Unit XIV
The Internet

Unit XIV
The Internet

In this unit, you are asked to do a little bit of exploring on the Internet, and to learn how to set up a simple web server on a Macintosh. In the process, you will learn a bit about how the Internet works, and how it is related to some other terms, like bitnet and Ethernet.

This notion of cross-country networking of educational institutions was the brainchild of a group known as Educom, a consortial organization supported by educational instutions, started in Princeton, NJ in the seventies. The idea was simple; a sort of daisy chain of phone links between major sites, with a routing diagram and a protocol for electronic messages. This system was known as bitnet. This was run in parallel with some military networks, and some commercial networks. The idea was, you could send a message or file from one point to another as follows: we at Grinnell would send a message to the local major hub, which I believe was at Iowa State University. There it would be held by the local computer for a while, which might then call up the machine in Madison Wisconsin, and send it to there. Madison would then call up say Carnegie Mellon, which might then forward it to its destination, say Dickinson College in Pennsylvania. You get the idea. But it depended on the intervening machines to forward the message, which could be delayed if a machine was down, or particularly busy. So messages went, but it could take from minutes to perhaps hours. Even so, it was a tremendous step forward, because it was now possible for machines made by Digital Equipment to talk to machines made by Data General, and IBM, and Harris, all across the nation.

Two important changes have occurred since about 1990. First is the merger, or more properly interconnection, of the educational network bitnet with other networks, including private industry networks and government networks. This consolidation is what is called the Internet. The second important change is the development of more and more dedicated communication hardware--cable, fiber, satellites, computers--so that communication over the Internet can happen in "real time." What this means is that a message can be sent over the Internet and a response returned before the user falls asleep: a period of seconds to at most a few minutes instead of minutes to hours. This has made possible interactive use of cross-country and world-wide networks. We are now still at the beginning of this revolution, and it is already making a change in network use that is reminiscent of the change in computer use that ensued when computers became fast enough to handle multiple terminals rather than just reading decks of cards submitted at a window. Ask anyone who went through that transition; it is the computing equivalent of moving from the Spirit of St. Louis to the Space Shuttle!

Although it has been less important to the individual user, we have also seen a tremendous shift from mainframe and minicomputer usage to local networks (LAN's or local area networks) of individual microcomputers (like our own little network of Power Macs) and workstations (like MathLAN). These are typically connected together by wires through a set of hardware conventions like "Ethernet." Ethernet refers to the actual type of wiring, as well as the rules for what signals mean what over that wire, so Mac's can talk to PC's can talk to SUN's. However, this is only used locally, and is not fast enough to talk off campus.

These changes gave rise first to an interactive network protocol known as Gopher, which was developed at the University of Minnesota (not a surprise to you college sports fans). This set up rules for how to establish a menu page. There would be a gopher "server" program at a source site that would have a bunch of information that others might want. At your computer, you ran your own gopher "client" program that would send off requests for information, and present the responses in menu form that allowed you to search down to finer detail. In addition, there was a file downloading system known as FTP (File Transfer Protocol) that had a similar set of rules for passing files about. These files, if they actually were programs, often came in an encrypted form, because some of the individual character strings used in most computer programs have special meaning over the Internet, so those characters had to be harmlessly translated first, and then translated back on your local machine. Again, another program and set of rules, worse yet, machine specific (i.e. a Mac translator for Mac programs, a PC translator . . .).

All of these have now largely been combined and expanded in the World Wide Web. The WWW was developed at CERN (nuclear and particle laboratory in Switzerland), and expands gopher-like capabilities to make the equivalent of menu selections from individual words in a page. This selection process is more subtle than simple menus, and naturally has its own name--hypertext. It allows links to any other WWW page anywhere, rather than just within your own document. The WWW also includes all sorts of multimedia goodies (like graphics, sound, and video), and generally includes lower level features such as gopher, FTP and electronic mail. The most prevalent tool both at Grinnell and elsewhere in the US for using the web is a program known as Netscape, although things change so fast in this business that we may see a successor in half a year!

First, you should learn how to simply use Netscape to connect to a WWW site you know about. Each of your machines has a copy of Netscape 3.0 (not the latest version, but fine for our purposes). Start the program up, and you will find yourself connected to the Grinnell Library home page, which I have specified as the default starting point for Netscape on our machines. This page is specified by the library home page URL (Universal Resource Locator): "http://www.lib.grin.edu/", which you see entered at the top of the page. This page gives you a bunch of interesting topics, but not much information yet. Somewhere near the top are the words Grinnell College Home Page. Click on this line, and you will suddenly switch to the page maintained by Computer Services (do you sense turf wars here?). Now try clicking on the "Back" icon up near the top, and you should jump back to the library page. That was fun. From the College home page, you can jump to some admission stuff under "Faculty and Curriculum" and get a nice view of my receding hairline (at least you could a year or so ago)! Yikes! Jump back, and then go to academic programs, and see if you can navigate your way to the Physics web page. Much better.

