

# Chapter 1. Basic Concepts

## Table of Contents

Objectives .....	2
1.1 Introduction .....	2
1.2 Hypertext .....	2
1.2.1 Anchors and Links .....	3
1.2.2 Jumps .....	3
1.2.3 Knowledge Additivity.....	4
1.2.4 Chain of Links .....	4
1.2.5 Loops and Mesh.....	4
1.2.6 Hypermedia.....	4
1.2.7 Authoring Hypertext .....	4
1.2.8 Getting lost in 'hyperspace' .....	5
1.3 The Ultimate Hypermedia System: The World Wide Web .....	5
1.3.1 Basic Ideas of the Web .....	5
1.3.2 Fields of Application .....	6
1.3.3 The Web as a Digital Library.....	6
1.4 Summary of Web Terminologies.....	7
1.4.1 Network Protocols .....	7
1.4.2 Web Application (Webapp).....	7
1.4.3 Uniform Resource Locator (URL).....	8
1.4.4 HyperText Markup Language (HTML).....	8
1.4.5 HyperText Transfer Protocol (HTTP).....	8
1.5 The Client-server Computing Model.....	9
1.5.1 A Definition and some History .....	9
1.5.2 Functionality .....	10
1.5.3 Information and Processing on the Web.....	11
1.5.4 MIME Types.....	11
1.5.5 Web Servers.....	12
1.5.6 Distributed Processing .....	13
1.6 Review Questions .....	14
1.7 Answers to Exercises .....	15
1.7.1 Exercise 2.....	15
1.7.2 Exercise 3.....	16
1.7.3 Exercise 4.....	16
1.7.4 Exercise 5.....	16
1.7.5 Exercise 6.....	17
1.7.6 Review Question 1.....	17
1.7.7 Review Question 2.....	17
1.7.8 Review Question 3.....	17
1.7.9 Review Question 4.....	17
1.7.10 Review Question 5.....	17
1.7.11 Review Question 6.....	18

# Objectives

At the end of this chapter you will be able to:

- Explain the basics of Hypertext;
- Explain web technology terminologies;
- Understand the client-server computing model.

## 1.1 Introduction

In 2009 the Internet celebrated its 40th anniversary, and the World Wide Web had been in existence for over 15 years. The concepts of computer networks and hypertext on which these technologies rely are only a little older. And yet the speed of development of these technologies, the speed of uptake by companies, and the speed of acceptance by consumers is unlike anything mankind has witnessed. Although both the Internet and the Web are firmly rooted in academic, altruistic endeavour, there is no doubt that the commercial interests are currently driving much of the technological development. This module aims to prepare you for contributing to this endeavour by helping you to understand the basic ideas and technologies behind the Internet, and giving you the opportunity to design and write Web pages using HTML5 and JavaScript.

The module starts here with, inevitably, the more theoretical aspects of the Internet and the Web. We begin by explaining hypertext before moving on to the most elaborate hypermedia system, the Web, and the ideas of client-server computing that allow us to use it.

## 1.2 Hypertext

Take a dictionary and observe how its content is linked together. How do you search for the meaning of a word? How can you find another word synonymous with that word? The dictionary is a paper example of a hypertext system. So are encyclopedias, product catalogues, user help books, technical documentation and many other kinds of books. Information is obtained by searching through some kind of index - the dictionary is arranged in alphabetical order, and each word is its own index. Readers are then pointed to the page of any other related information. They can read the information they are interested in without having to read the document sequentially from beginning to end.

Hypertext systems allow for non-sequential or non-linear reading. This is the underlying idea of a hypertext system. The result is a multidimensional document that can be read by following different paths through it. In this section we will look into the application of hypertext in computer systems, mainly the World Wide Web hypertext system.

The main use of hypertext is in information retrieval applications. The ease of linking different pieces (fragments) of information is the important aspect of hypertext information retrieval. The information can be of various media: it may be fragments of textual documents, structured data from databases, or list of terms and their definitions. Any of these, or a mixture thereof, can make up the contents of a hypertext document.

Therefore, in a hypertext system it is possible to:

- link with a term that represents aspects of the content of a document;
- connect two related documents;
- relate a term to a fragment containing its definition and use; and
- link two related terms.

Such a hypertext system can store a large collection of textual and multimedia documents. Such a hypertext system gives the end-user access to a large repository of knowledge for reading, browsing and retrieving. This is a "database" of sorts, and is the reason why such a hypertext system is called a digital library. The Web started as an extensively large digital library. As it has grown in popularity, it has offered the possibility of interactive applications and commerce on the Internet, making it much more than a digital library.

### To do

Read about networked hypertext and hypermedia in your textbooks.

We now explain some basic concepts on the use of hypertext.

### 1.2.1 Anchors and Links

A hypertext document contains links referring to other parts of the document, or even to whole other documents. A hypertext document does not have to be read serially; the fragments of information can be accessed directly via the links contained in the document.

The links embedded in a document are known as hyperlinks. When selected, these hyperlinks allow for the portion of the document linked to by the hyperlink to be displayed. This allows the reader to jump to another part of the same page, another page in the same document, or another document. By following a series of hyperlinks, the reader can follow their own path through the document.

A computerised hypertext system implements this idea by including anchors and links in documents, which are usually represented by files. An anchor is a fragment of information which links to another document or portion thereof. It is the visual representation of a link. A link is the actual reference (or "pointer") to the other document. For example, in Figure 1 the fragment of Document A containing 'You can find this in Section 5 of B' is an anchor from which there is a link to the relevant section in Document B.

