Linux Professional Institute Japan LPI-JAPAN

LPIC Level 1 Seminar in English

2013/2/23
Carl Stevens
Zeus Learning Power Co., Ltd.

Lecturer Profile

■Company Profile

- Zeus Learning Power Co., Ltd.

  http://www.zeus-learning.jp

■Lecturer

- Belongs to the Technical Management Department
- Teaches Linux and Networking

Today’s Program

■Introduction to the LPIC Test

- About the test

■Presentations

- Topic 103.7 Regular Expressions
- Topic 104.5 File and Directory Permissions
- Topic 109.1&2 Linux Networking

The LPIC Test

Test Outline

■World Class Qualification

- Regarded worldwide as a fair evaluation of Linux ability

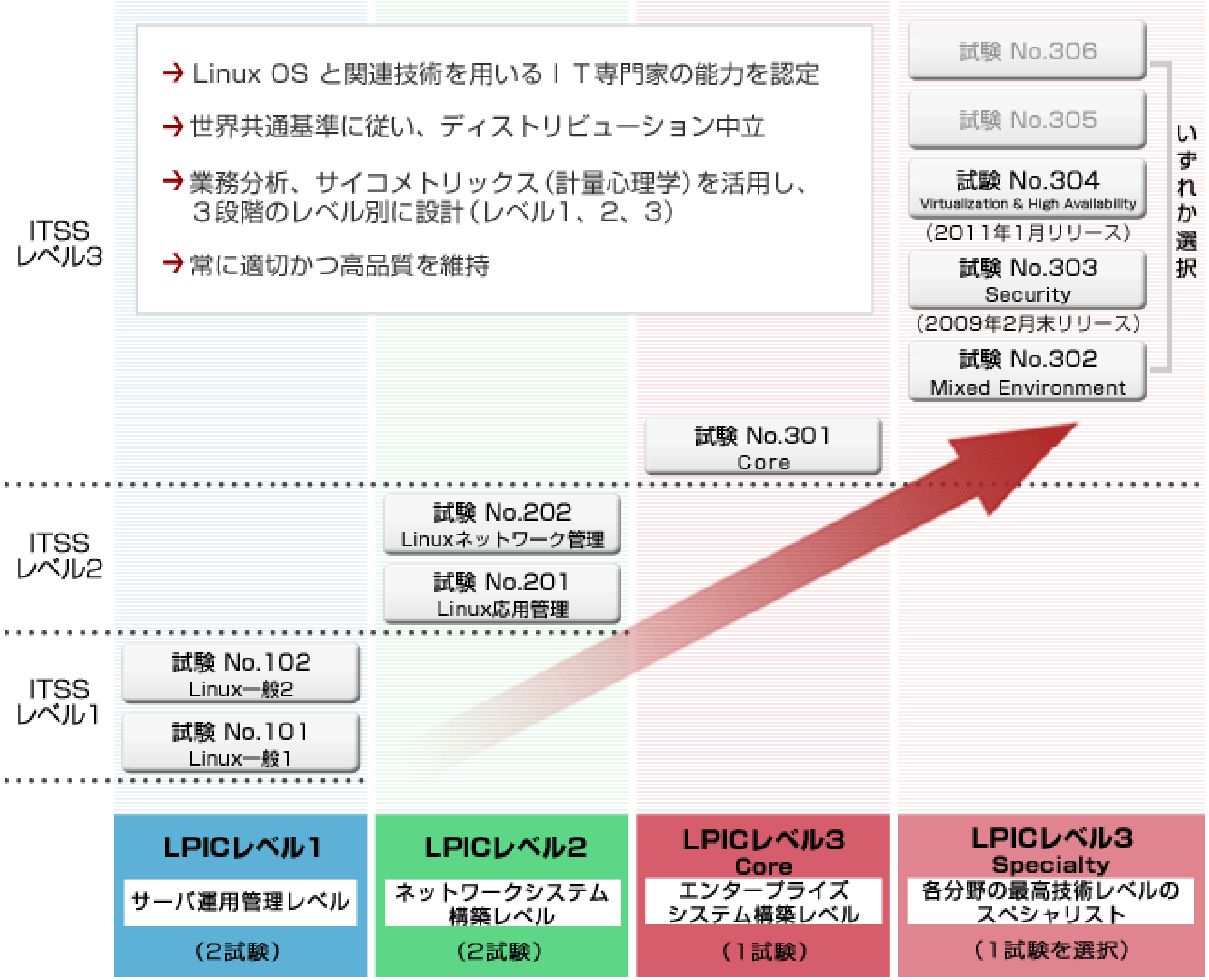
■Fair and Neutral

- Does not depend on vendor or distribution
- Evaluates Linux technical ability from a fair and neutral standpoint

■Popular Worldwide

- Over 300,000 people worldwide have taken the test with over 100,000 certified
- In Japan, over 47,000 certified Level 1, 13,000 certified Level 2, 5,000 certified Level 3 makes a total of 65,000 LPI certified

LPIC Test Details



→ Linux OS と関連技術を用いるＩＴ専門家の能力を認定

→ 世界共通基準に従い、ディストリビューション中立

→ 業務分析、サイコメトリックス（計量心理学）を活用し、３段階のレベル別に設計（レベル1、2、3）

→ 常に適切かつ高品質を維持

ITSS レベル3

- 試験 No.306
- 試験 No.305
- 試験 No.304 Virtualization & High Availability （2011年1月リリース）
- 試験 No.303 Security （2009年2月末リリース）
- 試験 No.302 Mixed Environment

いずれか選択

- 試験 No.301 Core

ITSS レベル2

- 試験 No.202 Linuxネットワーク管理
- 試験 No.201 Linux応用管理

ITSS レベル1

- 試験 No.102 Linux一般2
- 試験 No.101 Linux一般1

LPICレベル1	LPICレベル2	LPICレベル3 Core	LPICレベル3 Specialty
サーバ運用管理レベル	ネットワークシステム構築レベル	エンタープライズシステム構築レベル	各分野の最高技術レベルのスペシャリスト
（2試験）	（2試験）	（1試験）	（1試験を選択）

LPIC LEVEL 1

Certifies people as able to perform basic administrative tasks on a Linux computer

Shows one to be ready to study server set-up and maintenance

LPIC 101
Objective 103.7

Regular Expressions

What is a Regular Expression?

- A regular expression is a string which match patterns in text
- A string is a row of characters. For example: ^S.*[0-9]$
- Regular expression is often shortened to regex or regexp
- The regular expression "at" matches three words in the following text. Can you see them?

  Example text:

  Can I have your attention please?

  The atrium will be closing at three today.

- It is important to read regular expressions one character at a time, i.e. "at" is "a followed by t"

What are Regular Expressions Used For?

- A lot of data is stored in text format
- Examples: Server configuration files, web pages, data bases, plain text files
- Regular expressions allow us to search and manipulate this data with ease!

Where are Regular Expressions Used?

- Regular expressions are used by utilities, text editors and programming languages to search for and manipulate text
- Examples: grep, sed, awk, vi, LibreOffice
- … Perl, Python, Ruby, Java, data bases, etc.
- Regular expressions are not standardized
- Different tools mean different regular expressions

What are Regular Expressions Made of?

- Regular Expressions are made of literals and metacharacters
- Let’s look at each in turn . . .

Literals

- Literals are characters which have no special meaning
- "a" matches a, "1" matches 1, "-" matches -, etc.
- If I wanted to match all lines in a file containing the string "Tokyo", I could use the regular expression "Tokyo"
- Literals are the easiest to use

Metacharacters

- Metacharacters are characters which have a special meaning
- "^" means "beginning of the line", "|" means "or", etc.
- If I wanted to match all of the lines in a file which contained "Tokyo" or "tokyo", I could use the regular expression "Tokyo|tokyo"
- Metacharacters are the big hurdle to understanding and using regular expressions

The grep Command

- The name grep comes from an old regular expression syntax: g/re/p, which reads "global regular expression print"
- An extremely useful tool for extracting specific data from files
- grep searches each line of a file for a pattern and displays any lines which contain the pattern
- The syntax for grep is: grep regex file
- Example: grep root /etc/passwd will display all lines from the /etc/passwd file which contain the pattern "root"

Quoting Regular Expressions (1)

- Regular expressions often must be quoted to hide them from the shell
- This is because the shell will interpret any metacharacters in the regex before it calls grep
- For example, grep -E Tokyo|tokyo Japan will fail because the shell will interpret | as the pipe and look for the tokyo command. There is no tokyo command, so the shell will produce an error message and stop without even calling the grep command
- Quoting the regular expression: grep -E "Tokyo|tokyo" Japan will solve the problem

Quoting Regular Expressions (2)

Quotation Example	Explanation
'regex'	Single quotes: strong quotation
"regex"	Double quotes: weak quotation
¥regex	Backslash

- Single quotes hide all metacharacters from the shell
- Double quotes hide all but $var, ' ', " ", etc.
- The backslash must be placed before the metacharacter

- The following commands all work the same:

```
grep -E 'Tokyo|tokyo' Japan
grep -E "Tokyo|tokyo" Japan
grep -E Tokyo¥|tokyo Japan
```

Metacharacters (1) Anchors

Metacharacter	Explanation
^	Caret: beginning of the line
$	Dollar sign: end of the line

- File secret

```
Agent 007 is James Bond
Bond works for MI5
```

- Command 1. grep "^Bond" secret
- Command 2. grep "Bond$" secret

Metacharacters (2) Wildcard and Quantifier

Metacharacter	Explanation
.	Dot: any single character
*	Asterisk: zero or more of the preceding character

- File words

```
act
cat
cut
coat
```

- Command 1.   grep "c.t" words
- Command 2.   grep "c.*t" words

Metacharacters (3) Brackets

Metacharacter	Explanation
[ ]	Any character in the brackets
[-]	Any character in the range
[^]	Not any character in the brackets

- File years

```
2001
2002
…
```

- Command 1. grep "[567]" years
- Command 2. grep "[5-7]" years

Metacharacters (4) Named Classes

Metacharacter	Explanation
[:alpha:]	Any one alphabetic character
[:digit:]	Any one number
[:alnum:]	Any one letter or number
[:upper:]	Any one upper case character

- File mailist

```
name@domain.com
name1@domain.com
name1a@domain.com
```

- Command 1.   grep "name[[:digit:]]*@" mailist
- Command 2.   grep "name[[:alnum:]]*@" mailist

Metacharacters (5) Backslashes

Metacharacter	Explanation
¥n	Newline
¥t	Tab
¥s	Whitespace
¥b	Word border

- File greece

```
¥zeus
        zeus
hera
heracles
```

- Command 1.   grep '¥szeus' greece
- Command 2.   grep 'hera¥b' greece

Extended Regular Expressions

- Extended regular expressions extend the number of metacharacters
- Extended regular expressions need the egrep command or the -E option with grep

Metacharacters (6) Extended

Metacharacters	Explanation
|	Or
+	One or more
?	Zero or one
()	Groups together expressions

- File colors

```
color
colour
gray
grey
```

- Command 1. egrep "colou?r" colors
- Command 2. egrep "gr(a|e)y" colors

The fgrep Command

- fgrep stands for “fixed string grep”

- All metacharacters lose their special meaning with fgrep

- File regex

```
The regular expression .* matches any string of characters
The regular expression ¥s matches whitespace
```

- Command 1. fgrep '.*' regex
- Command 2. fgrep '¥s' regex

The sed Command

- sed stands for "stream editor"
- sed performs basic editing on its input
- Some basic functions are substituting and deleting
- The syntax for substituting is: sed 's/old/new/g' file
- Example: echo 2012 | sed 's/2$/3/g'
  2013

Thank You Very Much!

