /var/ftp)
anon_root=/data/directory
...
# Uncomment this to allow local users to log in.
local_enable=YES
...
# Uncomment this to enable any form of FTP write command.
# (Needed even if you want local users to be able to upload files)
write_enable=YES
...
# Uncomment to allow the anonymous FTP user to upload files. This only
# has an effect if global write enable is activated. Also, you will
# obviously need to create a directory writable by the FTP user.
#anon_upload_enable=YES
...
# Uncomment this if you want the anonymous FTP user to be able to create
# new directories.
#anon_mkdir_write_enable=YES
...
# Activate logging of uploads/downloads.
xferlog_enable=YES
```

```
...
# You may override where the log file goes if you like.
# The default is shown below.
xferlog_file=/var/log/vsftpd.log
...
```

To activate or deactivate a feature, remove or add the # at the beginning of the appropriate line.

Other vsftpd.conf Options

There are many other options you can add to this file:

- Limiting the maximum number of client connections (max_clients)
- Limiting the number of connections by source IP address (max_per_ip)
- The maximum rate of data transfer per anonymous login. (anon_max_rate)
- The maximum rate of data transfer per non-anonymous login. (local_max_rate)

Descriptions on this and more can be found in the vsftpd.conf man pages.

4.3 Security Issues

FTP has a number of security drawbacks, but you can overcome them in some cases. You can restrict an individual Linux user's access to non-anonymous FTP, and you can change the configuration to not display the FTP server's software version information, but unfortunately, though very convenient, FTP logins and data transfers are not encrypted.

The /etc/vsftpd.ftpusers File

For added security, you may restrict FTP access to certain users by adding them to the list of users in the /etc/vsftpd.ftpusers file. The VSFTPD package creates this file with a number of entries for privileged users that normally shouldn't have FTP access. As FTP doesn't encrypt passwords, thereby increasing the risk of data or passwords being compromised, it is a good idea to let these entries remain and add new entries for additional security.

Anonymous Upload

If you want remote users to write data to your FTP server, then you should create a write-only directory within /var/ftp/pub. This will allow your users to upload but not access other files uploaded by other users. The commands you need are:

```
[root@bigboy tmp]# mkdir /var/ftp/pub/upload
[root@bigboy tmp]# chmod 722 /var/ftp/pub/upload
```

FTP Greeting Banner

Change the default greeting banner in the vsftpd.conf file to make it harder for malicious users to determine the type of system you have. The directive in this file is.

ftpd_banner= New Banner Here

FTP Users with Only Read Access to a Shared Directory

In this example, anonymous FTP is not desired, but a group of trusted users need to have read only access to a directory for downloading files. Here are the steps:

1) Disable anonymous FTP. Comment out the anonymous_enable line in the vsftpd.conf file like this:

```
# Allow anonymous FTP?
anonymous_enable=NO
```

2) Enable individual logins by making sure you have the local_enable line uncommented in the vsftpd.conf file like this:

```
# Uncomment this to allow local users to log in.
local_enable=YES
```

3) Start VSFTP.

```
[root@bigboy tmp]# service vsftpd start
```

4) Create a user group and shared directory. In this case, use /home/ftp-users and a user group name of ftp-users for the remote users

```
[root@bigboy tmp]# groupadd ftp-users
[root@bigboy tmp]# mkdir /home/ftp-docs
```

5) Make the directory accessible to the ftp-users group.

```
[root@bigboy tmp]# chmod 750 /home/ftp-docs
[root@bigboy tmp]# chown root:ftp-users /home/ftp-docs
```

6) Add users, and make their default directory /home/ftp-docs

```
[root@bigboy tmp]# useradd -g ftp-users -d /home/ftp-docs user1
[root@bigboy tmp]# useradd -g ftp-users -d /home/ftp-docs user2
[root@bigboy tmp]# useradd -g ftp-users -d /home/ftp-docs user3
[root@bigboy tmp]# useradd -g ftp-users -d /home/ftp-docs user4
[root@bigboy tmp]# passwd user1
[root@bigboy tmp]# passwd user2
[root@bigboy tmp]# passwd user3
[root@bigboy tmp]# passwd user4
```

7) Copy files to be downloaded by your users into the /home/ftp-docs directory

8) Change the permissions of the files in the /home/ftp-docs directory for read only access by the group

```
[root@bigboy tmp]# chown root:ftp-users /home/ftp-docs/*
```

```
[root@bigboy tmp]# chmod 740 /home/ftp-docs/*
```

Users should now be able to log in via FTP to the server using their new usernames and passwords. If you absolutely don't want any FTP users to be able to write to any directory, then you should set the write_enable line in your vsftpd.conf file to no:

```
write_enable = NO
```

Remember, you must restart VSFTPD for the configuration file changes to take effect.

Sample Login Session To Test Functionality

Here is a simple test procedure you can use to make sure everything is working correctly:

1) Check for the presence of a test file on the ftp client server.

```
[root@smallfry tmp]# ll
total 1
-rw-r--r-- 1 root root 0 Jan 4 09:08 testfile
[root@smallfry tmp]#
```

2) Connect to bigboy via FTP

```
[root@smallfry tmp]# ftp 192.168.1.100
Connected to 192.168.1.100 (192.168.1.100)
220 ready, dude (vsFTPd 1.1.0: beat me, break me)
Name (192.168.1.100:root): user1
331 Please specify the password.
Password:
230 Login successful. Have fun.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

As expected, we can't do an upload transfer of testfile to bigboy.

```
ftp> put testfile
local: testfile remote: testfile
227 Entering Passive Mode (192,168,1,100,181,210)
553 Could not create file.
ftp>
```

But we can view and download a copy of the VSFTPD RPM located on the FTP server bigboy.

```
ftp> ls
227 Entering Passive Mode (192,168,1,100,35,173)
150 Here comes the directory listing.
-rwxr----- 1 0 502 76288 Jan 04 17:06 vsftpd-1.1.0-1.i386.rpm
226 Directory send OK.
ftp> get vsftpd-1.1.0-1.i386.rpm vsftpd-1.1.0-1.i386.rpm.tmp
local: vsftpd-1.1.0-1.i386.rpm.tmp remote: vsftpd-1.1.0-1.i386.rpm
227 Entering Passive Mode (192,168,1,100,44,156)
150 Opening BINARY mode data connection for vsftpd-1.1.0-1.i386.rpm (76288 bytes).
226 File send OK.
76288 bytes received in 0.499 secs (1.5e+02 Kbytes/sec)
```

```
ftp> exit
221 Goodbye.
[root@smallfry tmp]#
```

As expected, anonymous FTP fails.

```
[root@smallfry tmp]# ftp 192.168.1.100
Connected to 192.168.1.100 (192.168.1.100)
220 ready, dude (vsFTPd 1.1.0: beat me, break me)
Name (192.168.1.100:root): anonymous
331 Please specify the password.
Password:
530 Login incorrect.
Login failed.
ftp> quit
221 Goodbye.
[root@smallfry tmp]#
```

Now that testing is complete, you can make this a regular part of your FTP server's operation.