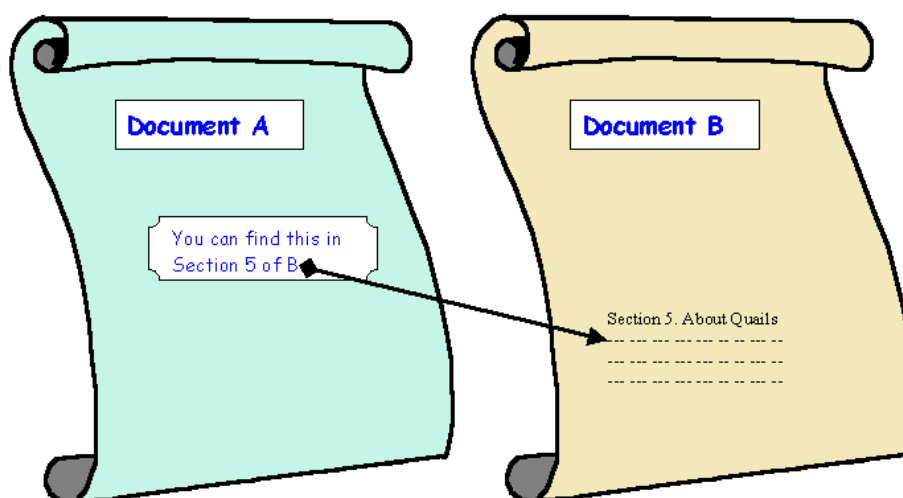


Figure 1: Illustration of Anchor and Link

Take care not to confuse anchor or link. A link is a pointer to another piece of information within the same document or in another document; often you cannot see how that link is implemented (it may be a hidden URL or some other programmed mechanism). An anchor is a fragment of information which the user interacts with in order to access the link. For instance, in a Web Browser the phrase "Click here to return to the previous page" is the anchor which the user interacts with — it contains the link to the previous page.

A hyperlink must have unambiguous reference to the document: this is usually information on the document's location (where in some file space or network it is) and the mechanism to access it (called the communication protocol). In Chapter 3 you will be introduced to HTML anchors and how the referenced documents are identified and located with URLs.

### 1.2.2 Jumps

A hypertext document allows links to portions of the document occurring before the link's anchor. This allows the reader to loop to portions of the document that they have already seen.

The table of contents at the beginning of this chapter is a collection of anchors with explicit links to the internal parts of the chapter. A book's bibliography is another collection of links but it refers to external information. To refer to the internal parts of this chapter is simple: the reader can merely turn to the appropriate page, identified by page number. However, referring to the external information given in a bibliography requires a more complicated effort of searching.

In computer-based hypertext documents, the mechanism to follow a link (the jump) is automatic. Jumping to an external link (another document) is as easy as jumping to an internal link within the same document. As long as the link is sufficiently specified with the name and the exact location of the linked document, the user can directly access the linked document with a simple click on the anchor.

## Basic Concepts

### 1.2.3 Knowledge Additivity

Links can be created to associate related subjects. Therefore the information given can be extensive and wide. The combination of two related subject areas is known as *knowledge additivity*.

Let's say you want to find out how to tailor a shirt using a sewing machine. You would probably look in a book on tailoring a shirt and another on using a sewing machine. The information read would then be linked together in your brain. However, with the hypertext concept, this knowledge additivity would be simpler with association links. You can just continue clicking to read on both subject areas within the perceived single document.

### 1.2.4 Chain of Links

A series of successive jumps constructs a chained path through a series of documents. There is no limit as to the number of jumps, therefore the size of the chain is not constrained.

There may be more than one link in a page and the reader is free to choose any of these links to follow. The path a reader takes will then be different from the path of another reader. Each sequence of jumps forms a different path to fragments of the overall information in the hypertext document. Generally, there is no rigid order to read the information in.

There are two different but complementary purposes of chaining documents via links:

- **Focusing:** each jump along the path, the user can narrow the scope of the search until the fragment containing the topic of their interest is reached.
- **Broadening:** Multiple outgoing links from a document allow the user to broaden their search. This is useful when the user does not have a precise idea of what is being searched for, or wishes to conduct a broad search in a certain domain.

Travelling through hypertext documents usually poses no technical difficulty. However, the reader might experience practical difficulties in retrieving a particular piece of information from a document with numerous alternative links.

### 1.2.5 Loops and Mesh

Just as the reader is free to choose which links and jumps a path through a hypertext document is to follow, it is possible for a user to return to a point previously visited. In other words, loops may exist. A path may even return to the original (home) document. Hence, the structure does not necessarily follow a linear pattern; instead, the documents are connected together in a graph / mesh defined by the links.

This critical property shifts the burden of devising suitable exploration paths from the designer of a hypertext document to the user. This changes the way information is stored and retrieved. Instead of searching directly for information, hypertext allows browsing for information. However, the mesh of information creates difficulty in navigating through the hypertext document.

### 1.2.6 Hypermedia

One of the original purposes for hypertext was the storage and management of textual documents. As computer and telecommunications technology has improved, the capabilities of hypertext systems have been extended to include any digitised media, such as sound and images.

This means that music and videos can be accessed via hyperlinks. This addition of multimedia to hypertext is known as Hypermedia. A combination of text, graphics, video or sound can now easily be interlinked in hypermedia document to offer a rich, often interactive, environment.

### 1.2.7 Authoring Hypertext

The process of preparing hypertext documents or, quite often, of converting a flat (linear) collection of documents into hypertext, is referred to as authoring.

Often an initial collection of documents has to be reorganised by splitting up the original documents into multiple sub-documents. Then links between these new documents must be constructed. Authors of hypertext documents are not only responsible for the content of these documents, but must link documents together, create paths through them, and build references that point to external documents associated to them.