Now that you have found the physics page, let's make a "bookmark." This stores the URL for this location conveniently in the program so you can get back to it with a more familiar name, like "physics page." Once you are on the page, select the "Add bookmark" command from the bookmark menu (it has a funny icon that looks vaguely like a bookmark). Open the "bookmark" menu again, and notice a new bookmark title has appeared at the bottom of the menu, but it has an obscure name. You can fix this by selecting "Bookmarks" under the "window" menu. This gives you a new window titled bookmarks, and a new set of menus. Select the new bookmark by clicking *once* on its name in that window. Under the "Item" menu

choose the "edit bookmark" command. This gives you a window in which you can change the name to "physics page" or whatever you find handy.

There is a lot to explore locally, but most of the interesting stuff is "out there on the web." Let's try something a little more exotic:

http://hepth.hanyang.ac.kr/index.html

which should connect you to a physics home page in Korea. In fact, a lot of the entries at this web site are actually taken from locations here in the United States.

This is all well and good if you have a URL to start with, but what if you want to find out what is out there about a specific topic? Often you just want to find some company, but you don't know their web site URL. Often companies are very clever and actually use their real name. Netscape is clever too, and makes some reasonable guesses about what the URL really is. For example, say we wanted to find out about some Panasonic products, as I did a few weeks ago. Where you had previously typed

http://hepth.hanyang.ac.kr/index.html

now just type

panasonic

without any http:// or www or anything. With any luck, Netscape will find it for you. Similarly, you might want to know more about Mathematica. You just need to know that Mathematica is produced by a company named after the owner/founder, Steve Wolfram. Just type

wolfram

and see what you find.

Often, of course, you want to find web sites that don't have names you can guess. Well, there are searching tools that help you find things on the web. There is a catch here, however. You saw how long it took to connect to a single web site--some significant fraction of a minute. It would be impossible for a computer to actually search all the web sites by interrogating them all; it would take so long the information would be out of date by the time you got it! So, the search has to be done by a computer somewhere that has compiled a catalog of web pages. But how many web pages? And through what amount of the text can you search? Are only titles searched, or just home pages? And who selects those pages? And then there will be the compromise of breadth of entries versus speed of search. So, as you might imagine, there are a variety of web search tools, with a variety of virtues. The library home page has a nice entry that leads you into this. Select the heading of search tools, and then select the phrase "a guide." This then gives you some information about the different search tools, and their various pros and cons.

Let's see how this works. First, let's guess that NASA has a cool web site that includes space pictures in it. Let's try using Infoseek to search for this. Get to Infoseek from the library home page, and then just plunk in NASA in the search window, and hit the search now button. You will get LOTS of "hits," but they are ranked by some mysterious process. Look through the top ten, and from the URL's you should be able to find one that looks like a real NASA product. Alternatively, there are a variety of people and organizations that put together their own list of good web sites that you might explore. I came up with "Scott's Internet Hotlist" and the "Internet Pearls Index." By any number of routes, you should be able to easily come up with some nice NASA pictures from the Hubble Space Telescope. Give it a try.

By now you may be wondering, how does all this work? Well, let's learn a little bit about it by setting up our own web servers. Many of you may have your own web pages on the VAX already, or perhaps on MathLAN. In this case, the VAX or a MathLAN machine acts as the server, and you can put your own information, and even pictures in your personal account. However, it is a bit cumbersome moving things to the account from personal computers, a scanner, etc, and space is at a premium. We have some very simple shareware server programs that run quite well in the background of a Mac (similar programs of course exist in the PC world too). I will pretend that you have just downloaded such a program from a web site; it has arrived in a compressed form called NetPresenz-400.sit, which you should find on your hard disk. Click on this twice, and it will start a self-extracting program (restoring all those control characters that the internet doesn't like that I described earlier, and decompressing other sections of the program) that will create a full folder of goodies.

