TRABAJO PRÁCTICO DE INTEGRACIÓN N º 1

howstuffworks

Does adding more RAM to your computer make it faster?

Inside this Article

One topic you might hear people discussing when they're talking shop about computers is how much random access memory (RAM) they need to add to their computer. Up to a point, adding RAM will normally cause your computer to seem faster on certain types of operations. RAM is important ***because*** it eliminates the need to "swap" programs in and out.

When you run a program ***such as*** a word processor *or* an Internet browser, the microprocessor in your computer pulls the **executable file** *(.exe)* off the hard disk and loads it into RAM. Large programs ***like*** Microsoft Word or Excel use large amounts of memory. The microprocessor ***also*** pulls in a number of shared **dynamic link libraries (DLLs)** -- shared pieces of code used by multiple applications. The DLLs take many more megabytes.

Then the microprocessor loads in the data files at which you want to look, which might total several megabytes ***if*** you are looking at more than one document or browsing a page with a lot of graphics. ***So*** a big application can easily take 100 megabytes of RAM or more, which can slow your system down significantly ***if*** there isn't enough memory. On your machine, at any given time you might have the following applications running:

- A word processor
- A spreadsheet
- An e-mail program
- A drawing program
- Three or four browser windows
- A fax program
- A Telnet session

Besides all of those applications, the operating system itself is taking up a good bit of space. Everything together may need more RAM than your machine has. Where does all the extra RAM space come from?

Next Page

Tips for Adding RAM

Kim Jae-Hwan/AFP/Getty Images

A microchip show owner in Seoul's electronics shopping mall checks the display of dynamic random access memory (DRAM) chips. Adding more RAM will make your programs run smoother, but up to a point any additional RAM has no effect.

The extra space in your computer's RAM is created by an important operating system component called the **virtual memory manager (VMM)**. The VMM looks at RAM and finds sections that aren't currently needed. It puts these sections of RAM in a place called the **swap file** on the hard disk. ***For example***, let's say you have your e-mail program open, ***even though*** you haven't looked at e-mail in the last 45 minutes. The VMM moves all of the bytes making up the e-mail program's .exe, DLLs and data out to the hard disk. That's called **swapping out** the program. ***The next time*** you click on the e-mail program, the VMM ***will*** **swap in** all of its bytes from the hard disk, ***and*** probably swap something else out in the process. ***Because*** the hard disk is relatively slow compared to RAM, the act of swapping things in and out causes a noticeable delay.

So should you just keep adding more RAM until your pockets are empty or your computer can't hold any more? ***If*** you have a very small amount of RAM (say, 256 megabytes), ***then*** the VMM is always swapping things in and out to get anything done. ***In that case***, your computer feels like it is crawling. ***As*** you add more RAM, you get to a point where you only notice the swapping when you load a new program or change windows. Once your computer has more RAM than the software running on the machine uses, the VMM has plenty of room and you should never see it swapping anything. After that, adding more memory would have no effect.

Some applications—***like*** Photoshop, many compilers, most film editing and animation packages—need large amounts of RAM to do their job. If you run them on a machine with too little RAM, they swap constantly and run very slowly. You can get a huge speed boost by adding enough RAM to eliminate the swapping. Programs ***like*** these may run 10 to 50 times faster once they have enough RAM.

For lots more information on RAM and computer technology, swap over to the next page.