Conceptually, related information is ultimately presented as a single, unique collection of hypertext documents. The remarkable aspect of hypertext or hypermedia documents that distinguishes them from other document types is that hypertext is 'shaped' by the user as he or she navigates the hypertext's network of link. Each sequence of links is a possible exploration path and each chosen sequence forms a single conceptual document for the user.

### 1.2.8 Getting lost in 'hyperspace'

The easy linking of different fragments of information crucial for browsing can produce hypertext documents that are very difficult to use. The user may become disoriented when they do not know where they are in the document and where he can go to. This problem of navigating a hypertext network is also known as being 'lost in hyperspace'. There are ways to minimise the risks of being lost in such a large information space.

#### **Return path**

The user simply backtracks through all the previous documents, link by link, until they reach the one they want to revisit. Alternatively, if the user remembers the reference of the required document, it may be selected from a list of the most recent documents explored.

#### **Home Page**

The starting fragment in a path is known as the home page. This home page is usually a well-defined document that contains the first links to a certain path. It helps to remind the user the path he has taken before and may even serve as a starting point to another path.

#### **Overview Diagrams**

This is the explicit display of the graph / mesh network of documents and links. Many websites have an overview site-map showing the paths the user may take to access certain information from the site.

#### **Guided Tours**

These are suggested paths arranged by the document's authors. Its purpose is to assist the user in the exploration of information in hypertext document. Tour documents form a logical path sequence by using simple 'next-document' or 'last-document' anchors.

#### **Direct jump**

This allows the user to move directly to a portion of a hypertext document. The user has to know the name and location of the portion to directly jump to it. In a Web browser, the *URL* address of the website is typed in and the requested page is retrieved and displayed to the user.

#### **Content-based retrieval**

Some documents may offer a search facility. Browsing for information through the search facility can help narrow the information space to the domain of interest. However, most current search facilities are restricted to textual information only.

## 1.3 The Ultimate Hypermedia System: The World Wide Web

### 1.3.1 Basic Ideas of the Web

The World Wide Web (Web) is a hypermedia system. It has largely achieved the goal of Tim Berners-Lee, its British inventor, of a universal information space. Tim Berners-Lee invented the World Wide Web in October 1994. Thanks to the global reach of the Internet, there is potentially universal access to an enormous volume of documents over the Internet. However, in many developing countries, access is poor, which raises issues of disenfranchisement and disempowerment. Many organisations make publicly available collections of hypermedia documents as part of either their marketing programme, customer service or global operations. Computer suppliers, for example, now publish very detailed specifications of their products via the Web.

Web servers and clients may be located at any part of the world and connected to each other by telecommunication links. If the Web is in some sense a digital library, it is one with no single geographical location. When it comes to commerce, distance begins to lose importance. As long as a supplier can provide goods or services where they are required, the location of the vendor and the consumer will not matter. This gives rise to issues about jurisdiction for taxes, consumer laws, legality of product, etc. This absence of distance is supported by the ease with which Web documents may be located world-wide; the mechanism is straightforward thanks to the way the location of such 'resources' are identified by a Uniform Resource Location (URL). The URL format unambiguously specifies locations of 'documents' on the Web. This location mechanism allows the actual implementation of geography-independent feature of the Web.

## Basic Concepts

Generally speaking, there is no central authority controlling the Web, although fully qualified domain names are subject to controlled allocation, and Internet Service Providers may be subject to the laws of the countries in which they operate. Furthermore, the World-Wide Web Consortium (W3C), headed by Tim Berners-Lee at the Massachusetts Institute of Technology, is influencing — and to a large degree controlling — how technologies are deployed on the Web. The W3C specifies HTML and XML, but other bodies, such as the European Computer Manufacturers Association (ECMA), have standardised other Web technologies, such as what we mostly call JavaScript. JavaScript is a programming language originally developed by Netscape.

Anyone with the appropriate knowledge, and with access to server space, can create a Web document. These Web documents can make reference to any other document. Moreover, a user does not require specific, proprietary software on their computer platform to access the Web, with many Web browsers being free software. While browsers can access and display the information on the Web, not all of them can supply the user with the interactive portions of the Web pages. For example, if Java applets are prohibited or a browser does not support JavaScript, interactivity with the Web document will be limited; some information may even be missing if that information required the presence of these interactive components.

The implications are easy to predict. With different browsers supporting different features, and with the navigation difficulties associated with hypertext's mesh / graph connections, chaos might ensue. However, even the most inexperienced users currently cope and the Web, and it is becoming both a universal world of information, and a universal place for doing business.

Dynamic pages can respond interactively to user input. It is possible to have portions of a hypertext document be produced by a programme as the document is requested. In this way, Web pages are increasingly being used as a front end to databases.

This allows the user to fill out a query and send it off for processing by the hypertext document. The server queries the database using the user's information and returns the output as HTML. To allow data to be sent in such a way to and from Web servers, a standard called the Common Gateway Interface (CGI) has been created. The difference between dynamic Web pages and non-dynamic (so called 'static') Web pages is transparent to the browser and user.

It is also possible to embed programmes inside HTML. When the browser loads such a page, the code is immediately executed. This mechanism supports remote transactions for the commercial aspect of the Web.

### 1.3.2 Fields of Application

The Web began as a tool to share knowledge and has successfully evolved into a general communications mechanism. With the support of transactions and synchronous communications, the Web has application in many different fields.

A primary use is the dissemination of knowledge, which takes many forms. For example, chat rooms and bulletin boards are integral to interactive discussion of all kinds of subjects. Frequently Asked Questions (FAQs) published on Web sites, offer answers to users' questions on how to do certain kinds of tasks. The variety of information that can be pulled out of the Web is wide-ranging.