LPIC 101
Objective 104.5

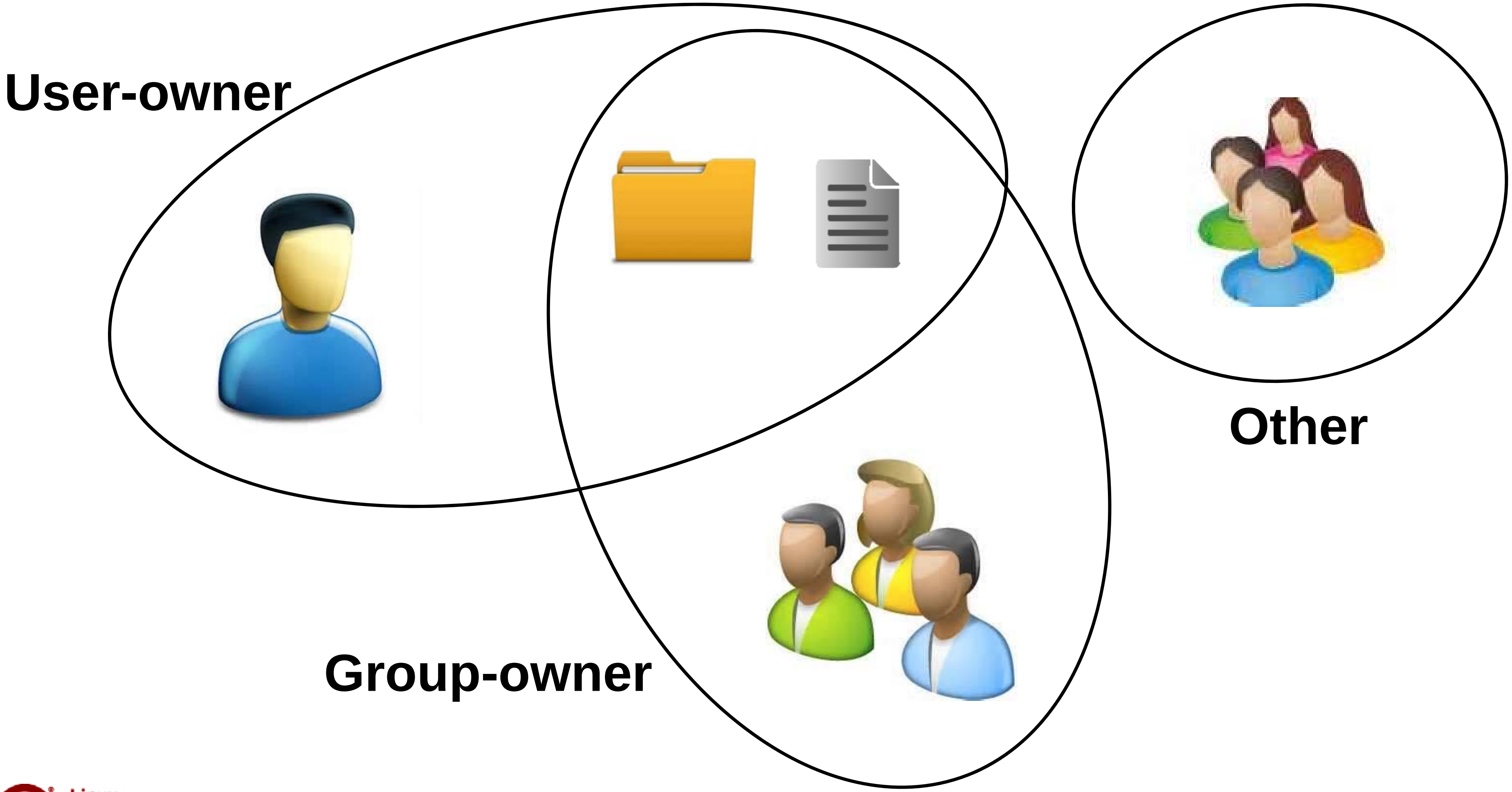
Permissions

User Accounts and Groups

- Linux is a multiuser system
- On a Linux system, there are two kinds of user: the super user (administrator) and regular users
- The super user is called root
- All users have a user account
- User account information is in /etc/passwd
- All users belong to one or more groups
- Group information is in /etc/group
- All users have a user ID and all groups have a group ID
- Root’s UID is 0. Regular users’ IDs start from 500

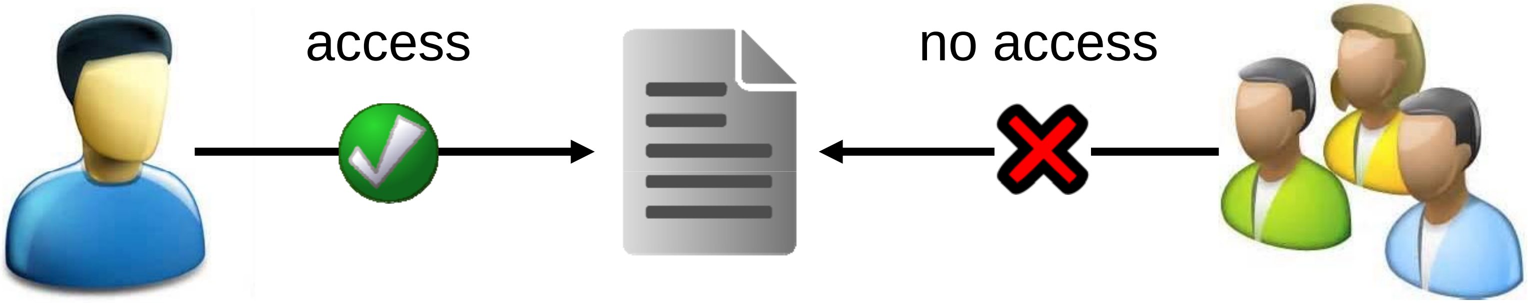
File and Directory Ownership

- Every file and directory has a user-owner and a group-owner
- Every user is either a user-owner, a group-owner or other



What are Permissions?

- Permissions are settings which allow a system administrator to control access to files and directories



The Three Permissions

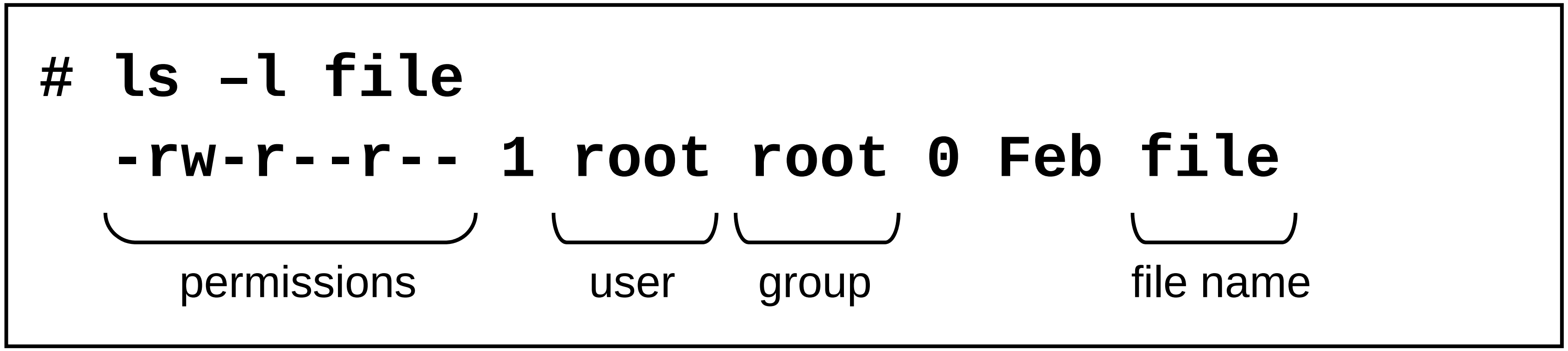
- There are three kinds of permissions a user can have on a file or directory: read, write and execute
- The meanings of read, write and execute are different for files and directories

Permission	File	Directory
read (r)	Open or display a file	List the contents of a directory
write (w)	Edit a file	Make or delete the contents of a directory
execute (x)	Execute a program	Access a directory

Displaying Permissions

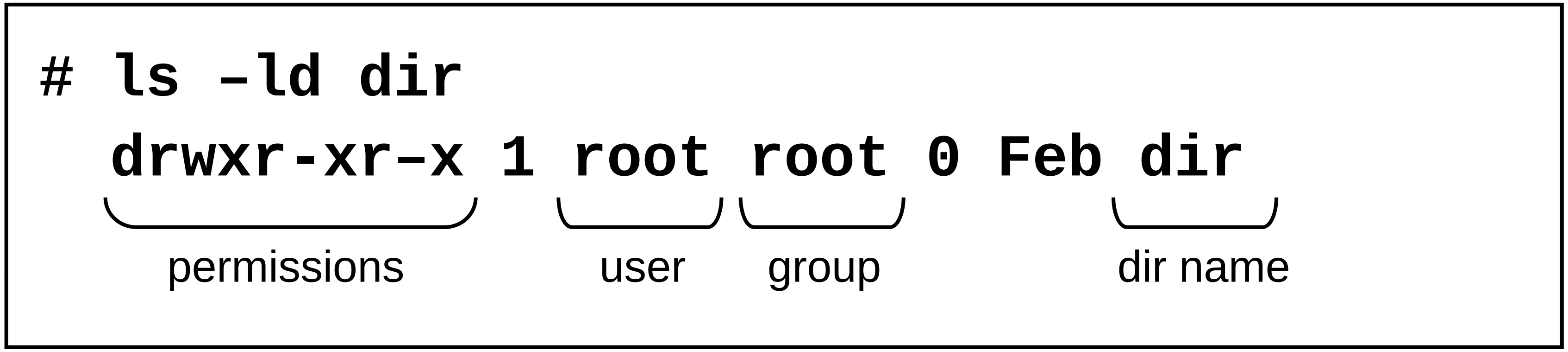
- Permission and ownership information is displayed with the **`ls`** command
- Use the -l option for files and -ld for directories

Files



```
# ls –l file
  -rw-r--r-- 1 root root 0 Feb file
```

Directories



```
# ls –ld dir
  drwxr-xr–x 1 root root 0 Feb dir
```

A Closer Look

- The first character indicates the type of file
- The rest indicates permissions for user, group and other

$$\underbrace{\texttt{-}}_{\text{type}}\underbrace{\texttt{rwx}}_{\text{user}}\underbrace{\texttt{rwx}}_{\text{group}}\underbrace{\texttt{rwx}}_{\text{other}}$$

Type	Meaning
-	File
d	Directory
l	Link

Permissions in Octal (1)

- Permissions can also be written with numbers
- Permissions are written in octal

Decimal: 0 1 2 3 4 5 6 7 8 9 10 …
Octal: 0 1 2 3 4 5 6 7 10 …

- There is one number for each of u, g and o: e.g. 655

Alphabetical Permission	Numerical Permission
r	**4**
w	**2**
x	**1**

Permissions in Octal (2)

- Let’s practice!

Problem 1. rw-	6
Problem 2. rwxr-x	75
Problem 3. rw-r--r--	644
Problem 4. r-xr-xr--	554

Linux Professional Institute

umask

- The umask determines the default permissions for new files and directories
- The first digit is the special permission bit - we’ll get to that later
- The next three are user, group and other
- The umask value is subtracted from the default maximum value for files or directories

Default umask
0022

Files

```
  666 (default maximum value)
– 022 (umask)
  ---------------------------
  644 (default permissions)
```

Directories

```
  777 (default maximum value)
– 022 (umask)
  ---------------------------
  755 (default permissions)
```

# chmod

- The chmod command changes permission settings
- Syntax:

  # chmod [permissions] [file / directory name]
- Example1.

  ```
  # chmod u+x file
  ```
- Example 2.

  ```
  # chmod g+wx file
  ```
- Example 3.

  ```
  # chmod o-rw file
  ```
- Example 4.

  ```
  # chmod g+w,o+x file
  ```
- Example 5.