Previous Page
Next Page

RELATED CONTENT

Prices: RAM

-

  Crucial Technology 4GB kit MB DDR2 RAM (BL2KIT25664AA80E)

  DDR2 RAM

  $65.00 - $75.00

-

  Crucial Technology 2GB, Ballistix 240-pin DIMM, PC2-6400 2 GB DDR2 RAM (BL25664AA80E)

  2 GB, DDR2 RAM

  $30.00 - $38.00

-

  Kingston 2 GB DDR3 SDRAM (KTA-MP1066K4/8G)

2 x 1 GB, ECC Unbuffered DDR3 SDRAM, 1066 MHz, DIMM 240-pin, 2 Modules

$115.00 - $136.00

http://computer.howstuffworks.com/question1751.htm

Guía de Lectura:

Parte I:

Antes de leer:

1- Recorra el para-texto indicando los elementos que reconoce.
2- Identifique tipo de texto y fuente de la que fue extraído.
3- ¿Cómo está organizado?
4- Refiérase a las secciones que componen el texto y elabore una predicción acerca del tópico del mismo.

Mientras lee:

1- Lea globalmente el texto dejando de lado aquellas palabras que no conoce e identificando palabras transparentes, conocidas, etc… (skimming)
2- Lea el texto con mayor detenimiento (scanning) y resuelva las siguientes consignas:

a. Identifique enumeraciones. ¿Qué recursos lingüísticos se utilizan para señalarlas? ¿A qué refieren?
b. Según el artículo, ¿cuál es el tema de conversación más común entre los usuarios de computadoras?
c. ¿Qué programas o aplicaciones consumen más espacio de memoria RAM?
d. La primera sección del artículo concluye con un interrogante. Identifíquelo, interprételo y cite la respuesta.
e. ¿Qué función cumple el 'virtual memory manager?
f. ¿A qué se denomina *swapping*?
g. ¿Cuándo es *recomendable* el agregado de memoria?
h. ¿A qué refiere la indicación al final de la segunda sección del artículo? (For…)

Después de leer:

1- Luego de haber leído el texto, ¿cuál sería la respuesta al interrogante planteado al inicio del artículo?

Parte II

Relea el texto y resuelva las siguientes consignas:

1- Las siguientes frases nominales aparecen enumeradas en el texto. ¿A qué refieren? Identifique el núcleo e interprételas:

- A Word processor:
- A spreadsheet:
- An e-mail program:
- A drawing program:
- Three or Four browser Windows:
- A fax program:
- A Telmet session:

2- Considere la siguiente afirmación: 'RAM is important because it eliminates the need to "swap" programs in and out. Podemos decir que la expresión 'swap' es un verbo cuya interpretación podría ser 'intercambiar', …

Interprete otras variantes lingüísticas de la expresión 'swap' en las siguientes afirmaciones y complete el cuadro:

- 'It puts these sections of RAM in a place called the _swap_ file on the hard disk'.
- 'That's called _swapping_ out the program'.
- '…the VMM will _swap_ in all of its bytes from the hard disk'.
- '…the VMM is always _swapping_ things in and out to get anything done'.

EXPRESIÓN	FUNCIÓN/CATEGORÍA	INTERPRETACIÓN
Swap		
swapping		
(will) swap		
(is) wapping		

3- Indique la función de los ***conectores subrayados*** y explique su uso particular en este contexto:

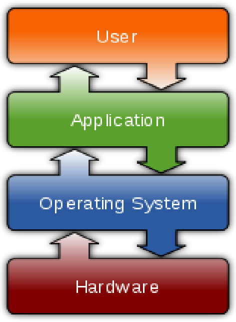
Because	***For example***
Such as	***Even though***
Like	***The next time***
Also	***In that case***
Then	***As***
If	
So	

4- Especifique cuál es el tiempo verbal predominante en el artículo y cite al menos un ejemplo de oraciones afirmativas, negativas e interrogativas.

TEXTOS 8 y 9:

TEXTO 8

1: OPERATING SYSTEM



COMMON FEATURES

- Process management
- Interrupts
- Memory management
- File system
- Device drivers
- Networking (TCP/IP, UDP)
- Security (Process/Memory protection)
- I/O

http://en.wikipedia.org/wiki/Operating_system

2: OPERATING SYSTEM

Reading: Match the labels to the four layers of this diagram with the help of the diagram caption.