Education includes a variation of the dissemination of knowledge. Open- or distance-learning programs spearhead this aspect of the Web. Basically, any kind of demonstration on how to carry out certain tasks can be considered education. For example, a user can learn how to create a Web page from the numerous websites publishing such instructions.

With the possibilities of online trading, business transactions are carried out on the Web. The user supplies their order and credit card details so as to buy products advertised on the Web. The Selling module would cover this subject area in depth.

### 1.3.3 The Web as a Digital Library

The Web as a vast digital library is becoming what is known as a 'Global Information Structure'. It will have a profound effect on how we live, work and play. We shall now look into a few of the social implications of the Web as a digital library and a marketplace.

#### i. Different Literacy

The hypermedia concept includes not only text and illustrations, but also music, animation, digital movies, video games and computer software. This diversity changes the form of literacy required when using the Web. The literacy needed when listening to music and watching a movie may be different from that used when reading a book. Less

## Basic Concepts

literacy may be required with innovative ways of using the digital library. For example, software that reads text aloud can assist people with visual handicaps.

### ii. Indeterminate Quality and Value

Editors and publishers employing traditional methods of publishing have little to gain from this type of publishing. As digital works can be copied at low costs, stored in almost no space and transported instantly anywhere in the world, writers can be their own publishers. Therefore, the works published are of indeterminate quality and value. Web publishing may provide no evaluation of work published.

### iii. Specialist Audiences

An article may perhaps interest a group of specialists in a particular field. With the Web, an average reader may browse through the article according to their degree of interest in the field. He or she may not want to be burdened with an additional flood of technicalities, or perhaps would navigate further to extract more in-depth information to supplement a deeper interest in the field

### iv. Copyright Issues and Ease of Purchasing

The ease of copying digital works causes difficulties in protecting copyrights. It may be tempting to make illegal copies rather than finding the rightful owners and paying them a fee. On the other hand, the non-issue of distance and the 24-hour, 365-day activity on the Web means that much can be easily bought through on-line shops. Consumers may come from distant areas or different time zones. With the Web, this market place is open at all times and can serve a very large global region. New technology even allows computational agents to staff the market place rather than people. Therefore, businesses are not constrained by distance or time.

### v. Sense of Place

Despite the irrelevance of distance, an electronic marketplace may be attractive as it *goes* to the consumers instead of them physically moving to the business environment. Its sense of place is created as an illusion for the benefit of the consumers.

### To Do:

Read more on the World-Wide Web in your textbooks.

## 1.4 Summary of Web Terminologies

Here we briefly summarise some of the terms you will need in the module; they will be studied to varying extents in later chapters.

### 1.4.1 Network Protocols

A network protocol is a standard way of regulating data transmission between computers. Just as diplomats adhere to protocols — rules of behavior — when in foreign lands, network communications do the same. They have to obey agreed rules if they are to communicate and 'get on with each other'. After many years of both public and private research and development, two network protocols are now dominant: TCP (Transaction Control Protocol) and IP (Internet Protocol), together known as TCP/IP. These were actually unlikely protocols to be so widely accepted, as faster, standardized protocols had been agreed upon, but none had the same robustness and extensibility as TCP/IP.)

Very often protocols were implemented without any formal acceptance and, because they worked most of the time, they became standards by default. Although TCP/IP is an accepted, de facto standard, work on Internet protocols continue in order to improve communication quality and support the continued growth of the Internet. There is no dictating authority for the Internet. Without a controlling authority, interim proposals about protocol changes are made by groups of interested individuals and then opened up for discussion. Documents containing the various proposed standards are published as Requests For Comment documents (RFCs). You may see references to a specific RFC as the best description of a protocol!

### 1.4.2 Web Application (Webapp)

A *web application* (or webapp), unlike standalone application, runs over the Internet. Examples of webapps are google, amazon, ebay, facebook and the UCT website. A webapp is typically a 3-tier (or multi-tier) client-server database application run over the Internet and it comprises five components:

## Basic Concepts

- HTTP Server: E.g., Apache HTTP Server, Apache Tomcat Server, Microsoft Internet Information Server (IIS), nginx, Google Web Server (GWS), and others. You will learn how to install Apache HTTP and Tomcat web servers in the next chapter.
- HTTP Client (or Web Browser): E.g., Internet Explorer (MSIE), FireFox, Chrome, Safari, and others.
- Database: E.g., Open-source MySQL, MariaDB, Apache Derby, mSQL, SQLite, PostgreSQL, OpenOffice's Base; Commercial Oracle, IBM DB2, SAP SyBase, MS SQL Server, MS Access; and others. You will learn how to install MySQL in the next chapter.
- Client-Side Programs: could be written in HTML Form, JavaScript, VBScript, Flash, and others. You will learn how to write client-side programs using HTML and JavaScript in this course.
- Server-Side Programs: could be written in Java Servlet/JSP, ASP, PHP, Perl, Python, CGI, and others.

### 1.4.3 Uniform Resource Locator (URL)

An URL is needed to locate any resources on the Web. It is an address format that specifies how and where to find a document. The general format is as follows, where the various items in italics must be substituted with part of a real URL, or omitted altogether.

```
http://machine_name:port/path/file_name.file_extension
```

**machine\_name** is either an IP address, for example 137.234.33.89, or a Fully Qualified Domain Name (also known as a DNS name, because Domain Name Servers map between Domain Names and IP addresses), for example, www.apple.com [http://www.apple.com]. In the machine name `http` is the protocol identifier, while `www.apple.com` is the resource name.