Look in this folder, and you should find some "Quick Start" instructions. See if you can manage to follow them. I have made sure that file sharing is turned on on all these machines, but you must create folders etc. as directed. After making your "Pub" directory, you need a file to share over the web! The web server will look for a file called index.html unless it is given more specific instructions, so we should make this file first. Start up Microsoft Word, and type in the following instructions in HTML (HyperText Markup Language). These are the instructions that are passed to the client machines running Netscape or a competitor program that tell the client how to make up the page--in terms of text, graphics, sound, video, and formatting. Feel free to modify the entries, but preserve the HTML code in the angle brackets < >.

```
<html>
<head>
<title> My Title </title>
</head>
<body>
<h1> Hello World </h1>
This is nothing really important.
<p>
</body>
</html>
```

You can probably discern the structure of header, title, and headline. Notice that many HTML commands have starts and ends, like <body> text </body>. Once you are finished with this file, save it with a "SAVE AS" command so you can select the "TEXT ONLY" option. Give the file the name "index.html" and make sure that it is in the "Pub" folder.

Now you can continue with the Quick Start instructions. Make sure to ask for help if you need it. When you think you have things set up, and NetPresenz is running, switch to Netscape. Try to find your own web page, but first you need to know your URL. It should be

http://132.161.114.xxx/

where xxx are three digits that specify your specific machine as follows:

Comp Phys A: 139
Comp Phys B: 187
Comp Phys C: 138
Comp Phys D: 136.

Type in your URL, and see if you can get a glimpse at your web page. You may want to check out those of your fellow students as well.

You may ask, why that funny number? That is the IP (Internet Protocol) address, which specifies your machine among all machines in the world on the Internet. The 132.161 part says Grinnell College to the outside world, in the same way as the words grin.edu. As you might imagine, these have to be registered and indexed properly so that the routers know where to send the information. We depend on GCCS in a similar way to generate complete IP addresses (GCCS makes up the remaining six digits) for individual machines; they are also the only ones that can create nice sounding names for those machines that will work instead of the numbers. If we were making these servers permanent, we would ask them to make names for them; however, we will remove these servers after this week.

So now we have created a "home page" for each machine, but this is hardly a web! We need those links between pages. So, we need a second page. Try something like the following

```
<html>
<head>
<title> Second Page </title>
</head>
<body>
<h1> Welcome Summer! </h1>
This is a link back to <A HREF ="index.html"> my home page </A>.
<p>
</body>
</html>
```

Save this file as "page2.html" as text only in the "Pub" directory. You should be able to find it now from Netscape by specifying the URL

http://132.161.114.xxx/page2.html/

where xxx is the correct three digits for your machine.

Play around now making other links, such as from your home page to page 2. NetPresenz assumes that if you only specify a file name, that it must be in your Pub directory; if you give a full URL, it allows you to link to any other place on the web. Try to make a link to one of the other group's pages, then try a link to somewhere far away.

Clearly this is just a start. You can learn more about the full HTML command set most easily from the web itself; one good site is

http://www.mcli.dist.maricopa.edu/tut/lessons.html

If you wish, you can see about including graphic elements such as a scanned image into your pages.

Another great way to learn about HTML tricks is to actually look at the code that is sent by a server for a page you like. You can ask Netscape to display the actual HTML code instead of interpreting it, and figure out what commands do what nice things.

You also may wish to have your pages do more than just dump information out. For example, businesses might want to process orders, or give price estimates over the web. This then has to respond to words and numbers typed by the client.

There are two basic ways to do this. The first is called "cgi's" (an acronymn that I have forgotten), in which the web server takes in these numbers and words, and then calls up a program (written in a regular programming language such as C, or a macro type language like AppleScript) which interprets them, does calculations or whatever, and gives some results back to the server to send back. The second solution involves sending an actual program to the client computer. This is a bit tricky, because you don't know what type of computer that might be. The Java language addresses this, by being written for a "virtual machine," so all sorts of different machines (PC's, Mac's, UNIX workstations) have Java interpreters or compilers that make these programs run once they get there. This is all great, but opens the door for creating viruses. These sorts of refinements of the web are beyond what we can accomplish here, but are good to know about!

Finally, you should be aware that there are many programs out there now that allow you to generate beautiful web pages without having to know much, or necessarily anything about HTML. For the Mac, one popular choice is Page Mill. Many word processor programs, or even programs like Mathematica, can generate some HTML code. Will there be any need for anyone to actually know HTML ten years from now? My guess is it will become a vanishing skill, much as familiarity with assembly language programming largely has evaporated since I was in graduate school fifteen years ago (as testimony to that--how many of you actually know what assembly language even means!).