Common FTP Commands

?	to request help or information about the FTP commands	
ascii	to set the mode of file transfer to ASCII (this is the default and transmits seven bits per character)	
binary	to set the mode of file transfer to binary (the binary mode transmits all eight bits per byte and thus provides less chance of a transmission error and must be used to transmit files other than ASCII files)	
bye	to exit the FTP environment (same as quit)	
cd	to change directory on the remote machine	
close	to terminate a connection with another computer	
	close brubeck	closes the current FTP connection with brubeck, but still leaves you within the FTP environment.
delete	to delete (remove) a file in the current remote directory (same as rm in UNIX)	
get	to copy one file from the remote machine to the local machine	
	get ABC DEF	copies file ABC in the current remote directory to (or on top of) a file named DEF in your current local directory.
	get ABC	copies file ABC in the current remote directory to (or on top of) a file with the same name, ABC, in your current local directory.
help	to request a list of all available FTP commands	
lcd	to change directory on your local machine (same as UNIX cd)	
ls	to list the names of the files in the current remote directory	
mkdir	to make a new directory within the current remote directory	
mget	to copy multiple files from the remote machine to the local machine; you are prompted for a y/n answer before transferring each file	
	mget *	copies all the files in the current remote directory to your current local directory, using the same filenames. Notice the use of the wild card character, *.
mput	to copy multiple files from the local machine to the remote machine; you are prompted for a y/n answer before transferring each file	
open	to open a connection with another computer	
	open brubeck	opens a new FTP connection with brubeck; you must enter a username and password for a brubeck account (unless it is to be an anonymous connection).
put	to copy one file from the local machine to the remote machine	
pwd	to find out the pathname of the current directory on the remote machine	
quit	to exit the FTP environment (same as bye)	
rmdir	to to remove (delete) a directory in the current remote directory	

5.0 SOCKET PROGRAMMING

In Java, two classes are provided for the datagram socket API: (a) The DatagramSocket class for the sockets (b) The DatagramPacket class for the packets exchanged. A process wishing to send or receive data using the datagram socket API must instantiate a DatagramSocket object, which is bound to a UDP port of the machine and local to the process.
To send a datagram to another process, the sender process must instantiate a DatagramPacket object that carries the following information: (1) a reference to a byte array that contains the payload data and (2) the destination address (the host ID and port number to which the receiver process' DatagramSocket object is bound).
At the receiving process, a DatagramSocket object must be instantiated and bound to a local port – this port should correspond to the port number carried in the datagram packet of the sender. To receive datagrams sent to the socket, the receiving process must instantiate a DatagramPacket object that references a byte array and call the receive method of the DatagramSocket object, specifying as argument, a reference to the DatagramPacket object. The program flow in the sender and receiver process is illustrated in Figure 3 and the key methods used for communication using connectionless sockets are summarized in Table 1.

6.0 Example of Socket Programs:

6.1 Write a Socket program to implement echo client and Echo server

Client Program:-

```
import java.net.*;
import java.io.*;
public class DateClient
{
        public static void main(String args[])
        {
          try{
                Socket s=new Socket("127.0.0.1",1111);


                BufferedReader in=new BufferedReader(new InputStreamReader(s.getInputStream()));
                BufferedReader stdin=new BufferedReader(new InputStreamReader(System.in));
                PrintStream out=new PrintStream(s.getOutputStream());

                System.out.println("Enter your name to echo...:");
                String str=stdin.readLine()

                out.println(str);
                str=in.readLine();

                System.out.println("Client catched from server:"+str);
                s.close();

            }catch(Exception e){}
        }
}
```

Server Program:-

```
import java.net.*;
import java.io.*;
import java.util.*;
public class DateServer
{
        public static void main(String args[])
        {
          try{
                ServerSocket ss=new ServerSocket(1111);
                Socket s=ss.accept();

                BufferedReader in=new BufferedReader(new InputStreamReader(s.getInputStream()));
                PrintStream out=new PrintStream(s.getOutputStream());


                String str=in.readLine();
                System.out.println("Server Got this...:"+str);

                out.println("Hello "+str);
                s.close();
                ss.close();
           }catch(Exception e){}
        }
}
```

6.2 Write a Socket program to show the system date and time of a server machine.

Client Program:-

```
import java.net.*;
import java.io.*;
public class DateClient
{
        public static void main(String args[])
        {
          try{
                Socket s=new Socket("127.0.0.1",1111);


                BufferedReader in=new BufferedReader(new InputStreamReader(s.getInputStream()));
                BufferedReader stdin=new BufferedReader(new InputStreamReader(System.in));
                PrintStream out=new PrintStream(s.getOutputStream());

                System.out.println("Enter your name to echo...:");
                String str=stdin.readLine();
                out.println(str);
                str=in.readLine();

                System.out.println("Client catched from server:"+str);
                s.close();

           }catch(Exception e){}
        }
}
```

Server Program:-

```
import java.net.*;
import java.io.*;
import java.util.*;
public class DateServer
{
        public static void main(String args[])
        {
          try{
                Date d=new Date();
                //String a=InetAddress.getLocalHost();
                ServerSocket ss=new ServerSocket(1111);
                Socket s=ss.accept();

                BufferedReader in=new BufferedReader(new InputStreamReader(s.getInputStream()));
                PrintStream out=new PrintStream(s.getOutputStream());


                String str=in.readLine();
                System.out.println("Server Got this...:"+str);

                out.println("Hello "+str+"The day"+d);
                s.close();
                ss.close();
           }catch(Exception e){}
        }
}
```

6.3 Write a Socket program to convert lowercase to uppercase letters.

Client program:-

```
import java.net.*;
import java.io.*;
public class UClient
{
        public static void main(String args[])
        {
          try{
                Socket s=new Socket("127.0.0.1",1111);


                BufferedReader in=new BufferedReader(new InputStreamReader(s.getInputStream()));
                BufferedReader stdin=new BufferedReader(new InputStreamReader(System.in));
                PrintStream out=new PrintStream(s.getOutputStream());

                System.out.println("Enter your name to echo...:");
                String str=stdin.readLine();



                out.println(str);
                str=in.readLine();

                System.out.println("Client catched from server:"+str);
                s.close();

           }catch(Exception e){}
        }
}
```

Server program:-

```
import java.net.*;
import java.io.*;
public class UServer
{
        public static void main(String args[])
        {
          try{
                String str;
                ServerSocket ss=new ServerSocket(1111);
                Socket s=ss.accept();

                BufferedReader in=new BufferedReader(new InputStreamReader(s.getInputStream()));
                PrintStream out=new PrintStream(s.getOutputStream());


                str=in.readLine();
                System.out.println("Server Got this...:"+str);

                String uStr=str.toUpperCase();
                out.println("Hello "+uStr);
                s.close();
                s.close();
            }catch(Exception e){}
        }
}
```