**port** is the TCP port to connect to; this is an entry point to software on the server; an optional part of a URL.

**path** is a relative file path from the server's document root; the server will start looking for a file in a specific directory and paths are relative to this

**file\_name** is the name of the file to be browsed, e.g. welcome

**file\_extension** is one of a number of suffixes which, by convention and operating system setup, indicate the type of data contained within the file, e.g. htm, html, txt. For example, in the URL below,

```
http://www.apple.com/retail/business/jointventure/terms.html
```

'terms.html' is a file with the html extension.

### 1.4.4 HyperText Markup Language (HTML)

This language provides the format for specifying simple logical structure and links in a hypertext document. As a markup language, special formatting commands are placed in the text describing how the final version should appear. These formatted documents are interpreted by a Web browser which uses the HTML code to format the page being displayed. Although most professionals use special authoring tools to write HTML documents and to manage sites, developers of e-commerce sites and applications need to know the nitty-gritty detail of HTML, and this is what you will study.

HTML has had several versions over the years. "HTML 2.0" was the first standard HTML specification which was published in 1995. HTML 4.01 was a major version of HTML and it was published in late 1999. Though HTML 4.01 version is widely used but currently we are having HTML 5 version which is an extension to HTML 4.01, and this version was published in 2012<sup>1</sup>. This course will take you through website creation using HTML5.

### 1.4.5 HyperText Transfer Protocol (HTTP)

HTTP is a network protocol used to retrieve documents from a variety of machines in a minimum amount of time. It was invented by Tim Berners-Lee to support a project in developing a distributed hypertext system. Distributed hypertext requires the retrieval of documents from many different machines. File Transfer Protocol (FTP), which predates the Web, would be too slow for this purpose as it is a connection-oriented protocol that requires a permanent connection to a server, thus requiring a connection-maintenance overhead when accessing different machines.

Therefore, to support browsing, HTTP has the following characteristics:

---

<sup>1</sup> [http://www.tutorialspoint.com/html/html\\_tutorial.pdf](http://www.tutorialspoint.com/html/html_tutorial.pdf)



## Basic Concepts

- **connection-less;** a connection is established only for the period of transfer, and the connection need not be maintained after thereafter;
- **stateless;** the server has no 'history' of client visits (although the implementation of cookies overcomes this);
- **comprehensive addressing;** diverse files on any HTTP server world-wide can be referenced via URLs;
- **diverse data;** using extensible MIME-types (see later), HTTP servers can supply information of every possible data types; and
- **rapid;** allows request-response cycles of less than 100 milliseconds

HTTP is not mandatory for distributed hypertext; there are other techniques and protocols that can be used to access or transfer information. However, like TCP/IP and HTML, HTTP is ubiquitous and so enables development in e-commerce.

### Exercise 1

Write down your ideas about the possible benefits of hypertext using the following headings. If you like, go on-line to discuss these with colleagues before writing them down.

- a) Ease of insertion of new information.
- b) Pointers to external materials.
- c) Browsing.

### Exercise 2

Write down your ideas about the possible drawbacks of hypertext using the following headings. If you like, go on-line to discuss these with colleagues before writing them down.

- a) Navigation Difficulties.
- b) No Main Catalogues.
- c) Network Overload.
- d) Link Fossilisation.

Discussions and answers can be found at the end of the chapter

### Activity 1: Analyzing Hyperlinks

Visit your favourite site and try to identify the following:

- a) A chain of links.
- b) A loop.
- c) A guided tour.

## 1.5 The Client-server Computing Model

When you are surfing the Web, you are using a Web browser. When you go to a website for documents, the site delivers them using software called the *Web server*. The browser is considered to be a client in the relationship with the server as it is requesting information services from the server. This is just one particular example of the client-server model of computing. You will learn how to setup your own web server using Windows and Ubuntu in the next chapter.

### 1.5.1 A Definition and some History

The client-server model has been defined as:

*A software partitioning paradigm in which a distributed system is split between one or more server tasks which accept requests, according to some protocol, from (distributed) client tasks, asking for information or action. There may be*

## Basic Concepts

*either one centralized server or several distributed ones. This model allows clients and servers to be placed independently on nodes in a network.*

Client-server computing is mainly about the client computer possessing its own computing power. In the days of mainframes, all the processing power took place on central computers. The client 'terminals' were little more than a television that could send and receive characters. When microprocessors became available, it was possible to make the terminals more powerful so that they could handle some of the processing. Over time this has meant that mainframes have been replaced by smaller server machines and terminals have been replaced by more powerful client workstations.

The client-server model provides a good division of processing power, since the server primarily provides information to the client, which is responsible for interpreting and displaying it. This means that servers do not have to be powerful machines, allowing more people to become service providers.

A more important characteristic is that because the client-server model provides for significant processing power at the (remote) client end, the operator of the client system has considerable autonomous power in contributing to the enterprise of which he or she is a part. This means that local decisions can be made possibly faster than if they were made remotely and action taken.

You may hear client-server computing being talked about as a modern computing 'paradigm'. Other than being part of a sales pitch, this is likely to mean that the model has made a significant impact on, and change to, the way we design and use computer systems. In particular, it is the current model for distributed business systems, and fits nicely into the emerging Web.

### 1.5.2 Functionality

In the context of the Web, users run client programs (i.e. Web browsers) which provide the following functionality:

- They allow the user to send a request for information to the server.
- They format the request so that the server can understand it.
- They format the response from the server in a way that the user can read it.

Server programs carry out the following:

- They receive a request from a client and process the request.
- They respond by sending the requested information back to the client.