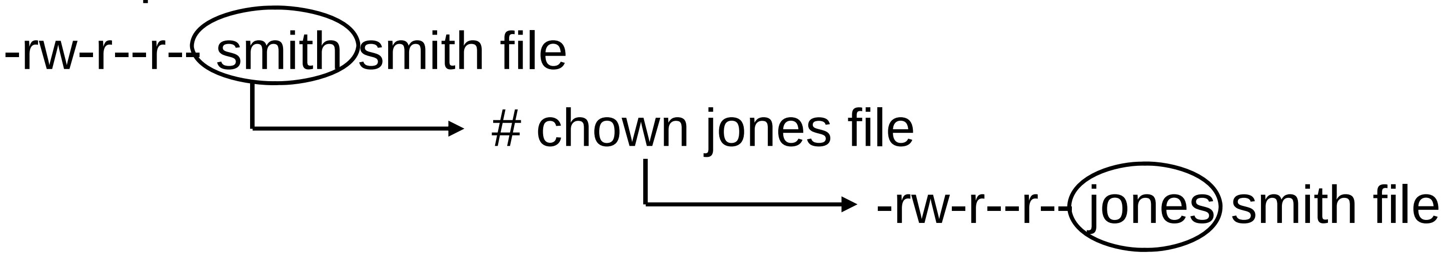
  ```
  # chmod 655 file
  ```

chown

- The chown command changes the file or directory’s owner
- Syntax:

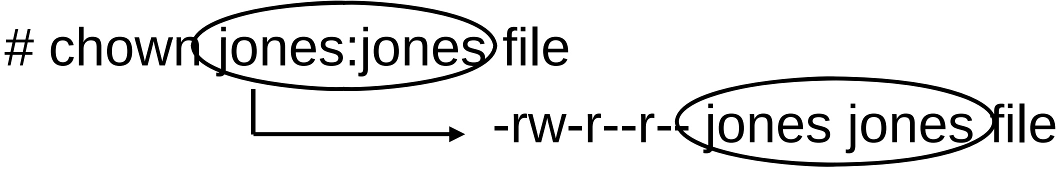
  ```
  # chown [new owner] [file/directory name]
  ```

- Example



```
-rw-r--r-- smith smith file
# chown jones file
-rw-r--r-- jones smith file
```

- You can also change the group owner



```
# chown jones:jones file
-rw-r--r-- jones jones file
```

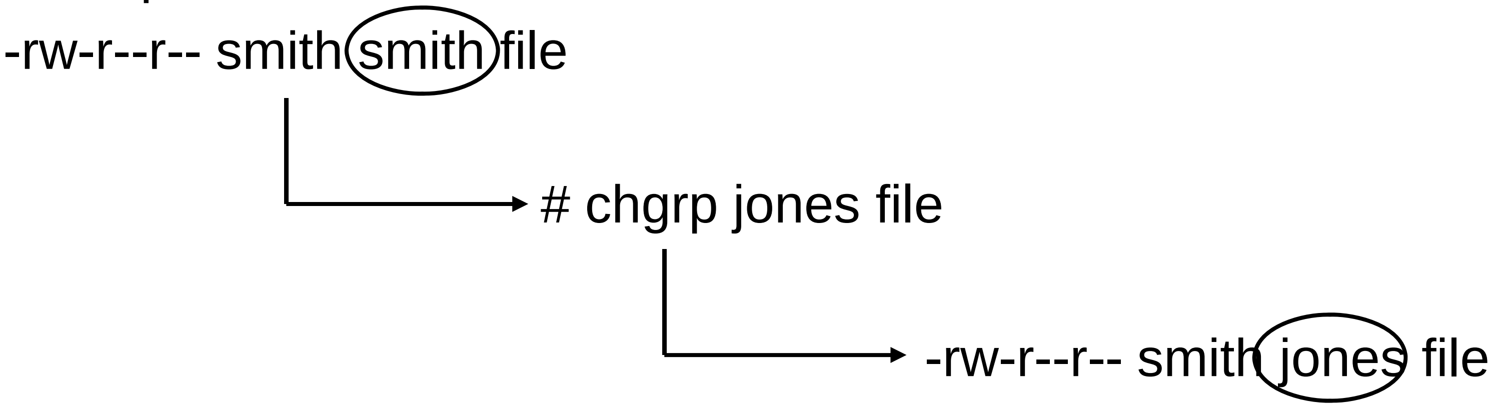
chgrp

- The chgrp command changes the file or directory’s group owner

- Syntax:

  ```
  # chgrp [new group] [file/directory name]
  ```

- Example:



```
-rw-r--r-- smith smith file
```

```
# chgrp jones file
```

```
-rw-r--r-- smith jones file
```

Special Permissions

Overview

- There are three special permissions
- Special permissions have different effects on programs or directories
- Like regular permissions, they can be expressed alphabetically or numerically

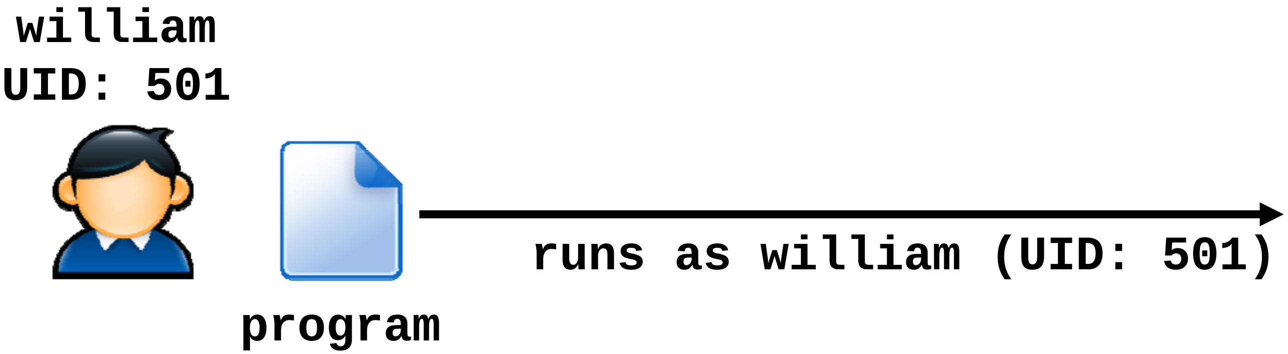
Perm	Set on Program	Set on Directory	Alphabetical	Num
SUID	○	×	$\underbrace{\texttt{--s}}_{\texttt{u}}\texttt{------}$	**4000**
SGID	○	○	$\texttt{---}\underbrace{\texttt{--s}}_{\texttt{g}}\texttt{---}$	**2000**
Sticky Bit	×	○	$\texttt{------}\underbrace{\texttt{--t}}_{\texttt{o}}$	**1000**

SUID (1)

- SUID stands for Set User Identification
- SUID is used so that regular users can run commands owned by the root user
- If the SUID bit is set on a program, the file runs with the UID of the owner of the program, not the UID of the user.

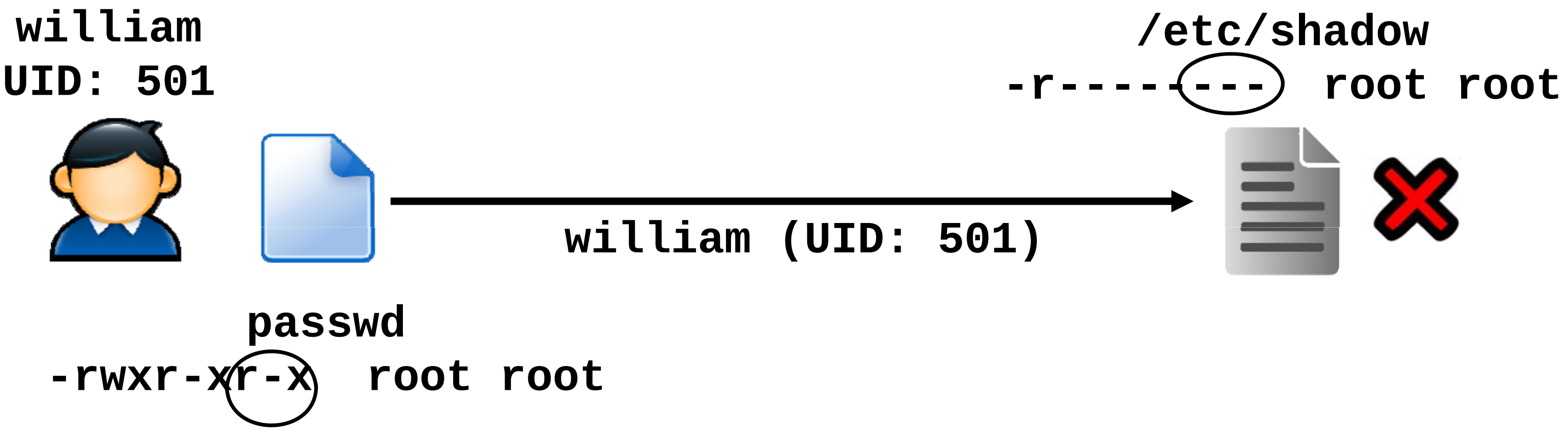
SUID (2)

- Programs run with a User ID (UID)
- Programs usually run with the UID of the user who ran the program
- Programs inherit the file access permissions of the user who runs them
- This is important because programs often have to access file to read or write to them



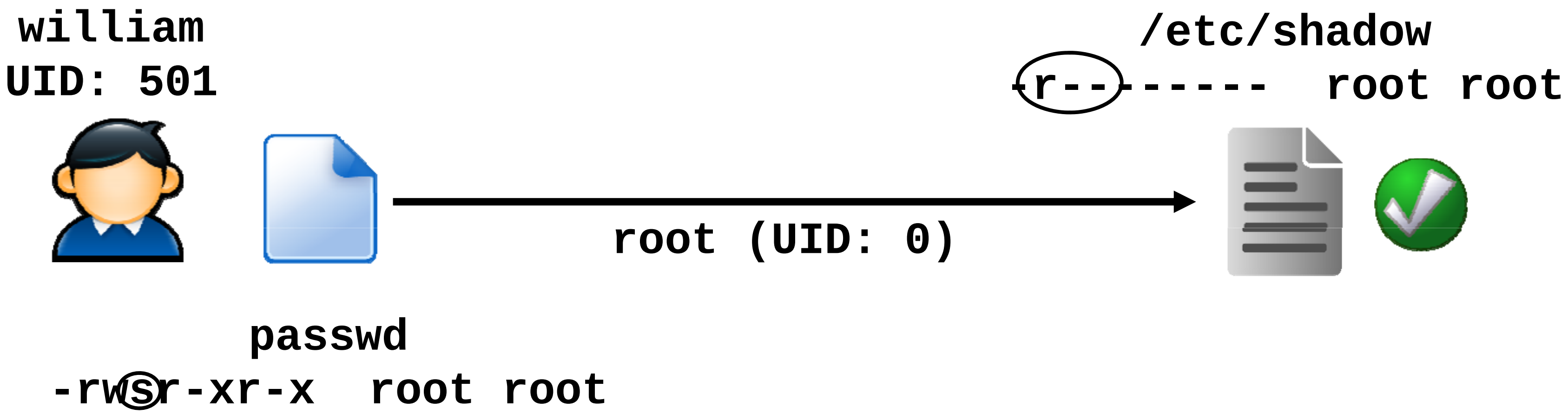
SUID (3)

- The passwd command sets or changes a user’s password
- The passwd command has the SUID bit set, but what would happen if it did not?
- The passwd command has to read the /etc/shadow file, but it has no permission to do this running as william



SUID (4)

- When the passwd command has the SUID set, it runs as root and is able to read the /etc/shadow file
- In this way, regular users are able to use the passwd command, even though they have no permission to read the /etc/shadow file



william
UID: 501

/etc/shadow
-r-------- root root

root (UID: 0)

passwd
-rwsr-xr-x root root

SGID (1)

- Set Group Identification

- When SGID is set on a program, it has the same effect as the SUID, only for group ownership rather than user ownership

- When set on a directory, all files made in the directory are owned by the directory’s group-owner rather than the file maker’s group

- The SGID is often used on shared directories

SGID (2)

- When the SGID is not set, a file's group-ownership is the maker's group

```
[william@station22 ~]$ touch /staff_docs/sep_report
```

```
drwxrwxrwx 1 root users staff_docs
```

```
-rw-r--r-- william william sep_report
```

SGID (3)

- When the SGID is set, a file's group-ownership is the directory's group

```
[william@station22 ~]$ touch /staff_docs/oct_report

drwxrwsr-x 1 root users staff_docs
```

```
-rw-r--r-- william william sep_report
-rw-r--r-- william users oct_report
```

Sticky Bit (1)

- The sticky bit is used on shared directories to prevent users other than the file’s owner from accidentally or maliciously deleting another user’s file

- When the Sticky Bit is set, only root and the file’s owner can delete the file

Sticky Bit (2)

- When the sticky bit is not set, any user with write permissions to the directory can delete a file in the directory

```
drwxrwsr-x 1 root users staff_docs
```

```
-rw-r--r-- william users sep_report
-rw-r--r-- william users oct_report
```

```
[timothy@station666 staff_docs]$ rm -f oct_report
```

Sticky Bit (3)

- When the sticky bit is set, only root and the file's owner can delete the file

```
drwxrwsr-t 1 root users staff_docs
```

```
-rw-r--r-- william users sep_report
```

```
[timothy@station666 staff_docs]$ rm -f sep_report

          rm : Operation not permitted
```

Thank You Very Much!