Glendinning, E. H. & McEwan, J. (2002), 'Oxford English for Information Technology', Unit 6: Operating Systems, p 37, Oxford University Press.

1 applications programs
2 user
3 hardware
4 operating system

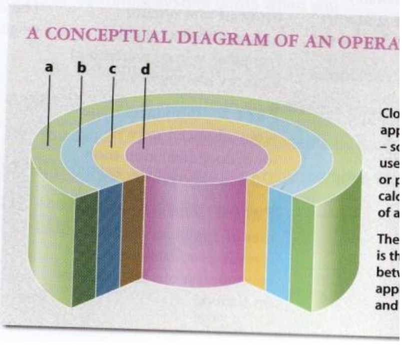


Fig 2
Conceptual diagram of an operating system

Antes de leer:

1. Observe los gráficos anteriores (Texto 8) detenidamente, lea el/los texto/s que los acompañan y, relacionando la información obtenida, defina qué es un sistema operativo.
2. Recorra el texto "Operating Systems: Linux" (Texto 9) e identifique: fuente original del texto y fuente de la que fue extraído.
3. ¿A qué público cree que está dirigida la publicación?

Mientras lee:

1. Lea el título y elabore una breve hipótesis acerca de su contenido.
2. Resuelva:
   a) ¿Qué es Linux? ¿Cuándo fue creado y por qué?
   b) ¿Por qué Linus decidió escribir su propio kernel?
   c) Defina 'source code' y refiérase a su importancia.
   d) ¿Qué diferencia existe entre el desarrollo de software 'open source' del software comercial?
   e) Defina el término 'distribution'.
   f) Indique a qué/quién refieren las siguientes siglas y nombres propios: GNU, Unix, Minix, KDE, Free Software Foundation, Richard Stallman, Andy Tannenbaum.
   g) ¿Cuál es la razón por la cual Linux es el SO más adoptado por los usuarios?
   h) Indique a que tiempos verbales refieren las siguientes expresiones: 'called' (line 2),'was studying' (line 3), 'was taught' (line 5), 'was' (line 6), 'cost' (line 12), 'handles, talks, makes, keeps (lines 18-20), 'developed' (line 21), ' 'd written' (lines 25-26), 'released' (line 26), 'don't have' (line 30), 'can't modify' (line 31), 'won't sell, will…do', will destroy' (lines 32-35), 'didn't do' (line43), ' 's called' (line 48), 'was designed' (lines57 & 59)
   i) ¿Qué tiempo verbal predomina?
   j) Lea la sección 'Voz Pasiva' y señale cuáles de las expresiones verbales anteriores corresponden a esta clasificación.
   k) ¿Qué indica el uso de 'if'?

l) Lea las siguientes frases u oraciones extraídas del texto e indique si ejemplifican situaciones probables o improbables. Luego subraye las expresiones verbales/verbos modales y señale a qué tiempo (presente/pasado) pertenecen:

- 'If you don't have the source code to a program, you can't modify it to fix bugs or add new features, ...' (lines 29-31)
-'…if they make it available it will destroy their revenue stream.' (lines 34-36)
-'If you use it heavily you may want to extend or develop or fix bugs in it.' (line 53)

Después de leer:

1. Elabore y redacte dos o tres conceptos que a su criterio serían claves para la realización de un resumen del texto leído.

2. Redacte un breve resumen del texto incorporando los conceptos claves mencionados.

Note: The following text has been adapted from: 'Smooth Operator' by Charles Stross, Computer Shopper Magazine, November 1998 in Glendinning and McEwan (2002), Oxford English for Information Technology, Unit 6, Operatin Systems, pp. 42-43, Oxford University Press.

TEXTO 9

LINUX

Linux has its roots in a student project. In 1992, an undergraduate called Linus Torvalds was studying computer science in Helsinki, Finland. Like most computer science courses, a big component of it was taught on (and about) Unix. Unix was the wonder operating system of the 1970s and 1980s: both a textbook example of the principles of operating system design, and sufficiently robust to be the standard OS in engineering and scientific computing. But Unix was a commercial product (licensed by AT&T to a number of resellers), and cost more than a student could pay.