6.4 Code for a Client Program that Requests a Server to Add Integers (from 1 to a Count value) and Return the Sum

```
import java.io.*;
import java.net.*;
class summationClient{
public static void main(String[ ] args){
try{
InetAddress serverHost = InetAddress.getByName(args[0]);
int serverPort = Integer.parseInt(args[1]);
long startTime = System.currentTimeMillis( );
int count = Integer.parseInt(args[2]);
Socket clientSocket = new Socket(serverHost, serverPort);
PrintStream ps = new PrintStream(clientSocket.getOutputStream());
ps.println(count);
BufferedReader br = new BufferedReader(new InputStreamReader(clientSocket.getInputStream()));
int sum = Integer.parseInt(br.readLine());
System.out.println(" sum = "+sum);
long endTime = System.currentTimeMillis();
System.out.println(" Time consumed for receiving the feedback from the server:
"+(endTime-startTime)+" milliseconds");
clientSocket.close( );
}
catch(Exception e){e.printStackTrace( );}
}
}
```

Code for an Iterative Server that Adds (from 1 to a Count value) and Returns the Sum

```
import java.io.*;
import java.net.*;
class summationServer{
public static void main(String[] args){
try{
int serverPort = Integer.parseInt(args[0]);
ServerSocket calcServer = new ServerSocket(serverPort);
while (true){
Socket clientSocket = calcServer.accept( );
BufferedReader br = new BufferedReader(new
InputStreamReader(clientSocket.getInputStream( )));
int count = Integer.parseInt(br.readLine( ));
int sum = 0;
for (int ctr = 1; ctr <= count; ctr++){
sum += ctr;
Thread.sleep(200);
}
PrintStream ps = new PrintStream(clientSocket.getOutputStream( ));
ps.println(sum);
ps.flush( );
clientSocket.close( );
}
}
catch(Exception e){e.printStackTrace( );}
}
}
```

6.5 Write a socket program to implement broadcast socket.

```
public class Client
{
   private String hostname= "localhost";
   private int port=1234;
   private InetAddress host;
   private DatagramSocket socket;
   DatagramPacket packet;

   public void run()
   {
      try
      {
         host = InetAddress.getByName(hostname);
         socket = new DatagramSocket (null);
         packet=new DatagramPacket (new byte[100], 0,host, port);
         socket.send (packet);
         packet.setLength(100);
         socket.receive (packet);
         socket.close ();
         byte[] data = packet.getData ();
         String time=new String(data);  // convert byte array data into string
System.out.println(time);
      }
      catch(Exception e)
      {
         e.printStackTrace();
      }
   }
}


public class Server
{
   public static final int DEFAULT_PORT = 1234;
   private DatagramSocket socket;
   private DatagramPacket packet;
public void run()
   {
      try
      {
         socket = new DatagramSocket(DEFAULT_PORT);
      }
      catch( Exception ex )
      {
         System.out.println("Problem creating socket on port: " + DEFAULT_PORT );
      }

      packet = new DatagramPacket (new byte[1], 1);

      while (true)
      {
         try
         {
            socket.receive (packet);
            System.out.println("Received from: " + packet.getAddress () + ":" +
                        packet.getPort ());
            byte[] outBuffer = new java.util.Date ().toString ().getBytes ();
            packet.setData (outBuffer);
            packet.setLength (outBuffer.length);
            socket.send (packet);
         }
         catch (IOException ie)
         {
            ie.printStackTrace();
         }
      }
   }
```

6.6 Program to implement Multicast Socket.

```
import java.io.*;
import java.net.*;
class multicastSender{
public static void main(String[ ] args){
try{
InetAddress group = InetAddress.getByName("224.0.0.1");
MulticastSocket multicastSock = new MulticastSocket(3456);
String msg = "Hello How are you?";
DatagramPacket packet = new DatagramPacket(msg.getBytes( ), msg.length( ), group,
3456);
multicastSock.send(packet);
multicastSock.close( );
}
catch(Exception e){e.printStackTrace( );}
}
}
import java.io.*;
import java.net.*;
class multicastReceiver{
public static void main(String[ ] args){
try{
InetAddress group = InetAddress.getByName("224.0.0.1");
MulticastSocket multicastSock = new MulticastSocket(3456);
multicastSock.joinGroup(group);
byte[ ] buffer = new byte[100];
DatagramPacket packet = new DatagramPacket(buffer, buffer.length);
multicastSock.receive(packet);
System.out.println(new String(buffer));
multicastSock.close( );
}
catch(Exception e){e.printStackTrace( );}
}
}
```

6.7 STOP & WAIT PROTOCOL USING SOCKETS IN JAVA

```
//SENDER PROGRAM

import java.io.*;
import java.net.*;
public class Sender{
Socket sender;
ObjectOutputStream out;
ObjectInputStream in;
String packet,ack,str, msg;
int n,i=0,sequence=0;
Sender(){}
public void run(){
 try{
 BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
System.out.println("Waiting for Connection....");
 sender = new Socket("localhost",2004);
sequence=0;

out=new ObjectOutputStream(sender.getOutputStream());
out.flush();
 in=new ObjectInputStream(sender.getInputStream());
 str=(String)in.readObject();
 System.out.println("reciver     > "+str);
System.out.println("Enter the data to send....");
packet=br.readLine();
n=packet.length();
do{
```

```
try{
 if(i<n){
msg=String.valueOf(sequence);
msg=msg.concat(packet.substring(i,i+1));
}
else if(i==n){
 msg="end";out.writeObject(msg);break;
}
out.writeObject(msg);
sequence=(sequence==0)?1:0;
out.flush();
System.out.println("data sent>"+msg);
ack=(String)in.readObject();
System.out.println("waiting for ack.....\n\n");
if(ack.equals(String.valueOf(sequence))){
i++;
System.out.println("receiver   >  "+" packet recieved\n\n");
}
else{
System.out.println("Time out resending data....\n\n");
sequence=(sequence==0)?1:0;
}
}catch(Exception e){}
}while(i<n+1);
System.out.println("All data sent. exiting.");
}catch(Exception e){}
finally{
try{
in.close();
out.close();
sender.close();
}
catch(Exception e){}
}
}
public static void main(String args[]){
Sender s=new Sender();
s.run();
}
}

//RECEIVER PROGRAM

import java.io.*;
import java.net.*;
public class Reciever{
ServerSocket reciever;
Socket connection=null;
ObjectOutputStream out;
```

```
ObjectInputStream in;
String packet,ack,data="";
int i=0,sequence=0;
Reciever(){}
public void run(){
try{
BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
reciever = new ServerSocket(2004,10);
System.out.println("waiting for connection...");
connection=reciever.accept();
sequence=0;
System.out.println("Connection established   :");
out=new ObjectOutputStream(connection.getOutputStream());
out.flush();
in=new ObjectInputStream(connection.getInputStream());
out.writeObject("connected    .");
do{
try{
packet=(String)in.readObject();
if(Integer.valueOf(packet.substring(0,1))==sequence){
data+=packet.substring(1);
sequence=(sequence==0)?1:0;
System.out.println("\n\nreceiver        >"+packet);
}
else
{
System.out.println("\n\nreceiver        >"+packet +"   duplicate data");
}
if(i<3){
out.writeObject(String.valueOf(sequence));i++;
}
else{
out.writeObject(String.valueOf((sequence+1)%2));
i=0;
}
}
catch(Exception e){}
}while(!packet.equals("end"));
System.out.println("Data recived="+data);
out.writeObject("connection ended    .");
}
catch(Exception e){}
finally{
try{
in.close();
out.close();
reciever.close();
}
catch(Exception e){}
}
}
public static void main(String args[]){
Reciever s=new Reciever();
while(true){
s.run();
}
}
}
```