In summary, the typical functionality of a client-server model is:

- A user, via a web browser (HTTP client), issues a URL request to an HTTP server to start a webapp.
- A client-side program (such as an HTML form) is loaded into client's browser.
- The user fills up the query criteria in the form.
- The client-side program sends the query parameters to a server-side program.
- The server-side program receives the query parameters, queries the database and returns the query result to the client.
- The client-side program displays the query result on the browser.
- The process repeats.

### Exercise 3

The client-server model applies to a lot of things outside of computers. Imagine going to a bank to withdraw some money? Who is the client and who is the server? Clearly, you are the client and the bank is the server. One of the advantages of the client-server model is that one server can handle many clients. The teller in the bank (server) handles many customers (clients). Also, you can use lots of different servers to get the service you need; that is there are a lot of tellers, and for that matter, bank branches and cash machines.

## Basic Concepts

For any website, say the University of Cape Town Computer Science website [<https://www.cs.uct.ac.za/>] or the University's Vula site [<https://vula.uct.ac.za/portal>], think about the following questions and write down your answers:

- a) Are there multiple clients?
- b) Who are these clients?
- c) Are there multiple servers?
- d) Why would there be multiple servers?

Discussions and answers can be found at the end of the chapter.

### 1.5.3 Information and Processing on the Web

Information is passed from the server to the browser. This information may be in the form of HTML documents, GIF files, Excel spreadsheets, movies — just about any digital content.

Information can also be passed from the browser to the server. When you click on a hyperlink you are sending information to the server, and when you fill in an online form, you are usually sending information to the server.

In addition to passing information backwards and forwards, some processing can also be done in the browser. For instance, you might have a simple Web page that calculates the overall cost of a loan once the initial value of the loan, the interest rate and the length of the loan have been entered.

But where does the processing take place? Does the server process the information and generate the result, or is it the client that processes the information? If the client does the processing, then this is a client-side application; if it is the server, it is a server-side application.

In the loan example above, the client has the information (the principle, rate and time). It could send this information to the server to process the information, generate the result and send it back to the client. Alternatively, the server could send a program to the client that will carry out the processing. In the latter case, since the client has all the information and the program is pretty small, it is probably better to run the application on the client side.

Of course, there is also a problem of who has the information. If the server has a database, and the client wants to query it, then there are two possibilities. The server could send the database and the querying program to the client to process it or the server could process it and simply send the result. In this case, it would probably be better to do the processing on the server side.

To summarize, where the processing is undertaken largely depends on where the information is, but it also depends on the processing loads of the machines as well as the size of the program being run.

#### Exercise 4

On the East Med. Trading Co. website, they would like to display to the user the number of pages that he or she has visited at that site. Think about the following questions and make a note of your answers.

- a) What data is needed?
- b) Where is the data stored?
- c) Should this be a client or a server side application?

Discussions and answers can be found at the end of the chapter.

### 1.5.4 MIME Types

A browser receives binary data from the server which it has to cope with. How does it know if the binary data is an HTML document, a GIF picture file or something entirely different? Even if it does know what kind of document it is, how does it process it? The answer to this is MIME types.

## Basic Concepts

MIME types — Multipurpose Internet Mail Extensions — were created to identify the differing types of possible email attachments. The MIME types have been extended to include new multimedia types as they have been introduced, and are now used with a variety of protocols including HTTP. When information is sent to a browser, a MIME header identifies the file type of the document. Attaching a MIME type to a file allows the browser to process the file's contents correctly without the browser having to guess at the data type from the file's extension. This is important, since while MS-DOS files require a three letter extension to identify a file type (and Windows XP uses a similar file extension system), not all operating systems do this.

This MIME header information has the following format:

```
Content-type: type/subtype
```

where

**type** is one of several general types such as: text, audio, image, video, application etc.

**subtype** is a more specific designation. This is a large and ever expanding category.

Some examples of MIME headers are

```
text/HTML video/MPEG  
image/GIF
```

### Processing MIME types

Mime types are processed as follows: Somehow the HTTP server must decide what type a file is. The server administrator can provide this information by instructing the server to map file extensions to certain file types. The server administrator must therefore supply a list of all the different file extensions for the files found on the server along with the equivalent MIME types for each of these file extensions.

The client browser must also be configured to know how to deal with these different types. Most browsers have been preconfigured, but they sometimes need to be updated to deal with new file types. On Google Chrome, for instance, the configurations of file type associations can be done under Settings. Files with different types can be viewed as follows:

View it in the browser. Files such a GIFs, JPGs etc. can all be handled by the browser

Use a plug in. Plug-ins are special pieces of code that software companies distribute to allow browsers to cope with new file formats. For example, plugins allow Google Chrome to use more features such as viewing animations using Flash.

Launch another application on your computer that can process the format.

If all else fails, the file can be saved to disk until a suitable program is found.

### Exercise 5

Suppose that there is a new document type that needs to be displayed on the client's computer and that you need to introduce a new MIME type for this document type. Let us use Microsoft Excel charts as an example (although this document type is obviously not new) and write down all the options to be considered on how this could be achieved.

Discussions and answers can be found at the end of the chapter

### 1.5.5 Web Servers

There are lots of different types of servers: FTP servers, gopher servers and, of course, Web servers. In this section we will concentrate on Web servers. What do Web servers need to do? Their functionality can be summarised as follows:

- They need to provide HTML pages (with an appropriate MIME type header).
- They need to provide other types of documents (also with an appropriate MIME type header).

## Basic Concepts

- They may need to process information from the user. For instance, if the user submits information to the site, the Web server must either process and store that information, or pass it on to another programme which can do so.
- They supply dynamic data (such as in response to user supplied information).