LPIC 102
Objectives 109.1 & 109.2

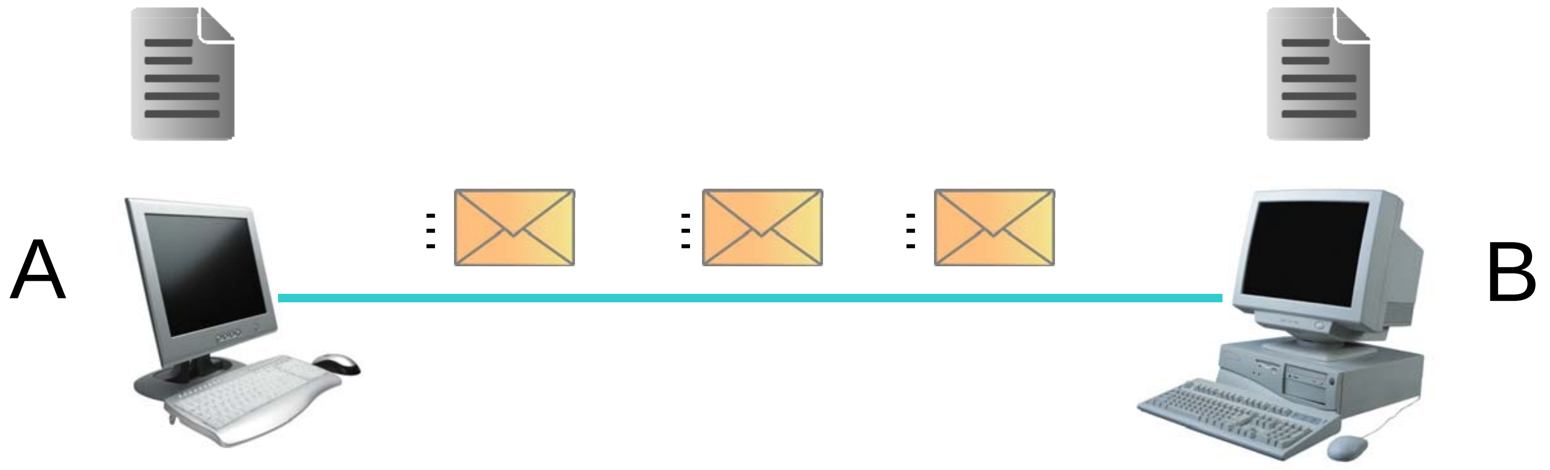
Networking

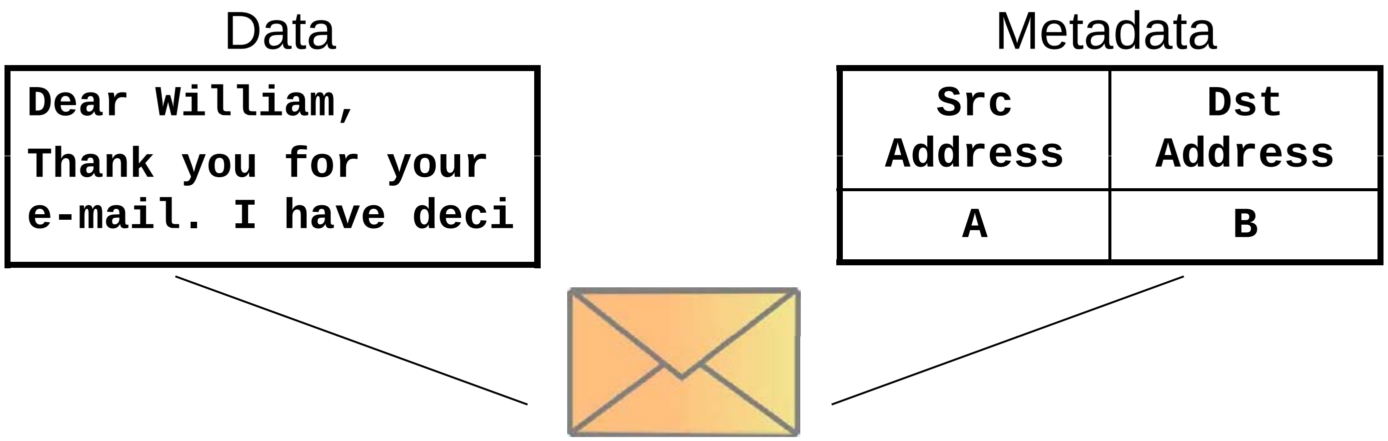
Computer Networks

- Computer networks allow us to send data between computers
- There are many factors involved in a computer network:
  - Servers
  - Protocols
  - Addresses
  - Host names, domain names

Packets

- Data is divided into packets and sent across the network





Data

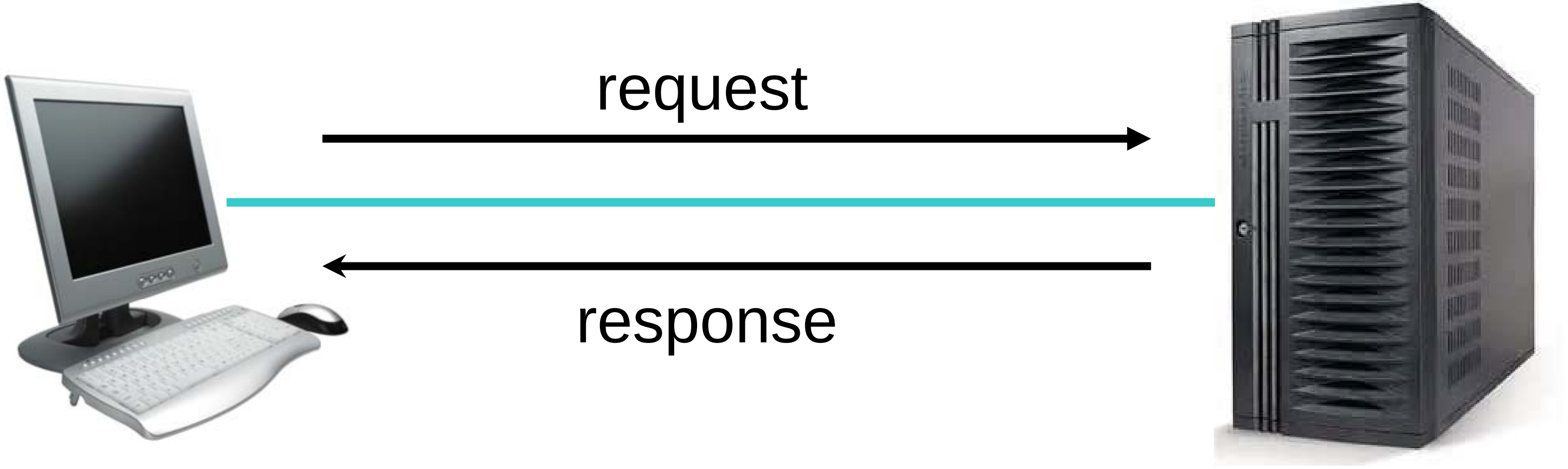
```
Dear William,
Thank you for your
e-mail. I have deci
```

Metadata

Src Address	Dst Address
A	B

Client ⇔ Server

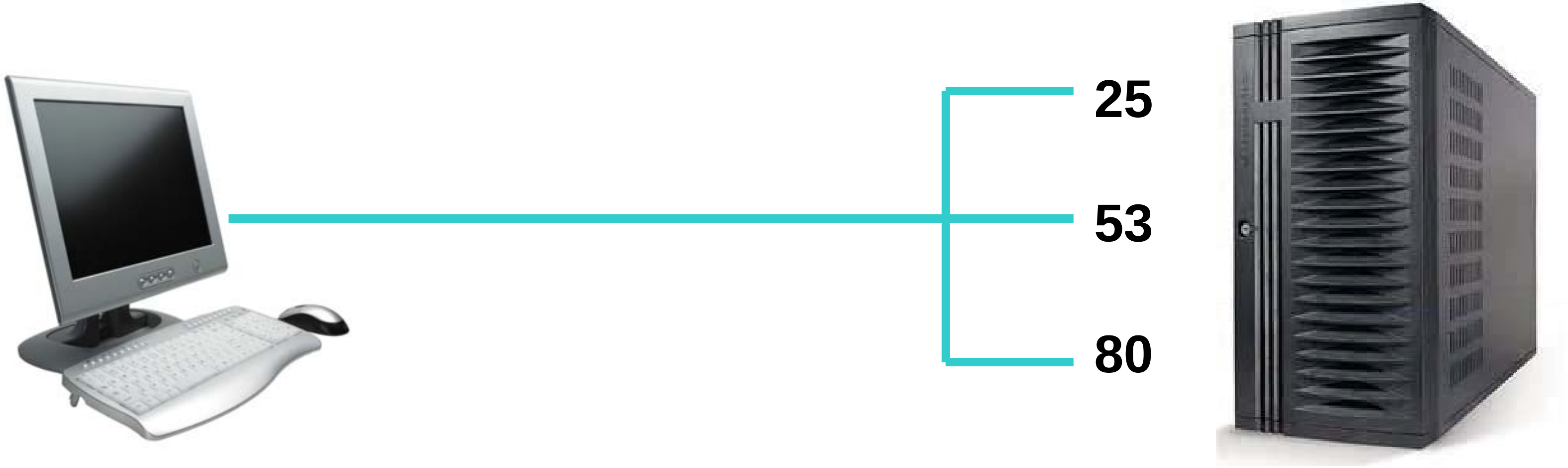
- Servers provide services to clients
- Clients connect to servers and make requests



Server Type	Service
Web	Provide web pages, e-commerce, etc.
E-mail	Store and deliver e-mail
DNS	Resolve domain names to IP addresses

Ports

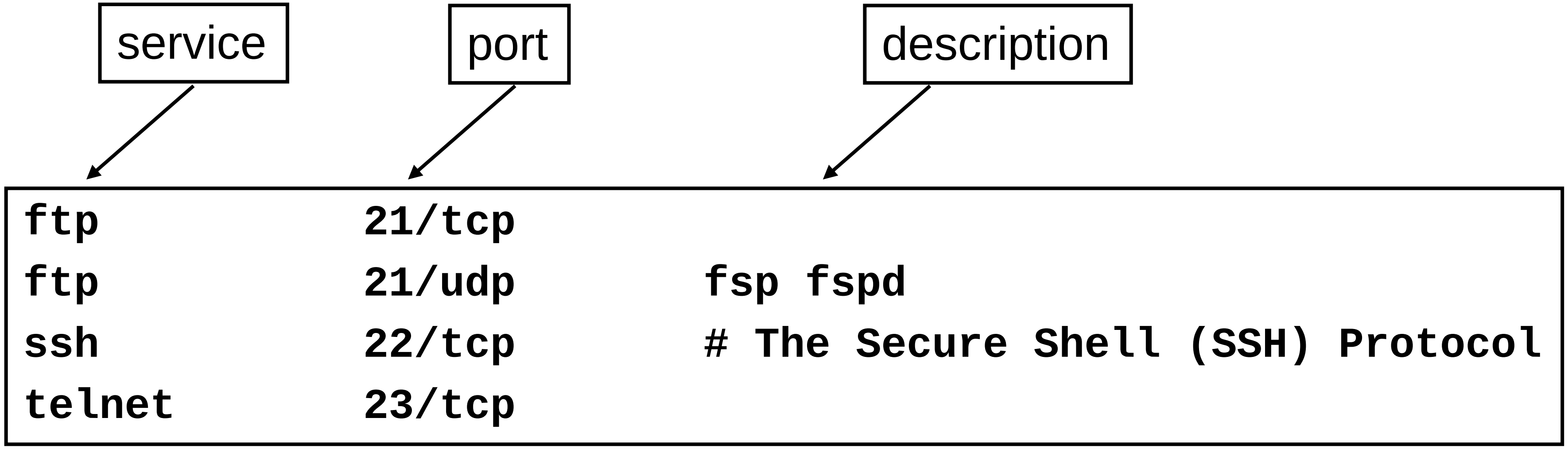
- Ports are numbers which differentiate services