6.8 Sliding Window Protocal

```
//SENDER PROGRAM

import java.net.*;
import java.io.*;
import java.rmi.*;
public class slidsender
{
public static void main(String a[])throws Exception
{
ServerSocket ser=new ServerSocket(10);
Socket s=ser.accept();
DataInputStream in=new DataInputStream(System.in);
DataInputStream in1=new DataInputStream(s.getInputStream());
String sbuff[]=new String[8];
PrintStream p;
int sptr=0,sws=8,nf,ano,i;
String ch;
do
{
p=new PrintStream(s.getOutputStream());
System.out.print("Enter the no. of frames : ");
nf=Integer.parseInt(in.readLine());
p.println(nf);
if(nf<=sws-1)
{

System.out.println("Enter "+nf+" Messages to be send\n");
for(i=1;i<=nf;i++)
{
sbuff[sptr]=in.readLine();
p.println(sbuff[sptr]);
sptr=++sptr%8;
}
sws-=nf;
System.out.print("Acknowledgment received");
ano=Integer.parseInt(in1.readLine());
System.out.println(" for "+ano+" frames");
sws+=nf;
}
else
{
System.out.println("The no. of frames exceeds window size");
break;
}
System.out.print("\nDo you wants to send some more frames : ");
ch=in.readLine(); p.println(ch);
}
while(ch.equals("yes"));
s.close();
}
}

//RECEIVER PROGRAM

import java.net.*;
import java.io.*;
class slidreceiver
{
public static void main(String a[])throws Exception
{
Socket s=new Socket(InetAddress.getLocalHost(),10);
 DataInputStream in=new DataInputStream(s.getInputStream());
PrintStream p=new PrintStream(s.getOutputStream());
int i=0,rptr=-1,nf,rws=8;
String rbuf[]=new String[8];
String ch; System.out.println();
do
{
nf=Integer.parseInt(in.readLine());
if(nf<=rws-1)
{
for(i=1;i<=nf;i++)
{
rptr=++rptr%8;
rbuf[rptr]=in.readLine();
System.out.println("The received Frame " +rptr+" is : "+rbuf[rptr]);
}
rws-=nf;
System.out.println("\nAcknowledgment sent\n");
p.println(rptr+1); rws+=nf; }
else
break;
ch=in.readLine();
}
while(ch.equals("yes"));
}
}
```

7.0 DNS Server

DNS, **D**omain **N**ame **S**ystem, translates hostnames or URLs into IP addresses. For example if we type www.unixmen.com in browser, the DNS server translates the domain name into its associated ip address. Since the IP addresses are hard to remember, DNS servers are used to translate the hostnames like **www.unixmen.com** to **173.xxx.xx.xxx**. So it makes easy to remember the domain names instead of its IP address.

Scenario

Primary(Master) DNS Server Details:

```
Operating System     : CentOS 6.5 server
Hostname            : masterdns.unixmen.local
IP Address          : 192.168.1.100/24
```

Secondary(Slave) DNS Server Details:

```
Operating System     : CentOS 6.5 server
Hostname            : secondarydns.unixmen.local
IP Address          : 192.168.1.101/24
```

Client Details:

```
Operating System     : CentOS 6.5 Desktop
Hostname            : Client.unixmen.local
IP Address          : 192.168.1.102/24
```

Setup Primary (Master) DNS Server

```
[root@masterdns ~]# yum install bind* -y
```

1. Configure DNS Server

Add the lines as shown below in **'/etc/named.conf'** file

```
[root@masterdns ~]# vi /etc/named.conf
//
// named.conf
//
// Provided by Red Hat bind package to configure the ISC BIND named(8) DNS
// server as a caching only nameserver (as a localhost DNS resolver only).
//
// See /usr/share/doc/bind*/sample/ for example named configuration files.
//
options {
listen-on port 53 { 127.0.0.1; 192.168.1.100; }; ### Master DNS IP ###
listen-on-v6 port 53 { ::1; };
directory "/var/named";
dump-file "/var/named/data/cache_dump.db";
      statistics-file "/var/named/data/named_stats.txt";
      memstatistics-file "/var/named/data/named_mem_stats.txt";
allow-query     { localhost; 192.168.1.0/24; }; ### IP Range ###
allow-transfer{ localhost; 192.168.1.101; };   ### Slave DNS IP ###
recursion yes;
dnssec-enable yes;
dnssec-validation yes;
dnssec-lookaside auto;
/* Path to ISC DLV key */
bindkeys-file "/etc/named.iscdlv.key";
managed-keys-directory "/var/named/dynamic";
};
logging {
      channel default_debug {
            file "data/named.run";
            severity dynamic;
      };
};
zone "." IN {
type hint;
file "named.ca";
};
zone"unixmen.local" IN {
type master;
file "forward.unixmen";
allow-update { none; };
};
zone"1.168.192.in-addr.arpa" IN {
type master;
file "reverse.unixmen";
allow-update { none; };
};
include "/etc/named.rfc1912.zones";
include "/etc/named.root.key";
```

2. Create Zone files

Create forward and reverse zone files which we mentioned in the **'/etc/named.conf'** file.

2.1 Create Forward Zone

Create forward.unixmen file in the **'/var/named'** directory.

```
[root@masterdns ~]# vi /var/named/forward.unixmen
$TTL 86400
@   IN  SOA     masterdns.unixmen.local. root.unixmen.local. (
      2011071001  ;Serial
      3600        ;Refresh
      1800        ;Retry
      604800      ;Expire
      86400       ;Minimum TTL
)
```

```
@       IN  NS         masterdns.unixmen.local.
@       IN  NS         secondarydns.unixmen.local.
@       IN  A         192.168.1.100
@       IN  A         192.168.1.101
@       IN  A         192.168.1.102
masterdns       IN  A   192.168.1.100
secondarydns    IN  A   192.168.1.101
client          IN  A   192.168.1.102
```

2.2 Create Reverse Zone

Create reverse.unixmen file in the **'/var/named'** directory.

```
[root@masterdns ~]# vi /var/named/reverse.unixmen
$TTL 86400
@   IN  SOA     masterdns.unixmen.local. root.unixmen.local. (
        2011071001  ;Serial
        3600        ;Refresh
        1800        ;Retry
        604800      ;Expire
        86400       ;Minimum TTL
)
@       IN  NS         masterdns.unixmen.local.
@       IN  NS         secondarydns.unixmen.local.
@       IN  PTR        unixmen.local.
masterdns       IN  A   192.168.1.100
secondarydns    IN  A   192.168.1.101
client          IN  A   192.168.1.102
100     IN  PTR        masterdns.unixmen.local.
101     IN  PTR        secondarydns.unixmen.local.
102     IN  PTR        client.unixmen.local.
```