Processing user information and supplying dynamic data is complex. Many servers do not provide this facility. While complex to implement, it does make the server more dynamic and useful.

User information can be processed on the server using server-side applications called CGI (Common Gateway Interface) scripts. Many other languages and interfaces also exist, e.g. Java Servlets and PHP.

The server passes the user's data to the CGI program which then processes it. This program may dynamically create an HTML file to be sent back to the client just as standard HTML stored on the server would be. (Note, this will be discussed further in the units on JavaScript.)

In the next chapter, you will learn how to set up your own web server. Apart from learning how to set up a web server, you may wish to have a web server at home where you can store your files and then you can access and share these files with your colleagues at work.

### 1.5.6 Distributed Processing

Client-server computing is concerned with distributing the load of information and processing. Until about 20 years ago, most information was stored on one computer — the same computer on which all the processing was done. The only reason an extra copy of the data might have been kept on another machine was for security or backup purposes. If many people needed either the data or the processing, they would get another computer and copy the data.

With client-server computing, a given machine acts both as a client and as server; that is, it can run both a Web server and a browser client. It can also run processes (i.e. programs) on other machines. Network technology has enabled this distribution of processing and data.

The goal of distributing processing is to reduce the overall time that is needed to process some information. For example, consider this: one machine (named A) is connected to two other machines, B and C. If there are three processes to run, they can all run on A. If each machine requires 10 seconds of processor time in order to complete, then it will take a total of 30 seconds of user time to run the processes on one machine. But if B and C are each asked to run a process as well (so that now three machines are being used), then the total processing time has been distributed, and while it still takes 30 seconds of processor time to complete the work, it only takes 10 seconds of user time. It is therefore three times faster.

However, there is an additional cost that was overlooked in the previous paragraph. If A has to ask B to run a process, some communication time between the machines is required. For instance, just sending a message takes a certain amount of time, and this assumes that computer B already has the necessary data and programmes to run the process. If not, A may have to send the data and possibly the programme as well. Additionally, time is also required for B to send the results of the processing back to A. (The same is true for Machine C as well.)

For simplicity sake, let us say that sending the data and the results each takes one second. In the first second, A sends the data to both B and C, and A starts processing. In the following second, B and C begin processing. At the tenth second A finishes its processing. At second 11, both B and C finish processing their data and send their responses to A. In second 12, A receives the data and everything is completed. The total time to run the three processes is 12 seconds

Now try some process balancing in the following exercise.

### Exercise 6

The table below shows a list of processes (P1-P6) and computers (A-E) on which their data currently resides. Each process will output some result after a given amount of execution time has passed (as listed below). Processes can only execute on those computers which contain all of its data. The amount of data the processes require (in megabytes) is also given. Note that in some cases the data is already present on multiple computers. This data may be transferred to other machines at the rate of 1 MB per second. After the data has transferred, the process may then run on that machine. The computed results may also be transferred to another computer taking one second of time. All the machines are directly connected to each other and are otherwise identical. Each computer can run only one process at a time, but after a process completes may execute another.

## Basic Concepts

Of the five computers, computer A wants the results from four of the processes: P1, P2, P3 and P4. Computer B wants the results from P5 and P6, and computers C, D, and E are essentially idle, wanting no results from any of the processes.

Programme	Run Time	Location of Data	Size of Data
P1	5 seconds	A	8 MB
P2	6 seconds	D and C	4 MB
P3	7 seconds	A and C	5 MB
P4	8 seconds	C	12 MB
P5	9 seconds	A and E	2 MB
P6	10 seconds	B	2 MB

1. Come up with five different ways of distributing the processing, and the total user time for processing. For example:
  - a) machine A runs P1 (5 seconds).
  - b) machine B runs P6 (10 seconds).
  - c) machine C runs P3 (7 seconds plus one second to transfer the answer back to A for 8 seconds).
  - d) machine D runs P4 (12 seconds to send the data from machine C, 8 seconds for processing, and 1 second to send the answer for a total of 21 seconds).
  - e) machine E runs P5 (9 seconds to run and 1 second to transfer the answer to B for a total of 10 seconds).
  - f) After running P3, machine C also runs P2 (6 seconds plus one for transfer for a total of 7), bringing the amount of time that machine C is occupied for up to 15 seconds.
  - g) The longest time taken is for process P4, which takes 21 seconds to complete. This means that the total time for obtaining all the required results is only 21 seconds.
2. What is the least amount of time required to execute all six processes and send their results to the machines which want them? Is it possible to complete all of this work in less than 12 seconds?

Discussions and answers can be found at the end of the chapter

## 1.6 Review Questions

The answers to the review questions can be found at the end of the chapter.

- a) Is it relatively simple to insert new information in hypertext? \
- b) Is hypertext different from a hyper-document?
- c) Explain the reason why it is difficult to retrieve required information with unlimited chaining of information.
- d) Explain why Knowledge additivity enhances the learning process.
- e) Here is a summary of what hypertext and hypermedia are about. Fill in the blanks.

A hypertext document is a \_\_\_\_\_ document that implements anchors containing \_\_\_\_\_ to connect various fragments of information into one \_\_\_\_\_ network.

\_\_\_\_\_ is an extension to hypertext that includes digitised sounds and moving images.

The user is free to choose which links to follow in multi-linked hypertext documents. Each sequence of constructs a unique \_\_\_\_\_.

Authoring hypertext requires the decomposition of documents into fragments of information and then the construction of links between the\_\_\_\_\_.

The Web is an example of\_\_\_\_\_hypermedia. There is no central authority \_\_\_\_\_ its development and its information content is geographically-\_\_\_\_\_. The ease of linking information is one of the major benefits of hypertext.