Port	Service
25	SMTP (E-mail)
53	DNS (Name resolution)
80	HTTP (Web)

/etc/services

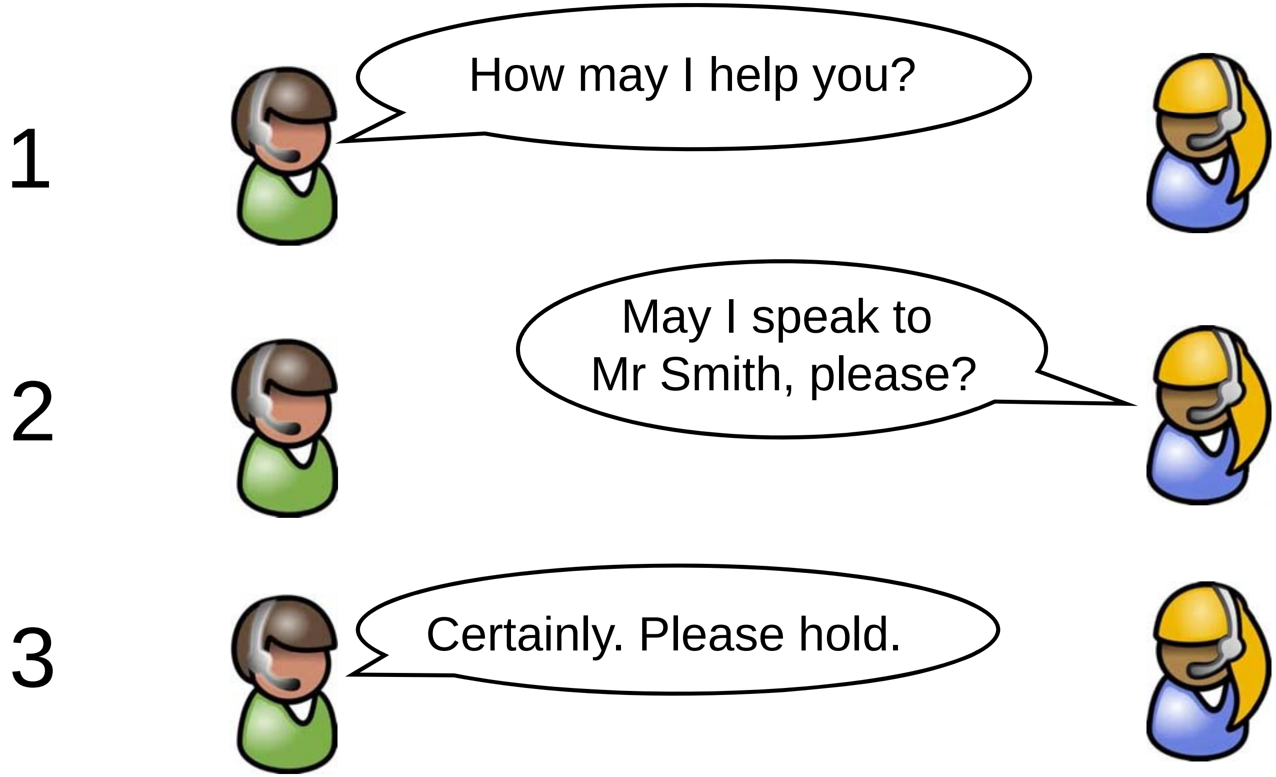
- The /etc/services file contains a list of services and port numbers



```
ftp             21/tcp
ftp             21/udp          fsp fspd
ssh             22/tcp          # The Secure Shell (SSH) Protocol
telnet          23/tcp
```

Protocols (1)

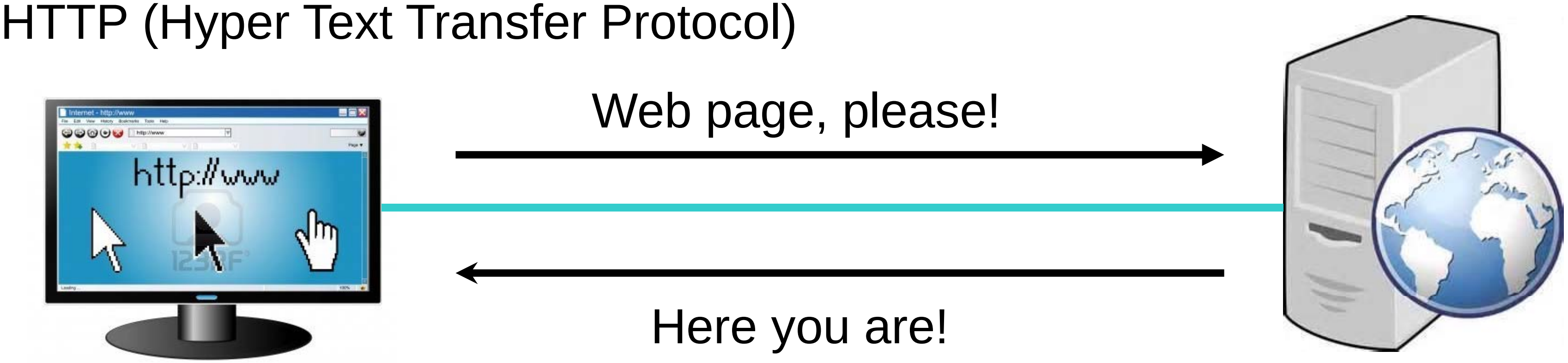
- Protocols are rules of communication
- Standardized protocols allow communication between different makes of computers



Protocols (2)

- Communication between computers is also governed by protocols

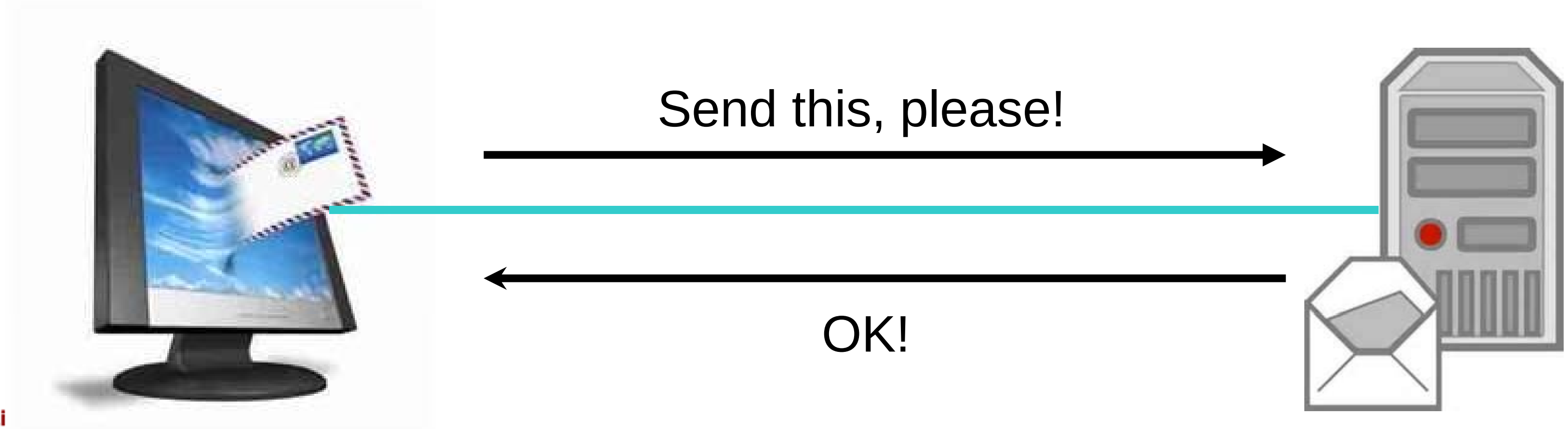
HTTP (Hyper Text Transfer Protocol)



Web page, please!

Here you are!

SMTP (Simple Mail Transfer Protocol)



Send this, please!

OK!

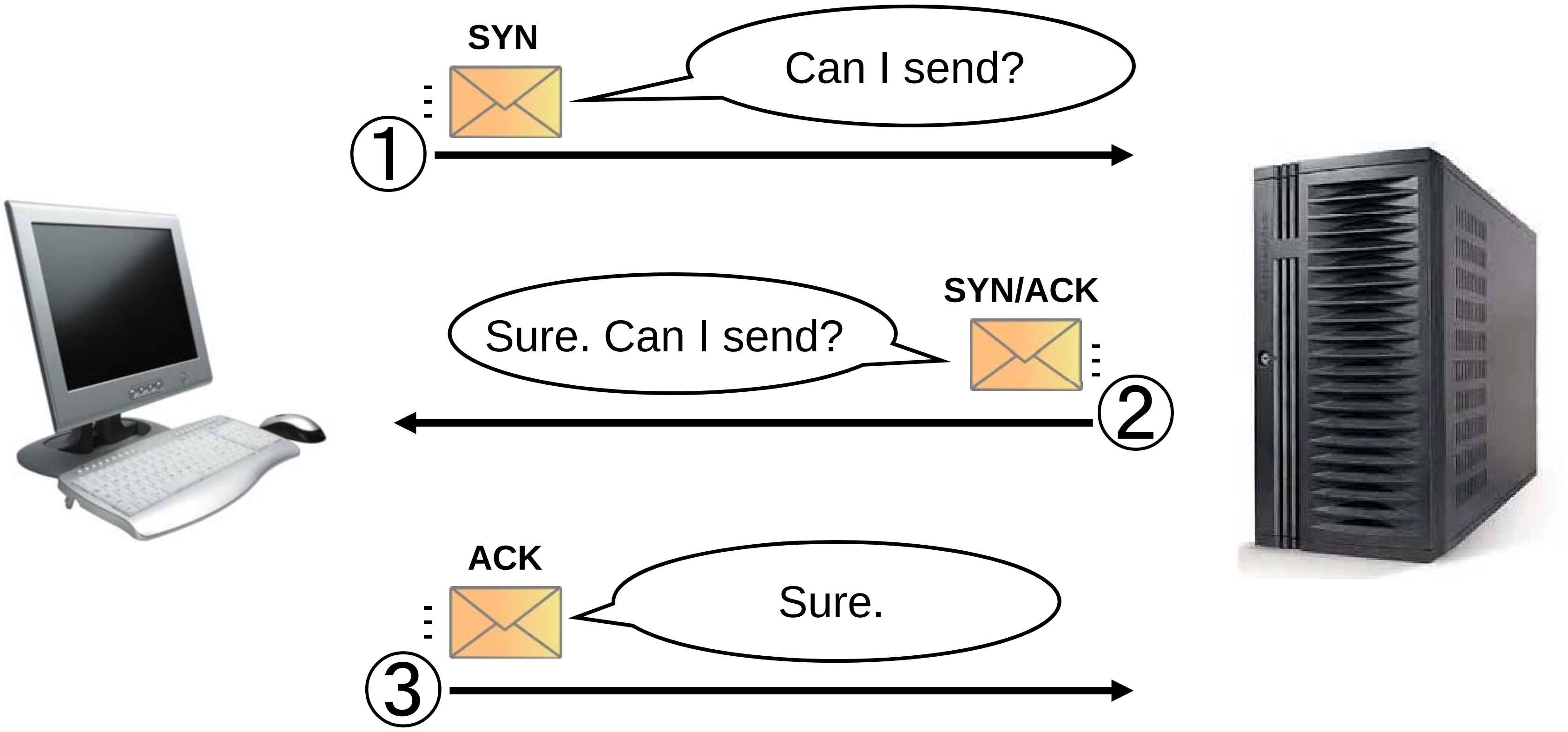
TCP (1)