3. Start the DNS service

```
[root@masterdns ~]# service named start
Starting named:                                   [  OK  ]
[root@masterdns ~]# chkconfig named on
```

4. Adjust iptables to allow DNS server from outside of the network

Add the lines as shown below in **'/etc/sysconfig/iptables'** file.

```
[root@masterdns ~]# vi /etc/sysconfig/iptables
# Firewall configuration written by system-config-firewall
# Manual customization of this file is not recommended.
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -p udp -m state --state NEW --dport 53 -j ACCEPT
-A INPUT -p tcp -m state --state NEW --dport 53 -j ACCEPT
-A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT
-A INPUT -j REJECT --reject-with icmp-host-prohibited
```

```
-A FORWARD -j REJECT --reject-with icmp-host-prohibited
COMMIT
```

5. Restart iptables

```
[root@masterdns ~]# service iptables restart
iptables: Flushing firewall rules:                    [  OK  ]
iptables: Setting chains to policy ACCEPT: filter         [  OK  ]
iptables: Unloading modules:                          [  OK  ]
iptables: Applying firewall rules:                     [  OK  ]
```

6. Test DNS configuration and zone files for any syntax errors

```
[root@masterdns ~]# named-checkconf /etc/named.conf
[root@masterdns ~]# named-checkzone unixmen.local /var/named/forward.unixmen
zone unixmen.local/IN: loaded serial 2011071001
OK
[root@masterdns ~]# named-checkzone unixmen.local /var/named/reverse.unixmen
zone unixmen.local/IN: loaded serial 2011071001
OK
```

7. Test DNS Server

```
[root@masterdns ~]# dig masterdns.unixmen.local
; <<>> DiG 9.8.2rc1-RedHat-9.8.2-0.10.rc1.el6_3.6 <<>> masterdns.unixmen.local
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 49834
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 1
;; QUESTION SECTION:
;masterdns.unixmen.local.INA
;; ANSWER SECTION:
masterdns.unixmen.local. 86400INA192.168.1.100
;; AUTHORITY SECTION:
unixmen.local.86400INNSsecondarydns.unixmen.local.
unixmen.local.86400INNSmasterdns.unixmen.local.
;; ADDITIONAL SECTION:
secondarydns.unixmen.local. 86400 INA192.168.1.101
;; Query time: 6 msec
;; SERVER: 192.168.1.100#53(192.168.1.100)
;; WHEN: Thu Mar  7 13:07:56 2013
;; MSG SIZE  rcvd: 114
[root@masterdns ~]# nslookup unixmen.local
Server:192.168.1.100
Address:192.168.1.100#53
Name:unixmen.local
Address: 192.168.1.102
Name:unixmen.local
Address: 192.168.1.100
Name:unixmen.local
Address: 192.168.1.101
```

Now the Primary DNS server is ready to use.

Setup Secondary(Slave) DNS Server

```
[root@secondarydns ~]# yum install bind* -y
```

1. Configure Slave DNS Server

Open the main configuration file **'/etc/named.conf'** and add the lines as shown below.

```
[root@secondarydns ~]# vi /etc/named.conf
//
// named.conf
//
// Provided by Red Hat bind package to configure the ISC BIND named(8) DNS
// server as a caching only nameserver (as a localhost DNS resolver only).
//
// See /usr/share/doc/bind*/sample/ for example named configuration files.
//
options {
listen-on port 53 { 127.0.0.1; 192.168.1.101; };
listen-on-v6 port 53 { ::1; };
directory "/var/named";
dump-file "/var/named/data/cache_dump.db";
        statistics-file "/var/named/data/named_stats.txt";
        memstatistics-file "/var/named/data/named_mem_stats.txt";
allow-query     { localhost; 192.168.1.0/24; };
recursion yes;
dnssec-enable yes;
dnssec-validation yes;
dnssec-lookaside auto;
/* Path to ISC DLV key */
bindkeys-file "/etc/named.iscdlv.key";
managed-keys-directory "/var/named/dynamic";
};
logging {
        channel default_debug {
                file "data/named.run";
                severity dynamic;
        };
};
zone "." IN {
type hint;
file "named.ca";
};
zone"unixmen.local" IN {
type slave;
file "slaves/unixmen.fwd";
masters { 192.168.1.100; };
};
zone"1.168.192.in-addr.arpa" IN {
type slave;
file "slaves/unixmen.rev";
masters { 192.168.1.100; };
};
include "/etc/named.rfc1912.zones";
include "/etc/named.root.key";
```

2. Start the DNS Service

```
[root@secondarydns ~]# service named start
Generating /etc/rndc.key:                          [  OK  ]
Starting named:                                    [  OK  ]
[root@secondarydns ~]# chkconfig named on
```

Now the forward and reverse zones are automatically replicated from Master DNS server to **'/var/named/slaves/'** in Secondary DNS server.

```
[root@secondarydns ~]# ls /var/named/slaves/
unixmen.fwd  unixmen.rev
[root@secondarydns ~]# cat /var/named/slaves/unixmen.fwd
$ORIGIN .
$TTL 86400; 1 day
unixmen.localIN SOAmasterdns.unixmen.local. root.unixmen.local. (
2011071001 ; serial
3600      ; refresh (1 hour)
1800      ; retry (30 minutes)
604800    ; expire (1 week)
86400     ; minimum (1 day)
)
NS masterdns.unixmen.local.
NS secondarydns.unixmen.local.
A192.168.1.100
A192.168.1.101
A192.168.1.102
$ORIGIN unixmen.local.
clientA192.168.1.102
masterdnsA192.168.1.100
secondarydnsA192.168.1.101
[root@secondarydns ~]# cat /var/named/slaves/unixmen.rev
$ORIGIN .
$TTL 86400; 1 day
1.168.192.in-addr.arpaIN SOAmasterdns.unixmen.local. root.unixmen.local. (
2011071001 ; serial
3600      ; refresh (1 hour)
1800      ; retry (30 minutes)
604800    ; expire (1 week)
86400     ; minimum (1 day)
)
NS masterdns.unixmen.local.
NS secondarydns.unixmen.local.
PTRunixmen.local.
$ORIGIN 1.168.192.in-addr.arpa.
100PTRmasterdns.unixmen.local.
101PTRsecondarydns.unixmen.local.
102PTRclient.unixmen.local.
clientA192.168.1.102
masterdnsA192.168.1.100
secondarydnsA192.168.1.101
```

3. Add the DNS Server details to all systems

```
[root@secondarydns ~]# vi /etc/resolv.conf
# Generated by NetworkManager
search ostechnix.com
nameserver 192.168.1.100
nameserver 192.168.1.101
nameserver 8.8.8.8
```