Annoyed by the shortcomings of Minix (a compact Unix clone written as a teaching aid by Professor Andy Tannenbaum) Linus set out to write his own 'kernel' – the core of an operating system that handles memory allocation, talks to hardware devices, and makes sure everything keeps running. He used the GNU programming tools developed by Richard Stallman's Free Software Foundation, an organisation of volunteers dedicated to fulfilling Stallman's ideal of making good software that anyone could use without paying. When he'd written a basic kernel, he released the source code to the Linux kernel on the Internet.

Source code is important. It's the original from which compiled programs are generated. If you don't have the source code to a program, you can't modify it to fix bugs or add new features. Most software companies won't sell you their source code, or will only do so for an eye-watering price, because they believe that if they make it available it will destroy their revenue stream.

What happened next was astounding, from the conventional, commercial software industry point of view – and utterly predictable to anyone who knew about the Free Software Foundation. Programmers (mostly academics and students) began using Linux. They found that it didn't do things they wanted it to do – so they fixed it. And where they improved it, they sent the improvements to Linus, who rolled them into the kernel. And Linux began to grow.

There's a term for this model of software development; it's called Open Source (see www.opensource.org/ for more information). Anyone can have the source code – it's free (in the sense of free speech, not free beer). Anyone can contribute to it. If you use it heavily you may want to extend or develop or fix bugs in it – and it is so easy to give your fixes back to the community that most people do so.

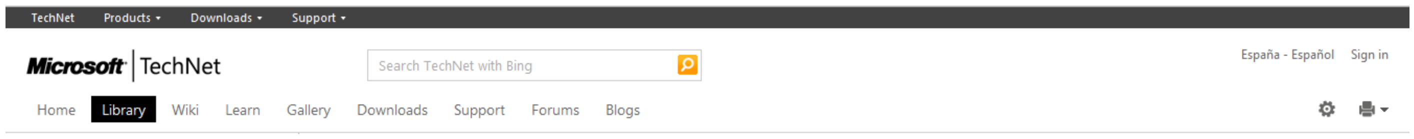
An operating system kernel on its own isn't a lot of use; but Linux was purposefully designed as a near-clone of Unix, and there is a lot of software out there that is free and was designed to compile on Linux. By about 1992, the first 'distributions' appeared.

A distribution is the Linux-user term for a complete operating system kit, complete with the utilities and applications you need to make it do useful things – command interpreters, programming tools, text editors, typesetting tools, and graphical user interfaces based on the X windowing system. X is a standard in academic and scientific computing, but not hitherto common on PCs; it's a complex distributed windowing system on which people implement graphical interfaces like KDE and Gnome.

As more and more people got to know about Linux, some of them began to port the Linux kernel to run on non-standard computers. Because it's free, Linux is now the most widely-ported operating system there is.

TRABAJO PRÁCTICO DE INTEGRACIÓN Nº2.

Trabaje con un compañero y desarrolle las consignas enumeradas a continuación del texto.



Defining Malware: FAQ
42 out of 65 rated this helpful Rate this topic
Published: October 1, 2003
By Robert Moir, Security MVP

Abstract
This document is a compilation of Frequently Asked Questions (FAQ) regarding “malware,” a general term coined for all forms of malicious software.

Questions about Malware

Q: What are malware, viruses, spyware, and cookies, and what differentiates them?

A: Let us take the easy one first. “Malware” is short for malicious software and is typically used as a catch-all term to refer to any software designed to cause damage to a single computer, server, or computer network, whether it’s a virus, spyware, et al.