Navigation difficulties through the various possibilities of different is the main drawback of hypertext. Extracting the required information can become tedious.

- f) What is so important about the client-server model of computing?

## 1.7 Answers to Exercises

### 1.7.1 Exercise 2

The following are some thoughts on the difficulties with working with hypertext

#### i. Navigation Difficulties

Navigation is the main drawback of hypertext. As the document is interlinked and may loop, readers can easily lose track of where they have been and where they are. The freedom to choose to follow any of the links may take the reader away from the item being sought. There are insufficient clues as to where an anchor links to.

#### ii. No Main Catalogues

A catalogue is readily available in a physical library and the user can easily find out whether the book he requires is available or not. It is difficult to index a hypertext document due to multiple links within a document, unless the reader is guided to a certain sequence of links.

Therefore, extraction of required information is tedious, especially in large hypertext documents. Search facilities attempt to make this easier. However, when searching a large content of information, such as the Web, the results of the search itself may be extensive. Moreover, these search facilities usually only recognise text descriptions and not the multimedia content of the document. This undermines the benefits of hypermedia.

#### iii. Link Fossilisation

Computer names may change and linked documents can be moved to other host computers. Hypertext requires explicit pointers to the names of the computers and the files on these computers, or the fragment of information that is linked to will no longer be accessible.

### iv. Network Overload

Hypertext content assumes universal coverage and infinite transfer capacity. The capacity of the telecommunications network may not be sufficient to cope with the usage without penalising or compromising other network activities. This happens when the reader is not warned of the document size, or is not conscious of the network implications. This results in a technical halt or slowdown in navigating through the hypertext.

#### 1.7.2 Exercise 3

- a) There are multiple clients for a website.
- b) The clients are all the people who visit it, or more precisely, they are the browsers used to view the site.
- c) There are no multiple servers for this site.
- d) For heavily used sites, the site is copied to another computer, called a mirror site. "Mirrors" are used to reduce traffic to the base site. Overall net traffic should also be reduced, as clients will go to a mirror that is closer than the base server. Mirrors add redundancy and make the site more likely to always be available. If one mirror goes down, other mirrors are likely to remain up.

#### 1.7.3 Exercise 4

The following might be needed to give a history of site usage.

- a) The data that is needed is a list of all the pages that a given browser has visited on the site.
- b) What the server can do is maintain a list of all the people who visit the site and which pages they have visited. This can be stored on the server.
- c) Alternatively, JavaScript code could be used to maintain the information on the browser using cookies. This is probably more difficult, and would only work for the current visit. (When the browser application was halted, all the information would be lost.) So, it would be better to store the information on the server side.
- d) If the data is stored on the server, then this should be a server-side application. The server can process the information and simply return a number representing the number of pages the user has visited. That number is returned to the browser. The alternative is to send all of the data (all the pages/user pairs) to the browser and then carry out the processing there.

#### 1.7.4 Exercise 5

The server will append the new type onto the information it sends. The server needs to know the type, but that should be fairly straightforward as the server is providing the data, so the administrator will have set that up. The client needs to know what the type is, and have a method of displaying the data. The browser itself could display the data, e.g. like a GIF file. The browser could use a plug-in. If it were a new file type, the group who developed the file type might provide a plug-in to read the type.

The other options are automatically invoked from current browsers. Since the browser does not know what to do with the file type, it puts up a dialogue box to ask the user. If the user knows which application to invoke, the user names that application that is then invoked and passed the file (the Excel chart file in this case). Instead of opening the file, you could execute Excel, open the file, save it, and deal with it later.



### 1.7.5 Exercise 6

Discuss the results of exercise 6 with your colleagues studying the module. In particular, you should be able to state the smallest amount of time which you were able to run all six processes in, and be able to explain your results.

### 1.7.6 Review Question 1

It is easy to create anchors which will link fragments of information together in a hypertext document. There is no real sequence to the information. New fragments can be inserted anywhere in a hypertext document as long as the anchors are properly implemented to link the new and existing fragments.

### 1.7.7 Review Question 2

Hyper-documents are hypertext documents. They are the same thing.

### 1.7.8 Review Question 3

If the hypertext document is small and does not contain many external links, information can be retrieved quickly with the browsing feature. The difficulty arises when the hyperspace is vast and there are many links in each page. Numerous links imply the possibilities of many different exploration paths. This makes navigation through the network of documents for the required information tedious.

### 1.7.9 Review Question 4

*Knowledge additivity* connects different aspects of information from different fields of study together, therefore the information is more useful. Let's say X=hunting skills and Y=using bows and arrow. Take a reader who wants to know how to hunt well with bows and arrows (Z). It is possible to achieve this with knowledge additivity in hypertext. ( $Z=X+Y$ ).

### 1.7.10 Review Question 5

A hypertext document is a non-linear document that implements anchors containing links to connect various fragments of information into one mesh network.

Hypermedia is an extension to hypertext that includes digitised sounds and moving images.

The user is free to choose which links to follow in multi-linked hypertext document. Each sequence of links constructs a unique navigation path.

Authoring hypertext requires the decomposition of documents into fragments of information and then the construction of links between the fragments.

The Web is an example of networked hypermedia. There is no central authority dictating its development and its information content is geographically-independent. The ease of linking information is one of the major benefits of hypertext.

Navigation difficulties through the various possibilities of different paths is the main drawback of hypertext. Extracting the information required can become tedious.

### **1.7.11 Review Question 6**

It is important because it allows for significant processing power at the remote client end so that the operator of the client system has considerable autonomous power in contributing to the enterprise of which he or she is a part.