4. Test DNS Server

```
[root@secondarydns ~]# dig masterdns.unixmen.local
; <<>> DiG 9.8.2rc1-RedHat-9.8.2-0.10.rc1.el6_3.6 <<>> masterdns.unixmen.local
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 21487
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 1
;; QUESTION SECTION:
;masterdns.unixmen.local.INA
;; ANSWER SECTION:
masterdns.unixmen.local. 86400INA192.168.1.100
;; AUTHORITY SECTION:
unixmen.local.86400INNSmasterdns.unixmen.local.
unixmen.local.86400INNSsecondarydns.unixmen.local.
;; ADDITIONAL SECTION:
secondarydns.unixmen.local. 86400 INA192.168.1.101
;; Query time: 15 msec
;; SERVER: 192.168.1.100#53(192.168.1.100)
;; WHEN: Thu Mar  7 13:27:57 2013
;; MSG SIZE  rcvd: 114
[root@secondarydns ~]# dig secondarydns.unixmen.local
; <<>> DiG 9.8.2rc1-RedHat-9.8.2-0.10.rc1.el6_3.6 <<>> secondarydns.unixmen.local
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 20958
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 1
;; QUESTION SECTION:
;secondarydns.unixmen.local.INA
;; ANSWER SECTION:
secondarydns.unixmen.local. 86400 INA192.168.1.101
;; AUTHORITY SECTION:
unixmen.local.86400INNSmasterdns.unixmen.local.
unixmen.local.86400INNSsecondarydns.unixmen.local.
;; ADDITIONAL SECTION:
masterdns.unixmen.local. 86400INA192.168.1.100
;; Query time: 4 msec
;; SERVER: 192.168.1.100#53(192.168.1.100)
;; WHEN: Thu Mar  7 13:31:53 2013
;; MSG SIZE  rcvd: 114
[root@secondarydns ~]# nslookup unixmen.local
Server:192.168.1.100
Address:192.168.1.100#53
Name:unixmen.local
Address: 192.168.1.101
Name:unixmen.local
Address: 192.168.1.102
Name:unixmen.local
Address: 192.168.1.100
```

Client Side Configuration

Add the DNS server details in **'/etc/resolv.conf'** file in all client systems

```
[root@client unixmen]# vi /etc/resolv.conf
# Generated by NetworkManager
search unixmen.local
nameserver 192.168.1.100
nameserver 192.168.1.101
nameserver 8.8.8.8
```

Test DNS Server

```
[root@client unixmen]# dig masterdns.unixmen.local
; <<>> DiG 9.8.2rc1-RedHat-9.8.2-0.10.rc1.el6 <<>> masterdns.unixmen.local
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 19496
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 1
;; QUESTION SECTION:
;masterdns.unixmen.local.INA
;; ANSWER SECTION:
masterdns.unixmen.local. 86400INA192.168.1.100
;; AUTHORITY SECTION:
unixmen.local.86400INNSmasterdns.unixmen.local.
unixmen.local.86400INNSsecondarydns.unixmen.local.
;; ADDITIONAL SECTION:
secondarydns.unixmen.local. 86400 INA192.168.1.101
;; Query time: 30 msec
;; SERVER: 192.168.1.100#53(192.168.1.100)
;; WHEN: Thu Mar  7 13:47:55 2013
;; MSG SIZE  rcvd: 114
[root@client unixmen]# dig secondarydns.unixmen.local
; <<>> DiG 9.8.2rc1-RedHat-9.8.2-0.10.rc1.el6 <<>> secondarydns.unixmen.local
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 14852
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 1
;; QUESTION SECTION:
;secondarydns.unixmen.local.INA
;; ANSWER SECTION:
secondarydns.unixmen.local. 86400 INA192.168.1.101
;; AUTHORITY SECTION:
unixmen.local.86400INNSsecondarydns.unixmen.local.
unixmen.local.86400INNSmasterdns.unixmen.local.
;; ADDITIONAL SECTION:
masterdns.unixmen.local. 86400INA192.168.1.100
;; Query time: 8 msec
;; SERVER: 192.168.1.100#53(192.168.1.100)
;; WHEN: Thu Mar  7 13:48:38 2013
;; MSG SIZE  rcvd: 114
[root@client unixmen]# dig client.unixmen.local
; <<>> DiG 9.8.2rc1-RedHat-9.8.2-0.10.rc1.el6 <<>> client.unixmen.local
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 14604
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 2
;; QUESTION SECTION:
;client.unixmen.local.INA
;; ANSWER SECTION:
client.unixmen.local.86400INA192.168.1.102
;; AUTHORITY SECTION:
unixmen.local.86400INNSmasterdns.unixmen.local.
unixmen.local.86400INNSsecondarydns.unixmen.local.
;; ADDITIONAL SECTION:
masterdns.unixmen.local. 86400INA192.168.1.100
secondarydns.unixmen.local. 86400 INA192.168.1.101
;; Query time: 5 msec
;; SERVER: 192.168.1.100#53(192.168.1.100)
;; WHEN: Thu Mar  7 13:49:11 2013
;; MSG SIZE  rcvd: 137
[root@client unixmen]# nslookup unixmen.local
Server:192.168.1.100
Address:192.168.1.100#53
Name:unixmen.local
Address: 192.168.1.102
Name:unixmen.local
Address: 192.168.1.100
Name:unixmen.local
Address: 192.168.1.101
```

Now the primary and secondary DNS servers are ready.

8.0 Apache Web Server

The *Apache HTTP Server* (Apache) is an open-source web server application. This guide explains how to install and configure an Apache web server on CentOS 6.

This guide is written for a non-root user. Commands that require elevated privileges are prefixed with `sudo`. If you're not familiar with the `sudo` command, you can check our Users and Groups guide.

Before You Begin

1. Ensure that you have followed the Getting Started and Securing Your Server guides, and the Linode's hostname is set.

   To check your hostname run:

   ```
   hostname
   hostname -f
   ```

   The first command should show your short hostname, and the second should show your fully qualified domain name (FQDN).

2. Update your system:

   ```
   sudo yum update
   ```

Install Apache

1. Install the Apache HTTP Server:

   ```
   sudo yum install httpd
   ```

2. Edit the main Apache configuration file to adjust the resource use settings. The settings shown below are a good starting point for a **Linode 2GB**:

   /etc/httpd/conf/httpd.conf

   ```
   KeepAliveOff


   <IfModuleprefork.c>
   StartServers4
   MinSpareServers20
   MaxSpareServers40
   MaxClients200
   MaxRequestsPerChild4500
   </IfModule>
   ```

Configure Apache for Virtual Hosting

1. Create a file under /etc/httpd/conf.d named vhost.conf. Replace instances of example.com with your own domain information:

   /etc/httpd/conf.d/vhost.conf

   ```
   <VirtualHost *:80>
   ServerAdminadmin@example.org
   ServerNameexample.org
   ServerAliaswww.example.org
   DocumentRoot /srv/www/example.org/public_html/
   ErrorLog /srv/www/example.org/logs/error.log
   CustomLog /srv/www/example.org/logs/access.logcombined
   </VirtualHost>
   ```

   Additional virtual host blocks can be added to the file for any other domains you wish to host on the Linode.