- TCP stands for Transmission Control Protocol

- TCP provides mechanisms for reliable data transmissions
    - Three-Way Handshake
    - Flow Control

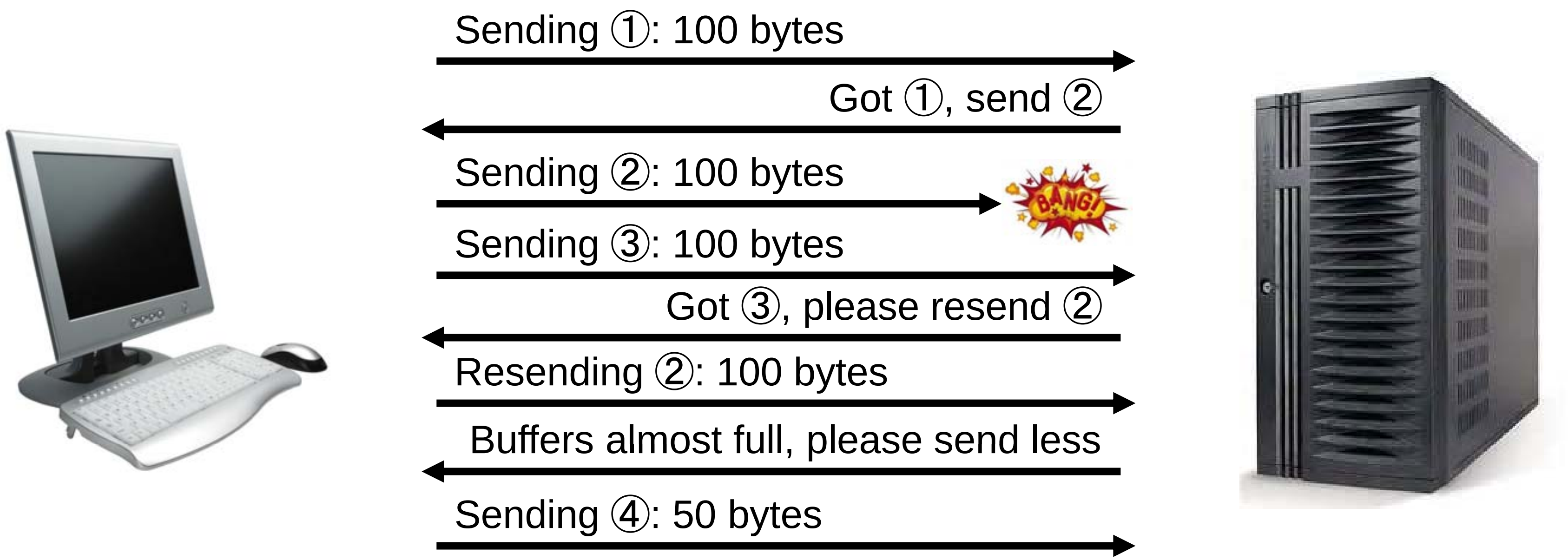
TCP (2) Three-Way Handshake

- The three-way handshake establishes a reliable line of communication



TCP (3) Flow Control

- Flow control includes:
  - Sequencing (sending packets in order)
  - Resending (resending lost packets)
  - Sliding Window (controlling the size of packets)



Sending ①: 100 bytes

Got ①, send ②

Sending ②: 100 bytes

Sending ③: 100 bytes

Got ③, please resend ②

Resending ②: 100 bytes

Buffers almost full, please send less

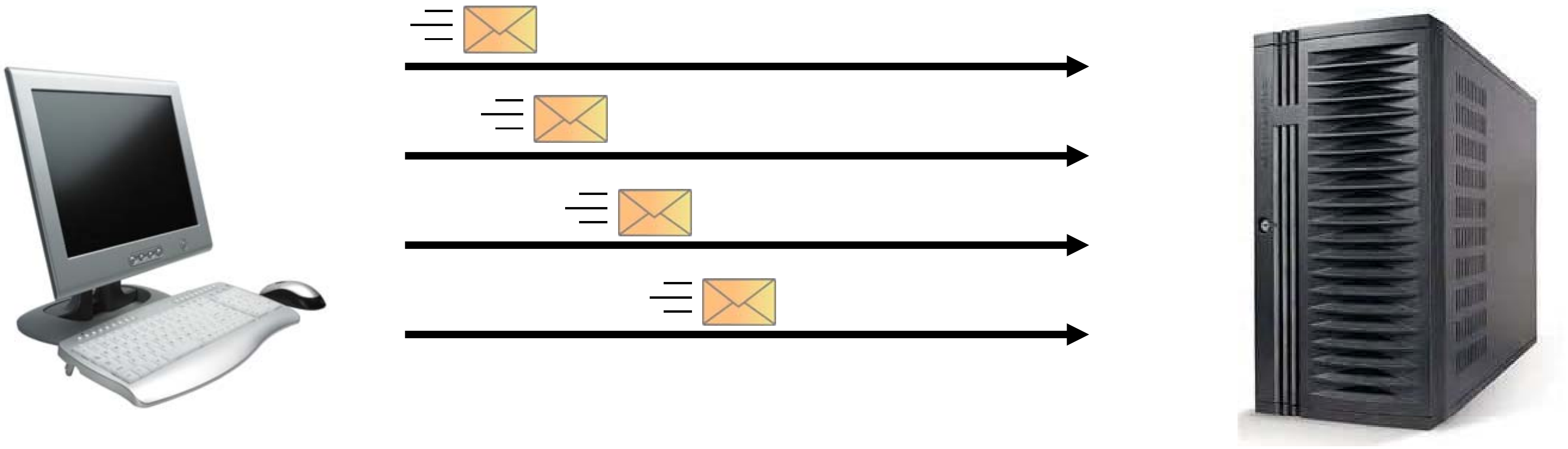
Sending ④: 50 bytes

UDP (1)

- UDP stands for User Datagram Protocol
- UDP is:
  - Unreliable, but fast
  - Free of TCP’s overhead
  - Used for streaming, graphics
  - Also used when an application has its own reliability controls

UDP (2)

- UDP simply sends the packets to the destination
- It does not guarantee their arrival!



IP

- IP stands for Internet Protocol
- IP provides a computer address scheme, making it possible to send data from one computer to another

IP Addresses

- IP addresses are 32 bits long (4 × 8)
- They are written in dotted quad notation: 4 numbers separated by dots
- IP addresses are usually written in decimal, although it is important to be able to understand them in binary, too!

Binary

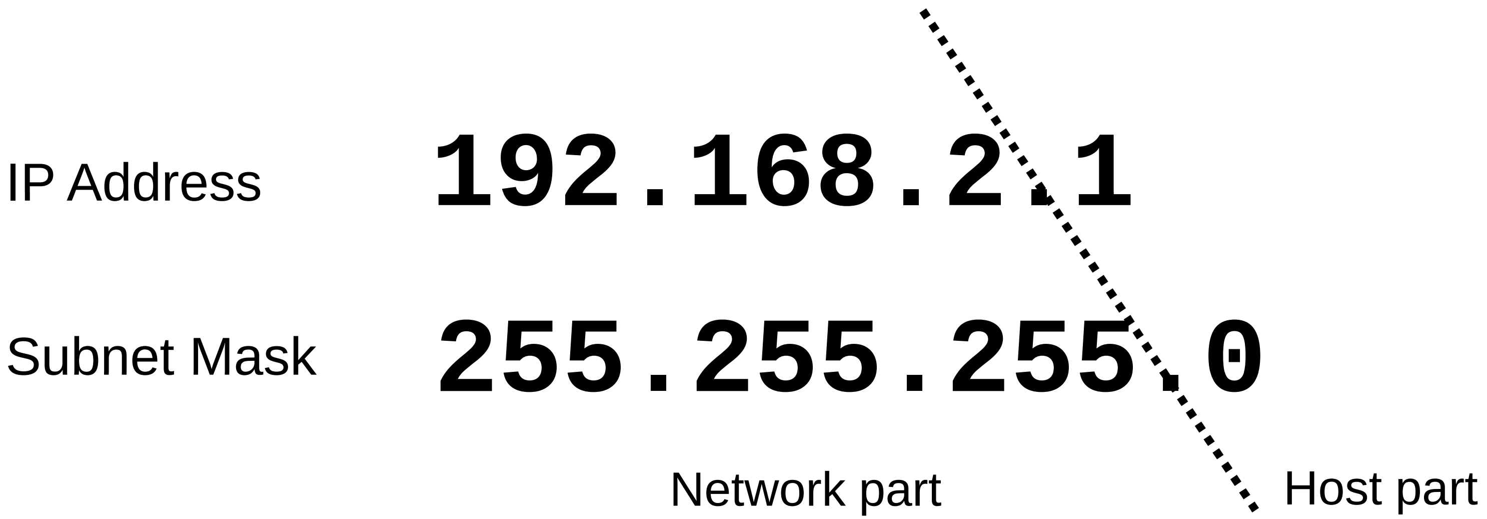
```
11000000.10101000.00000010.00000001
```

Decimal

```
192.168.2.1
```

Subnet Masks

- IP addresses are divided into a network part and a host part
- Subnet masks tell us where one ends and the other begins