Q. What exactly is a virus? Is a “worm” also a virus?

Viruses are computer programs or scripts that attempt to spread from one file to another on a single computer and/or from one computer to another, using a variety of methods, without the knowledge and consent of the computer user. A worm is a specific type of virus that propagates itself across many computers, usually by creating copies of itself in each computer’s memory.

Many users define viruses simply as trick programs designed to delete or move hard drive data, which, strictly speaking, is not correct. From a technical viewpoint, what makes a virus a virus is that it spreads itself. The damage it does is often incidental when making a diagnosis.

Obviously, any incidental damage is important, even when authors do not intend to create problems with their viruses; they can still cause harm unintentionally because the author did not anticipate the full effect or unintentional side effects. The most common method used for spreading a virus is through e-mail attachment. Sending a virus, even if designed to be harmless, can cause unforeseen damage.

Q. How can I prevent a virus from infecting my computer?

A virus scanner is the most common tool for prevention. This utility attempts to scan a computer program before it runs, and if it recognizes the signature of a malicious code, it shuts it down. Many scanners also evaluate programs to determine if it contains any virus-related characteristics.

The best way to stop viruses is to use common sense. If an executable computer program is attached to your e-mail and you are unsure of the source, then it should be deleted immediately. Do not download any applications or executable files from unknown sources, and be careful when trading files with other users.

Q. What is a “Trojan Horse”? Isn’t this a virus by any other name?

I have heard some arguments that Trojan Horse malware is a virus subset (and vice versa) but there are differences worth mentioning.

A Trojan Horse meets the definition of virus that most people use, in the sense that it attempts to infiltrate a computer without the user’s knowledge or consent. A Trojan Horse, similar to its Greek mythological counterpart, often presents itself as one form while it is actually another. A recent example of malware acting as a Trojan horse is the recent e-mail version of the “Swen” virus, which falsely claimed to be a Microsoft update application.

Trojans typically do one of two things: they either destroy or modify data the moment they launch, such as erase a hard drive, or they attempt to ferret out and steal passwords, credit card numbers, and other such confidential information.

Trojan Horses can be a bigger problem than other types of viruses as they are designed to be destructive or disruptive, as opposed to viruses and worms where the coder may not intend to do any harm at all. Essentially this distinction does not matter in the real world. You can lump viruses, Trojans, and worms together as “things I don’t want on my computer or my network”.

Q. How do I prevent a Trojan Horse attack?

The methods for dealing with Trojans are generally the same as for those for dealing with viruses. Most virus scanners attempt to deal with some of the common Trojans with varying degrees of success. There are also specific “anti-Trojan” scanners available, and your best weapon is common sense yet again. Score another point for safe computing!

Q. What are cookies and spyware? How are they different?

A cookie is just a bit of text in a file on your computer, containing a small amount of information that identifies you to a particular website, and whatever information that site wanted to retain about the user when they are visiting.

Cookies are a legitimate tool used by many websites to track visitor information. As an example, I might go to an online computer store and place an item in the basket, but decide not to buy it right away because I want to compare prices. The store can choose to put the information about what products I put into my basket in a cookie stored on my computer. This is an example of a good use of cookies to help the user experience.

The only websites that are supposed to be able to retrieve the information stored in a cookie are the websites that wrote the information in that particular cookie. This should ensure your privacy by stopping anyone other than the site you are visiting from being able to read any cookies left by that site.

Q. Do some websites use cookies to exploit user information?

A. Unfortunately, yes. Some may deceive users or omit their policies. For example, they may track your Web surfing habits across many different websites without informing you, and then use this data to customize the advertisements you see on websites, etc., typically considered as an invasion of privacy.

It is difficult to identify this and other forms of “cookie abuse,” which makes it difficult to decide if, when, and how to block them from ones system. In addition, the acceptable level of shared information varies between users, so it is difficult to create an “anti-cookie” program to meet the needs of everyone.