2. Create the directories referenced above:

   ```
   sudo mkdir -p /srv/www/example.org/public_html
   sudo mkdir -p /srv/www/example.org/logs
   ```

3. Start Apache:

   ```
   sudo service httpd start
   ```

4. Set Apache to start at boot:

   ```
   sudo chkconfig httpd on
   ```

Apache Mods and Scripting

Install Apache Modules

By default, modules are located in the /etc/httpd/modules/ directory. Configuration directives for the default modules are located in /etc/httpd/conf/httpd.conf, while configuration options for optional modules installed with yum are generally placed in .conf files in /etc/httpd/conf.d/.

1. List available Apache modules:

   ```
   sudo yum search mod_
   ```

2. Install any desired modules:

   ```
   sudo yum install mod_[module-name]
   ```

3. Modules should be enabled and ready to use following installation

Install Support for Scripting

The following commands install Apache support for server-side scripting in PHP, Python, and Perl. Support for these languages is optional based on your server environment.

To install:

- Perl support:

```
sudo yum install mod_perl
```

- Python support:

```
sudo yum install mod_wsgi
```

- PHP support:

```
sudo yum install php php-pear
```

Network Lab Assignments

1) Familiarization with networking devices like Hub, Switch etc.
2) Write the procedure of NIC installation.
3) Create Straight cable, Cross cable , Console cable using crimping tools.
4) Write down the steps to configure IP address of a machine (Windows'98/2000/XP/Linux).
5) Implement the following Network Commands.
    a) ping
    b) ifconfig
    c) traceroute
    d) telnet
    e) ftp
6) Develop a program which can send and receive messages using Message Queue.
7) Implement a program that can retrieve the IP address of any host.
8) Write a Socket program to show the time of the system date and time of a server machine in a Client machine. (Time Server)
9) Write a Socket program to send a message from client machine to server machine and reply back the same message from server to client.
10) Write a Socket program which will take numbers as input from client machine and send it to server machine where calculation would be done and sent the result to the client machine.
11) Write a Socket program, which takes domain name as input and displays corresponding address if the domain is found otherwise, shows an error message.
12) Write down TCP client and server program.
13) Write a socket program to implement Stop-and-Wait ARQ.

APPENDIX A

GUIDELINES FOR PREPARING LABORATORY REPORTS

In the following, general guidelines are given to help the CS 291 laboratory students to write well-prepared laboratory reports. A good laboratory report must have the following items:

1. A front page: this page should contain:
    a. Name of the Institute.
    b. Name of the Department.
    c. Paper Code.
    d. Paper Name.
    e. Assignment <part_name>
    f. Student's name
    g. Student's roll_no.
    h. Dept./Group name
    i. Date of Submission
    j. Sign. of teacher
    k. Date

2. Contents inside the Laboratory Manual.
    a. Assignment Type: ………………
    b. Assignment No.: …………………
    c. Problem Statement
    d. Algorithm and Flow Chart
    e. Program Listing
    f. Input/Output
    g. Remarks

Note: <u>Take extra care to write the report in proper language using your own words. Poor English gives a poor impression and may cause the laboratory instructor not to understand the full meaning of the idea that you want to convey. This may also result in lowering the grade of your report.</u>

A sample report is shown next. Examine this report and check if the abovementioned items have been properly implemented.

APPENDIX B

Glossary

10Base2 - Ethernet specification for thin coaxial cable, transmits signals at 10 Mbps (megabits per second) with a distance limit of 185 meters per segment.

10Base5 - Ethernet specification for thick coaxial cable, transmits signals at 10 Mbps (megabits per second) with a distance limit of 500 meters per segment.

10BaseF - Ethernet specification for fiber optic cable, transmits signals at 10 Mbps (megabits per second) with a distance limit of 2000 meters per segment.

10BaseT - Ethernet specification for unshielded twisted pair cable (category 3, 4, or 5), transmits signals at 10 Mbps (megabits per second) with a distance limit of 100 meters per segment.

100BaseT - Ethernet specification for unshielded twisted pair cabling that is used to transmit data at 100 Mbps (megabits per second) with a distance limit of 100 meters per segment.

1000BaseTX -Ethernet specification for unshielded twisted pair cabling that is used to trasmit data at 1 Gbps (gigabits per second) with a distance limitation of 220 meters per segment.

Asynchronous Transfer Mode (ATM) - A network protocol that transmits data at a speed of 155 Mbps and higher. It is most often used to interconnect two or more local area networks.

AppleTalk - Apple Computer's network protocol originally designed to run over LocalTalk networks, but can also run on Ethernet and Token Ring.

AUI Connector (Attachment Unit Interface) - A 15 pin connector found on Ethernet cards that can be used for attaching coaxial, fiber optic, or twisted pair cable.

Backbone - A cable to which multiple nodes or workstations are attached.

Bit - Binary digit in the binary numbering system. Its value can be 0 or 1. In an 8-bit character scheme, it takes 8 bits to make a byte (character) of data.

BNC Connector (Bayone-Neill-Concelman) - Standard connector used to connect 10Base2 coaxial cable.

Bridge - Devices that connect and pass packets between two network segments that use the same communications protocol.

Cable - Transmission medium of copper wire or optical fiber wrapped in a protective cover.

Client/Server - A networking system in which one or more file servers (Server) provide services; such as network management, application and centralized data storage for workstations (Clients).

CSMA/CA - Carrier Sense Multiple Access Collision Avoidance is a network access method in which each device signals its intent to transmit before it actually does so. This prevents other devices from sending information, thus preventing collisions from occurring between signals from two or more devices. This is the access method used by LocalTalk.

CSMA/CD - Carrier Sense Multiple Access Collision Detection is a network access method in which devices that are ready to transmit data first check the channel for a carrier. If no carrier is sensed, a device can transmit. If two devices transmit at once, a collision occurs and each computer backs off and waits a random amount of time before attempting to retransmit. This is the access method used by Ethernet.

Coaxial Cable - Cable consisting of a single copper conductor in the center surrounded by a plastic layer for insulation and a braided metal outer shield.

Concentrator - A device that provides a central connection point for cables from workstations, servers, and peripherals. Most concentrators contain the ability to amplify the electrical signal they receive.

DIN - A plug and socket connector consisting of a circular pattern of pins in a metal sleeve. This type of connector is commonly seen on keyboards.

Dumb Terminal - Refers to devices that are designed to communicate exclusively with a host (main frame) computer. It receives all screen layouts from the host computer and sends all keyboard entry to the host. It cannot function without the host computer.

E-mail - An electronic mail message sent from a host computer to a remote computer.

End User - Refers to the human executing applications on the workstation.

Ethernet - A network protocol invented by Xerox Corporation and developed jointly by Xerox, Intel and Digital Equipment Corporation. Ethernet networks use CSMA/CD and run over a variety of cable types at 10 Mbps (megabits per second).

Expansion Slot - Area in a computer that accepts additional input/output boards to increase the capability of the computer.

Fast Ethernet - A new Ethernet standard that supports 100 Mbps using category 5 twisted pair or fiber optic cable.

Fiber Distributed Data Interface (FDDI) - A network protocol that is used primarily to interconnect two or more local area networks, often over large distances.

Fiber Optic Cable - A cable, consisting of a center glass core surrounded by layers of plastic, that transmits data using light rather than electricity. It has the ability to carry more information over much longer distances.

