



Fundamental notions of programming (NFP)

Class 10

Debugging

After being developed, and before being released onto the market, a program must be thoroughly tested. This consists into observing the relationship between inputs and outputs within that application to ensure the entire program works perfectly well. Different data should then be used to make sure the program provides the expected responses.

If everything goes well, the program can be released. But let's be honest, that rarely if never happens. During the testing phase of any applications, it is common to find unexpected behaviours (bugs) which are listed in a errors log book so they can be looked after and corrected.

Defensive Programming (Bug Prevention)

Some programmers may be eager to code as fast as possible so they can come up with something to show as soon as possible. Such a manner usually results in a preliminary application containing several bugs that will need to be corrected. This is mostly why the main part of production time is generally dedicated to fixing code rather than writing it.

A better approach to developing applications is called *defensive programming*. This approach promotes to work more carefully and with proper planning instead of rushing into coding only to finally have to correct several bugs and sometimes having to rewrite huge parts of an application.

The goal here isn't to become a very good debugger but rather to be a good programmer whose code won't be needing as much debugging time.

Debugging Techniques

Whether you adopted the defensive programming approach or not, you will undoubtedly need to fix bugs you've created (the code doesn't write itself, so it's never the computer's fault).

There are three possible types of errors:

Syntax errors :

This simply means you wrote the code wrong. This could be a spelling mistake or a missing character. The most common mistakes are misspelling variables names and forgetting to insert a semicolon at the end of a statement.

Logical errors

This one's a little more complex. Here, it isn't about misspelling but about a logical flaw. This could mean a function is called or an object is created in a wrong moment (or scope) in your application. To solve such an issue, it is needed to read the code trying to understand precisely the flow of data.

Runtime errors

A runtime error means the application cannot be executed. This could be that you have done a *divide by zero* situation or that your program is trying to open a program that doesn't exist.

Debugging Output Statements

One of the best ways to perform debugging is to output content (messages or logs) throughout your application. In this case, it is possible to follow how far the application went (and if it skipped something) before the error happens. This gives an idea of where to look for the bugs.

Flags

A flag is an output that is created in order to make sure a certain block (or part of the code) was «entered» (executed). For instance, a variable on a greater scope with a false value and in a specific condition or loop can be created and its value may be changed to *true* to simulate some instructions were executed correctly.

Comment out

A common technique consists in commenting out part of the code though of breaking or providing wrong information. This is some sort of *divide to conquer* technique. Fragmenting the code into smaller chunks helps to pinpoint the bug(s).

Desk check

Desk checking means to manually going through the code, as if you were the application. This is a good way to find logical errors in situations where the application has no technical errors, but the results are still not those expected.

Databases

So far, we have explained how applications basically are bunches of data being captured, processed and output. But what happens when the application has stopped being used? How come every time an application is restarted, it seems to exactly be where it was left off?

Information being physical, it needs to be stored somewhere. It is usually done in a location specifically created for this purpose: a database. The main purposes of a database are to store and organize data, allowing users to find a piece of information as quickly and easily as possible.

Data organization in a database

Elements of individual records are called fields. In a periodical database, for instance, records could include informations about a periodical article (author, source, date, title, etc.).

The structure (organization) of a database is quite similar to an Excel table. Each column of the table represents the fields and each row represents the individual record.

Here is an example of the database of a class's students :

LAST NAME	FIRST NAME	CLASS	HOMETOWN	BIRTH MONTH	GENDER
Smith	John	Freshman	Macon	August	Male
Turner	Evelyn	Freshman	Brunswick	April	Female
Arndale	Carrie	Sophomore	Atlanta	September	Female
Zwemke	Jason	Freshman	Cartersville	May	Male
Lane	Penny	Sophomore	Decatur	September	Female
Guoqing	Yu	Freshman	Augusta	April	Male
Jones	John	Junior	Waycross	December	Male
Vasser	Pricilla	Freshman	Blue Ridge	February	Female
Wright	Mandy	Freshman	Rome	September	Female

Let's suppose we would want to know the hometown of Penny Lane. It would be need to logically ask something like: «*Database, tell me the value associated to HOMETOWN in the record where the LAST NAME is Lane and the FIRST NAME is Penny*». The database should then return the value: «*Decatur*».

Saving Data to Files

Databases and applications are not the only places where data can be stored. Data can actually be stored in different types of files such as XML, CSV (Comma Separated Values) and TSV (Tabbed Separated Values). This is actually possible in JavaScript (Node.js – a JavaScript runtime) and it would look a little like the following :