IP Address **192.168.2.1**

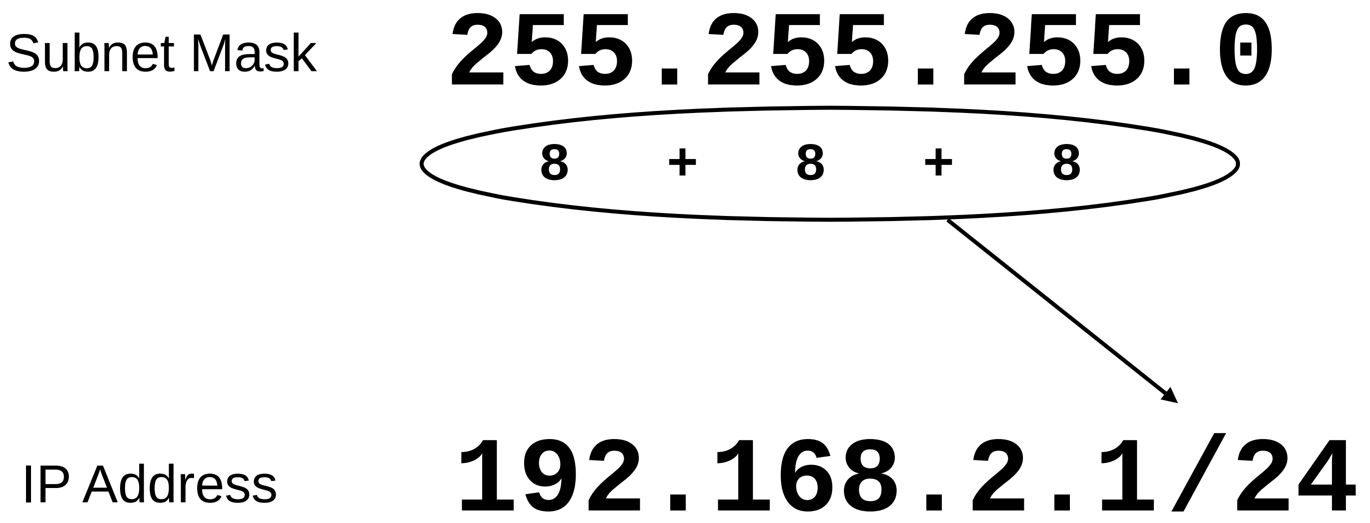
Subnet Mask **255.255.255.0**

Network part

Host part

CIDR

- CIDR is another way to write subnet masks



Subnet Mask

255.255.255.0

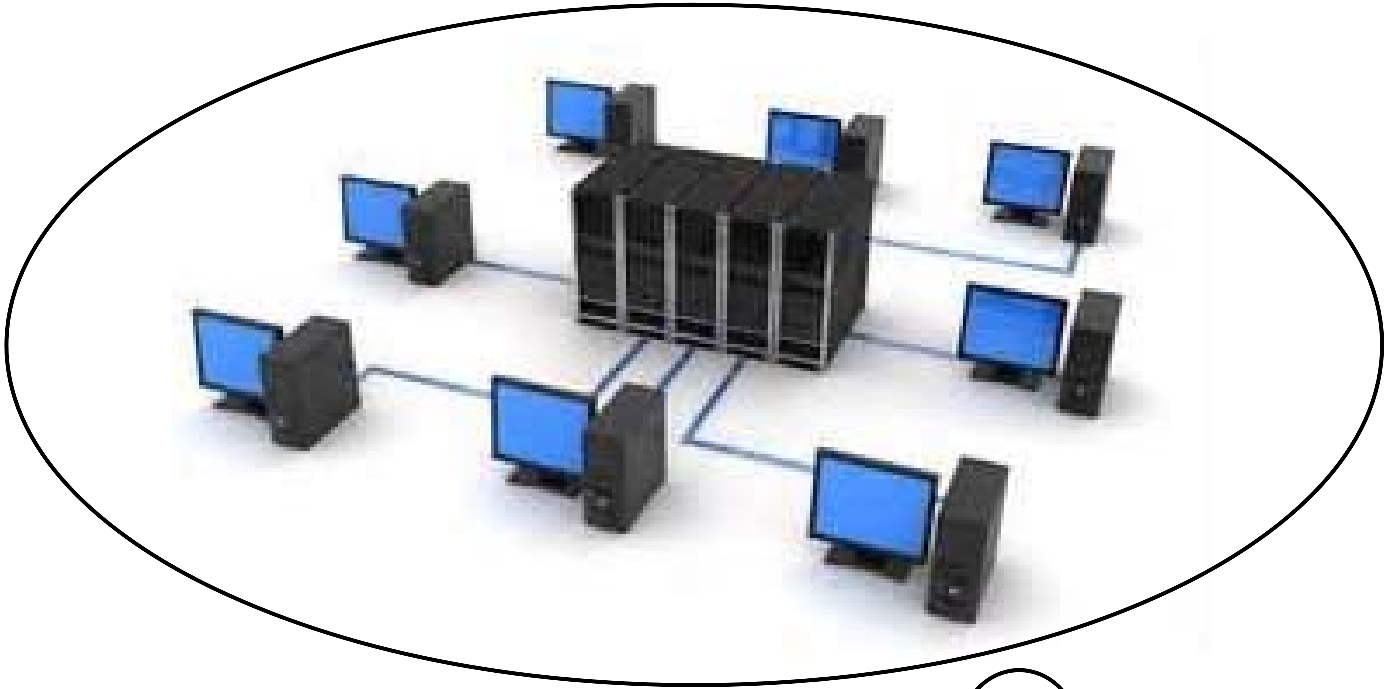
8 + 8 + 8

IP Address

192.168.2.1/24

Network Addresses

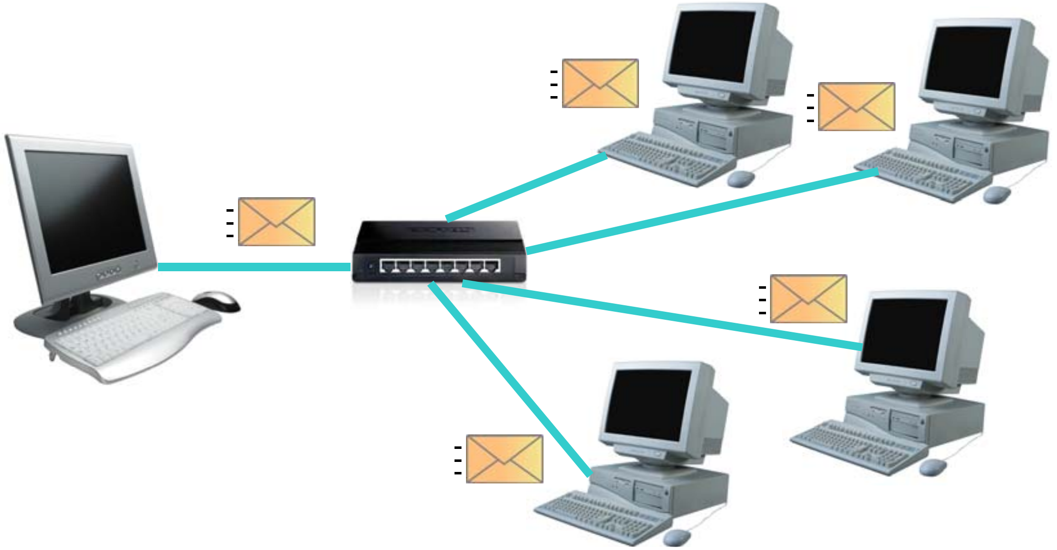
- Network addresses represent a whole network
- They have a zero in the host part of the IP address



192.168.2.0

Broadcast Addresses

- A broadcast is a transmission sent from one to many
- It is used to send packets to all computers on the LAN at once
- The address used has a 255 in the host part



192.168.2.255

IP Address Classes

- IP addresses are grouped into classes
- Classes D and E are for special use - you can ignore them!

Class	Range
A	1.0.0.0 ~ 127.255.255.255
B	128.0.0.0 ~ 191.255.255.255
C	192.0.0.0 ~ 223.255.255.255
D	224.0.0.0 ~ 251.255.255.255
E	252.0.0.0 ~ 255.255.255.255

Public vs. Private IP Addresses

Public

- Used on computers on the Internet
- Must be registered
- Are unique

Private

- Used in homes and businesses
- Can be used freely (it is OK to double them)
- Are not unique

Private IP Address Range

- Let’s memorize the private IP addresses!

Class	Private IP Address Range
A	10.0.0.0 ~ 10.255.255.255
B	172.16.0.0 ~ 172.31.255.255
C	192.168.0.0 ~ 192.168.255.255

IP Address Exhaustion

- IPv4 gives us 4,294,967,296 numbers – not enough!
- Recent years have seen an increase in Internet users and mobile devices
- IPv4 address exhaustion occurred in 2011

Solution: IPv6

- Used since 2006
- IPv6 gives us 340,000,000,000,000,000,000,000,000,000,000,000,000 addresses
- IPv6 addresses are 128 bits long
- IPv6 addresses are written in hexadecimal

Network Interface

- A network interface is used to connect to a network
- A computer needs a NIC (Network Interface Card) to connect to a network

The ifconfig Command

- The ifconfig command displays and sets network interface settings

Display

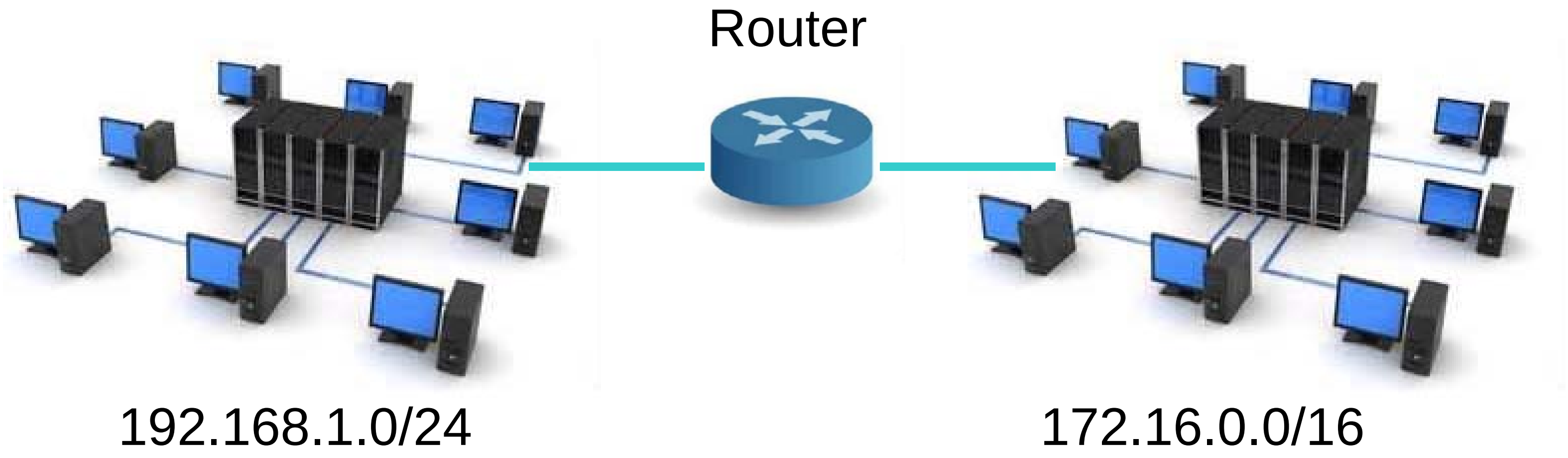
```
ifconfig eth0
```

Set

```
ifconfig eth0 192.168.2.1 netmask 255.255.255.0
```

Routing

- Routing is choosing the best path through the network for a packet to reach its destination
- Routing is handled by machines called . . . Routers
- Networks with different network addresses need a router



Routing Tables

- Routers and computers have routing tables, which dictate the route packets travel on the network
- The route command displays Linux’s routing table

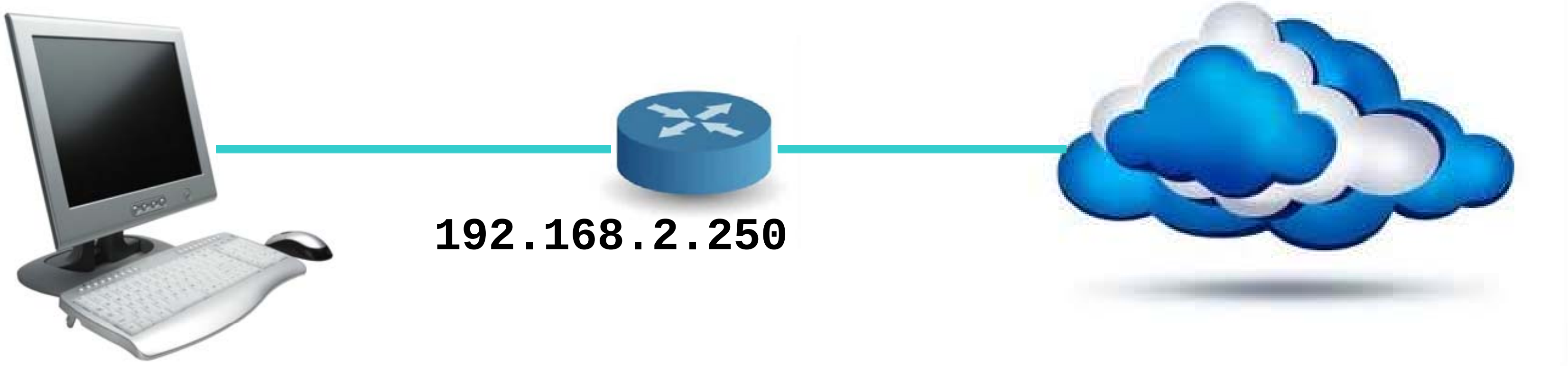
```
# route -n
```

```
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref   Use Iface
192.168.2.0     0.0.0.0         255.255.255.0   U     1      0     0    eth0
0.0.0.0         192.168.2.250   0.0.0.0         UG    0      0     0    eth0
```

Default Gateway

- A default gateway is a router which connects a computer to the Internet
- The route command sets a default gateway