File Server - A computer connected to the network that contains primary files/applications and shares them as requested with the other computers on the network. If the file server is dedicated for that purpose only, it is connected to a client/server network. An example of a client/server network is Novell

Netware. All the computers connected to a peer-to-peer network are capable of being the file server. Two examples of peer-to-peer networks are LANtastic and Windows for Workgroups.

Gigabit Ethernet - An Ethernet protocol that raises the transmission rates to 1 Gbps (gigabits per second). It is primarily used for a high speed backbone of a network.

Gigabyte (GB) - One billion bytes of information. One thousand megabytes.

Hub - A hardware device that contains multiple independent but connected modules of network and internetwork equipment. Hubs can be active (where they repeat signals sent through them) or passive (where they do not repeat but merely split signals sent through them).

Infrared - Electromagnetic waves whose frequency range is above that of microwaves, but below that of the visible spectrum.

Intranet - Network internal to an organization that uses Internet protocols.

Internet - A global network of networks used to exchange information using the TCP/IP protocol. It allows for electronic mail and the accessing ad retrieval of information from remote sources.

LAN (Local Area Network) - A network connecting computers in a relatively small area such as a building.

Linear Bus - A network topology in which each node attaches directly to a common cable.

LocalTalk - Apple Corporation proprietary protocol that uses CSMA/CA media access scheme and supports transmissions at speeds of 230 Kbps (Kilobits per second).

MAN (Metropolitan Area Network) - A network connecting computers over a large geographical area, such as a city or school district.

MAU (Multistation Access Unit) - A Token Ring wiring hub.

Modem (Modulator/Demodulator) - Devices that convert digital and analog signals. Modems allow computer data (digital) to be transmitted over voice-grade telephone lines (analog).

Multiplexer - A device that allows multiple logical signals to be transmitted simultaneously across a single physical channel.

Network Modem - A modem connected to a Local Area Network (LAN) that is accessible from any workstation on the network.

Network Interface Card (NIC) - A board that provides network communication capabilities to and from a computer.

Network Operating System (NOS) - Operating system designed to pass information and communicate between more than one computer. Examples include AppleShare, Novell NetWare, and Windows NT Server.

Node - End point of a network connection. Nodes include any device attached to a network such as file servers, printers, or workstations.

Node Devices - Any computer or peripheral that is connected to the network.

PCMCIA - An expansion slot found in many laptop computers.

Peer-to-Peer Network - A network in which resources and files are shared without a centralized management source.

Physical Topology - The physical layout of the network; how the cables are arranged; and how the computers are connected.

Point-to-Point - A direct link between two objects in a network.

Ports - A connection point for a cable.

Protocol -A formal description of a set of rules and conventions that govern how devices on a network exchange information.

RAID (Redundant Array of Inexpensive Disks) - A configuration of multiple disks designed to preserve data after a disk casualty.

RAM (Random Access Memory) - The working memory of a computer where data and programs are temporarily stored. RAM only holds information when the computer is on.

Repeater - A device used in a network to strengthen a signal as it is passed along the network cable.

RJ-45 - Standard connectors used for unshielded twisted-pair cable.

Router -A device that routes information between interconnected networks. It can select the best path to route a message, as well as translate information from one network to another. It is similar to a superintelligent bridge.

SCSI (Small Computer Serial Interface) - An interface controller that allows several peripherals to be connected to the same port on a computer.

Segment - Refers to a section of cable on a network. In Ethernet networks, two types of segments are defined. A populated or trunk segment is a network cable that has one or more nodes attached to it. A link segment is a cable that connects a computer to an interconnecting device, such as a repeater or concentrator, or connects a interconnecting device to another interconnecting device.

Sneaker-Net - Refers to a manual method of sharing files in which a file is copied from a computer to a floppy disk, transported to a second computer by a person physically walking (apparently wearing sneakers) to the second computer, and manually transferring the file from floppy disk to the second computer.

Speed of Data Transfer - The rate at which information travels through a network, usually measured in megabits per second.

Star Topology - LAN topology in which each node on a network is connected directly to a central network hub or concentrator.

Star-Wired Ring - Network topology that connects network devices (such as computers and printers) in a complete circle.

Tape Back-Up - Copying all the data and programs of a computer system on magnetic tape. On tape, data is stored sequentially. When retrieving data, the tape is searched from the beginning of tape until the data is found.

Terminator - A device that provides electrical resistance at the end of a transmission line. Its function is to absorb signals on the line, thereby keeping them from bouncing back and being received again by the network.

Thicknet - A thick coaxial cable that is used with a 10Base5 Ethernet LAN.

Thinnet - A thin coaxial cable that is used with a 10Base2 Ethernet LAN.

Token - A special packet that contains data and acts as a messenger or carrier between each computer and device on a ring topology. Each computer must wait for the messenger to stop at its node before it can send data over the network.

Token Ring - A network protocol developed by IBM in which computers access the network through token-passing. Usually uses a star-wired ring topology.

Topology - There are two types of topology: physical and logical. The physical topology of a network refers to the configuration of cables, computers, and other peripherals. Logical topology is the method used to pass the information between workstations. Issues involving logical topologies are discussed on the Protocol chapter

Transceiver (Transmitter/Receiver) - A Device that receives and sends signals over a medium. In networks, it is generally used to allow for the connection between two different types of cable connectors, such as AUI and RJ-45.

Tree Topology - LAN topology similar to linear bus topology, except that tree networks can contain branches with multiple nodes.

Twisted Pair - Network cabling that consists of four pairs of wires that are manufactured with the wires twisted to certain specifications. Available in shielded and unshielded versions.

USB (Universal Serial Bus) Port - A hardware interface for low-speed peripherals such as the keyboard, mouse, joystick, scanner, printer, and telephony devices.

WAN (Wide Area Network) - A network connecting computers within very large areas, such as states, countries, and the world.

Workgroup - A collection of workstations and servers on a LAN that are designated to communicate and exchange data with one another.

Workstation - A computer connected to a network at which users interact with software stored on the network.

Laboratory Safety

Users` aspects:

- Every student should create his/her own directory with their group_name and roll number.
- Program name should be indicating the purpose of the program.
- In case of writing a program, after certain interval every student should save his/her file.

Physical Hazards

Adopt a good posture

A good working posture is one which can be sustained with the minimum of static muscular effort. In general, a varied working position is better than a fixed working posture. However, a working position which is static and relaxed is better than one which is static and tense.

Your upper body is most comfortable when:

- Your back is supported.
- Your head is up.
- Your upper arms are relaxed.

Your hands and wrists are most comfortable when:

- Your forearm is nearly at a right angle to your upper arm.
- Your wrist is in a straight line with your hand and forearm.

Adjust your seating position, in order to improve your posture, adjust your chair so that:

- Your lower back is supported.
- Your knees are level with your hips.
- Your feet are flat on the floor.
- Your eye level is just above the top of the screen. Tilt your screen if necessary.
- The screen is directly in front of you, not at an angle.

Type using both hands ... or better still, learn to touch type

References

1. Apache Web server manual
2. GNU Linux Administration manual
3. Java Programming, Herbert Schildt