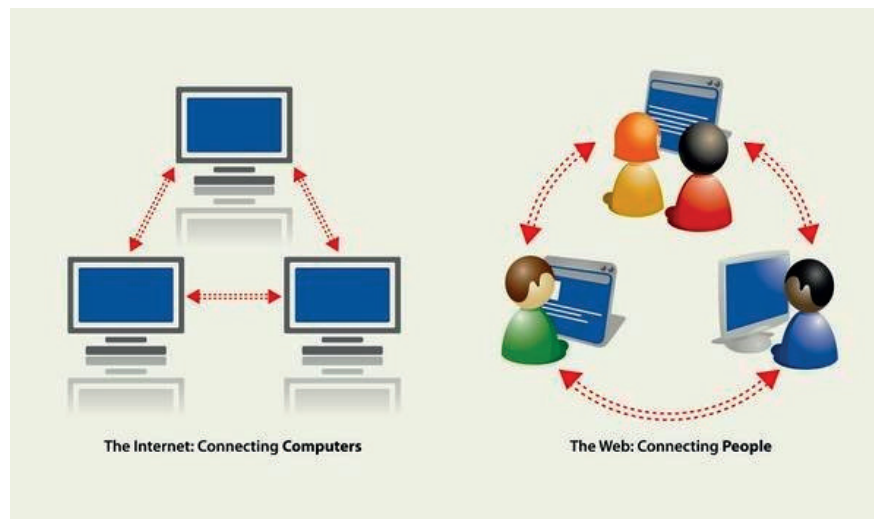
```
1. var fs = require('fs');
2. fs.readFile('/etc/passwd', function (err, data) {
3.   if (err) throw err;
4.   console.log(data);
5. });
```

Networks, Internet and World Wide Web

Many types of networks exist and are used depending on needs and situations. Let's see the basic difference between them.

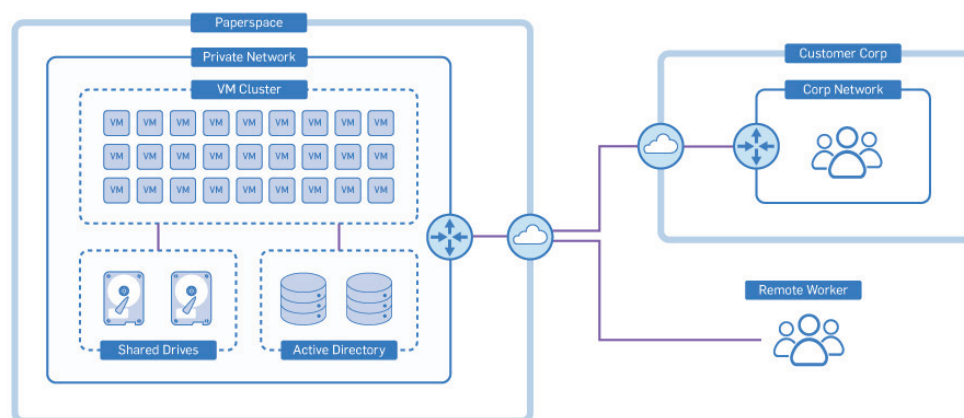
Internet vs. World Wide Web

The Internet is a massive interconnection system of devices made of millions of public and private networks. The World Wide Web, on the other hand, is an array of text documents and web resources created using various programming languages. Being a tool to access the Internet (among others) using HTTP and TCP/IP protocols, the WWW is one of the ways (the most common way) to exchange data over the Internet.



Private Network

As opposed to **public networks** to which anyone can access with basically no restrictions, a **private network** consists of a system of various number of devices connected in an environment that promotes security. It is configured in such a way that only specific devices can access it, depending on certain criteria. It's normally encoded on its networks' *routers* and *access points*.

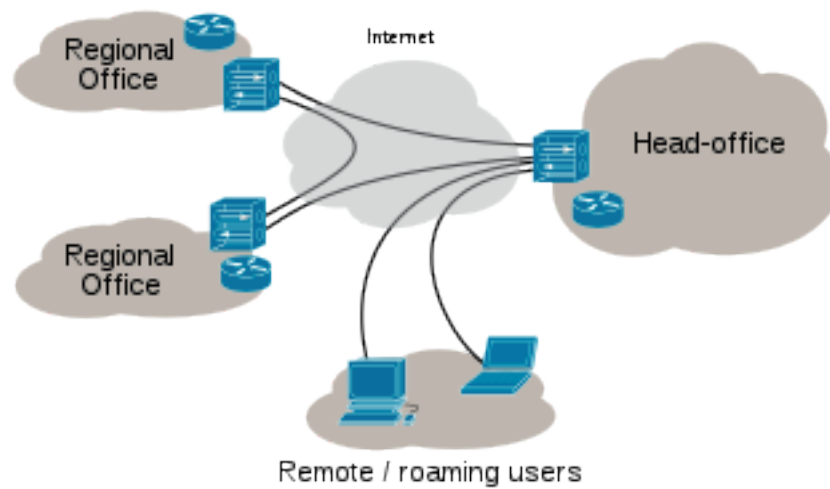


Private Networking

Virtual Private Network – VPN

As the name suggests, a virtual private network is just like a private network, but virtual. This means that instead of a user accessing the Internet directly, the access is made through a server (a virtual private network) and the server acts in the user's name.

VPN are more and more popular as they increase security. Using VPN, your actual IP address isn't visible anymore as it is replaced by the VPN's one. So if, for instance, you are from Montreal, you could appear as being from London.



Local Area Network – LAN

Computer network covering a relatively small area and number of devices. For instance a company may create a network allowing employees to communicate with each other and to access and share information without needing to send data around the world through the World Wide Web. A LAN is also more secure as it is (usually) not connected to external networks or, if it does, it is secured behind a firewall.

Assignment 8: Debugging

With the program supplied by the teacher, use the debugging techniques covered in this class to identify and correct the bug(s) found.