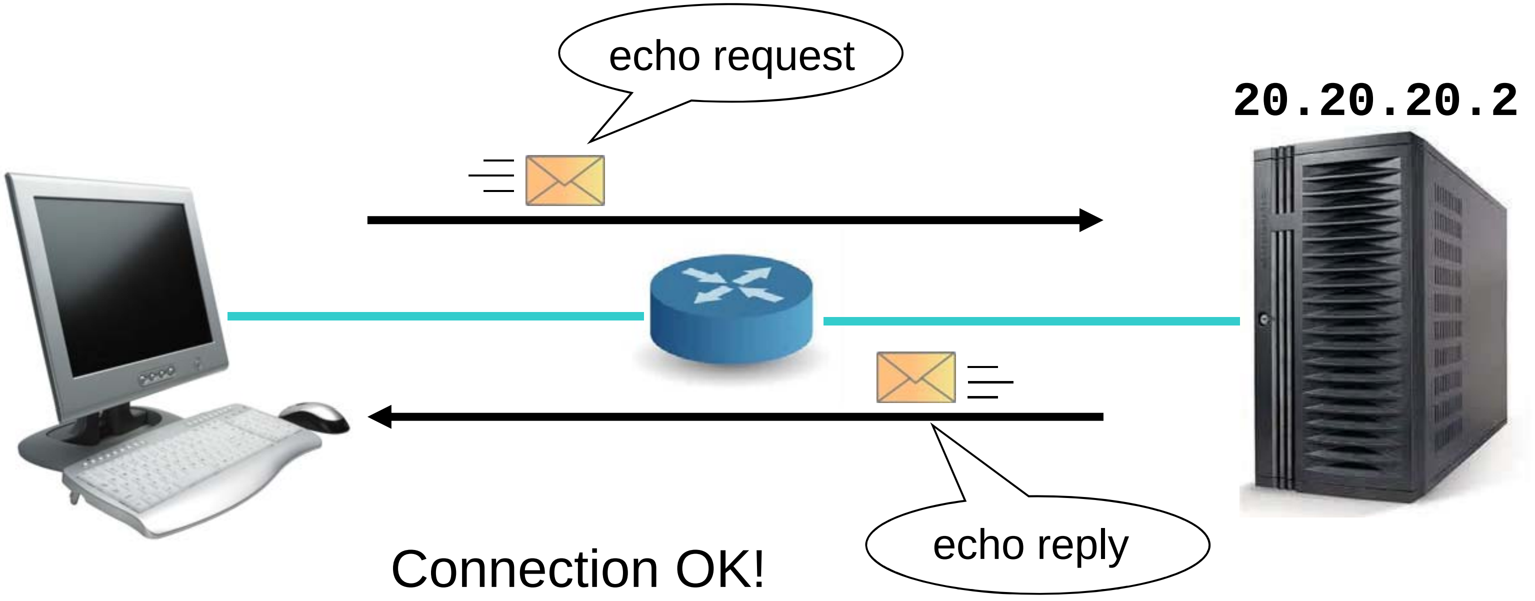
```
route add default gw 192.168.2.250
```



ICMP

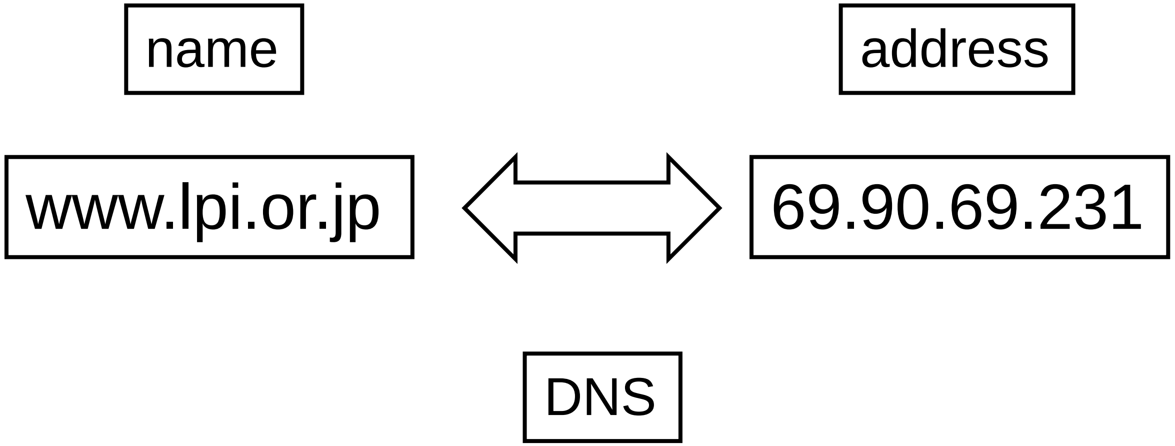
- The ICMP protocol is used for two things
  - Sending echo requests and responses
  - Sending error messages when connectivity fails
- The ping command uses the ICMP protocol

```
ping 20.20.20.2
```



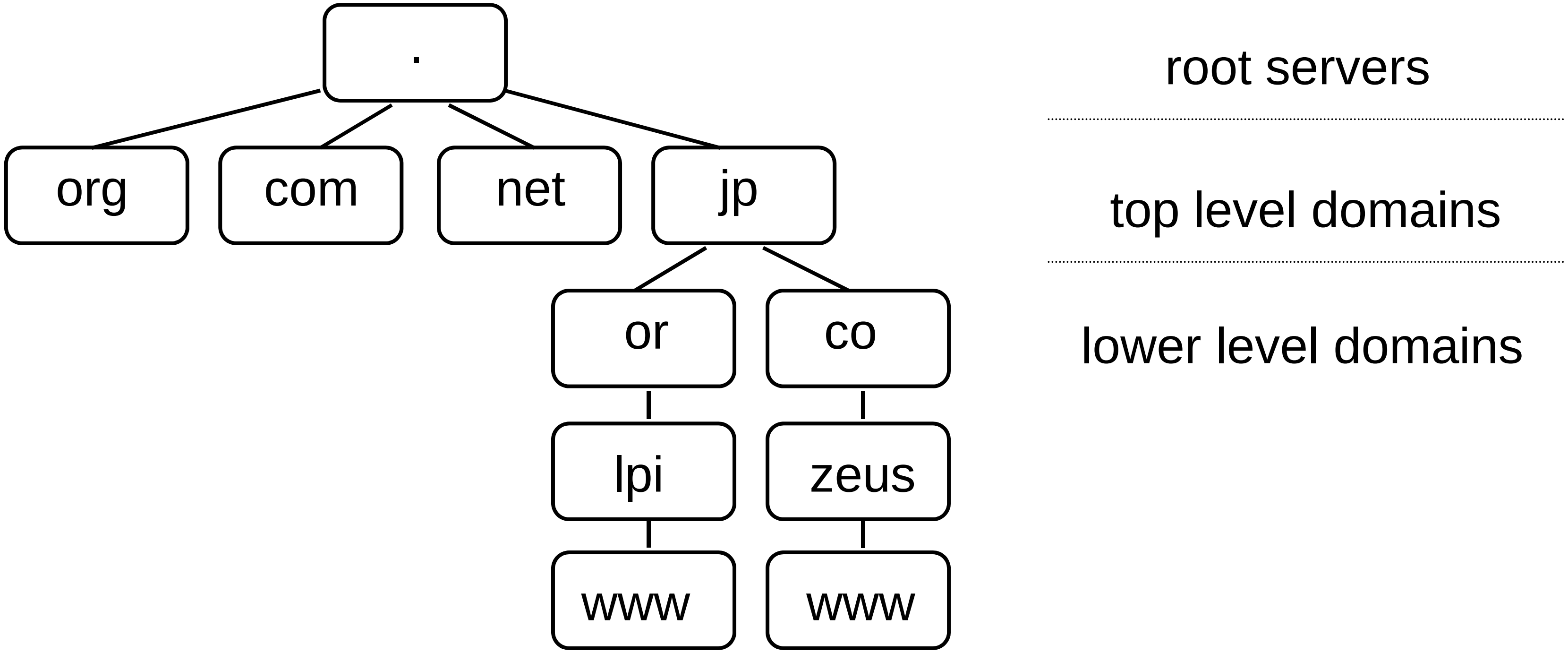
DNS (1)

- DNS stands for Domain Name System
- Computers like numbers; People like names
- People give names to computers: www.lpi.org
- Computers communicate with IP addresses
- DNS bridges the gap by enabling lookups between names and addresses



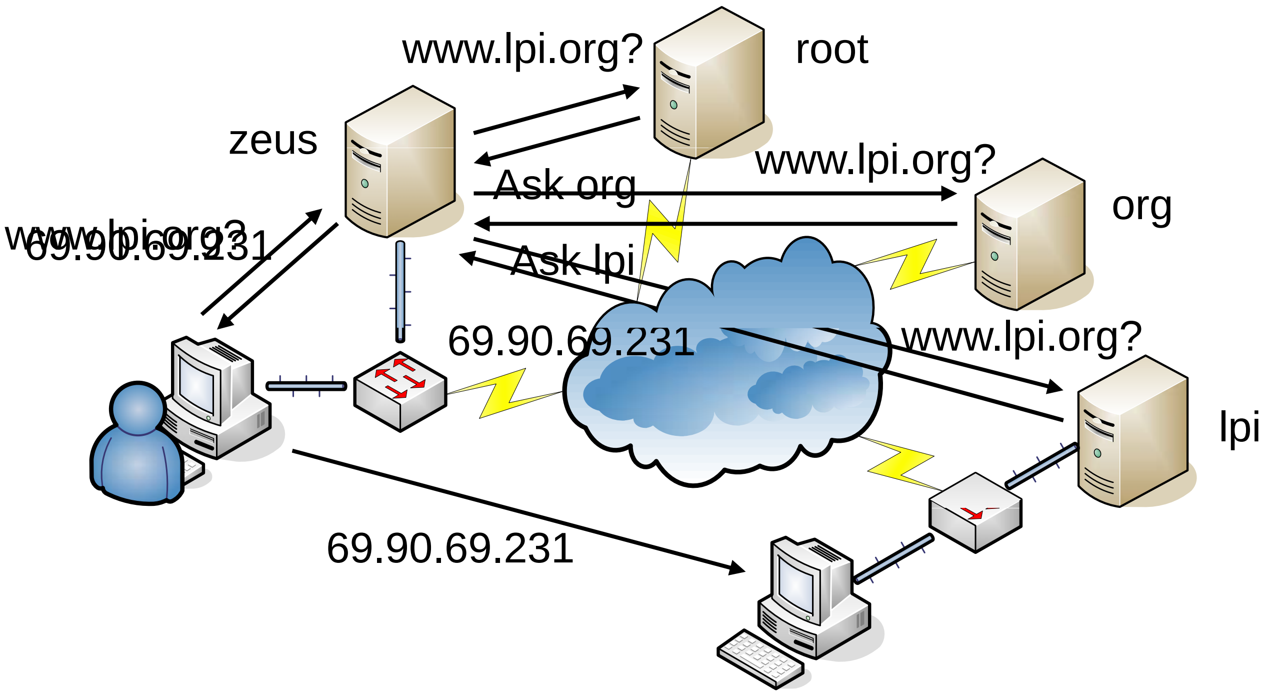
DNS (2) Domain Names

- A hostname is a computer’s name
- A domain name is (basically) the name of a company’s network(s).
- A fully qualified domain name (FQDN) is the whole name
- The DNS is a hierarchy



DNS (3) Server

- DNS servers are the telephone books of the Internet
- A client makes a request for an IP address lookup
- If the server does not know the address, it does a recursive lookup (i.e. goes and asks other servers)



DNS (4) Client Settings

- A client needs access to a DNS server if it wants to use names
- Client settings are in the /etc/resolv.conf file

```
# cat /etc/resolv.conf
```

```
nameserver 192.168.2.250
```

DNS (5) Client Commands

- DNS client commands are used to perform manual lookups of IP addresses

- You have to know three for the test

- host

Example

```
# host www.lpi.org
```

- nslookup

Example

```
# nslookup www.lpi.org
```

- dig

Example

```
# dig www.lpi.org
```

Thank You Very Much!