Q. How does spyware exploit user information?

The spyware problem is similar to the cookie problem from the point of view that both are an invasion of privacy, although spyware is different from cookies, technically speaking. Spyware is a program that runs on your computer and, again, tracks your habits and tailors these patterns for advertisements, etc. Because it is a computer program rather than just a bit of text in a cookie, spyware can also do some nasty things to ensure that the spyware keeps running and keeps influencing what you see.

Q. How do I know if spyware is running on my computer?

You can use detection programs such as Ad Aware and others. Similar to antivirus software, these programs compare a list of known spyware with files on your computer and can remove any that it detects. But again, what some consider unacceptable is perfectly acceptable to others.

Q. How does spyware install itself on computers?

Common tactics for surreptitious installation include rolling up advertising programs into “free” shareware program downloads, and once the spyware is installed it can download advertisements 24 hours a day and overlay them on websites and programs you are using. Anti-spyware programs can combat spyware from being installed, but the best strategy is to discriminate what you choose to download and install.

Q. Can spyware send tracked information to other people?

Some forms of spyware monitor a target’s Web use or even general computer use and sends this information back to the spyware program’s authors for use as they see fit. To fight this kind of problem, a spyware removal tool is obviously helpful, as is a firewall that monitors outgoing connections from your computer. Other forms of spyware take over parts of your Web browsing interface, forcing you to use their own search engines, where they can track your browsing habits and send pop-up advertisements to you at will.

The biggest concern regarding spyware is that most of them are poorly written or designed. Many people first realize their computer is running spyware when it noticeably slows down or stops responding, especially when doing certain tasks such as browsing websites or retrieving e-mail. In addition, poorly written spyware can often cause your computer to function incorrectly even *after* it has been removed.

Q. Do you have a quick summary of how to prevent malware problems?

A: Yes — see below.

Two of the biggest concerns for computer users today are viruses and spyware. In both cases, we have seen that while these can be a problem, you can defend yourself against them easily enough with just a little bit of planning:

- Keep your computer's software patched and current. Both your operating system and your anti- virus application must be updated on a regular basis.
- Only download updates from reputable sources. For Windows operating systems, always go to http://update.microsoft.com/microsoftupdate/ and for other software always use the legitimate websites of the company or person who produces it.
- Always think before you install something, weigh the risks and benefits, and be aware of the fine print. Does the lengthy license agreement that you don't want to read conceal a warning that you are about to install spyware?
- Install and use a firewall. If you are running Windows XP you can use the built-in software firewall under Control Panel, and there are free versions of firewalls that work on all versions of Windows.
- Prevention is always better than cure.

From: http://technet.microsoft.com/en-us/library/dd632948.aspx#feedback

Consignas de trabajo.

Parte I

1- Recorra el texto e indique:
-Tipo de texto:
-Fuente:
-Sección a la que pertenece:
-Autor:
-Título:
-Otros datos de publicación que considere relevantes:

2- Lea el 'abstract' y determine el tema del texto.
3- ¿Qué función cumple el 'abstract' en este tipo de publicaciones?
4- Lea las preguntas enunciadas en el texto y confirme, rectifique o amplíe el tema del texto.
5- ¿A qué refieren las siguientes siglas y abreviaturas: FAQ – MVP – A: - Q. – et. Al – etc.?
6- ¿Qué es 'virus'?
7- ¿Se puede utilizar el término 'worm' como sinónimo de 'virus'?
8- ¿Cómo puede prevenirse una infección virósica?
9- ¿Qué es un troyano o 'Troyan Horse' y cómo puede prevenirse su ataque?
10- El autor compara esta denominación con su "…Greek mythological counterpart…" ¿A qué se refiere o por qué recurre a esta comparación?
11- Defina "cookies" y "spyware": realice un cuadro, diagrama o esquema señalando diferencias y similitudes, cómo utilizan datos de los usuarios de computadoras, etc.
12- Deténgase en las preguntas referidas específicamente a "spyware" y resuma cómo puede detectarse su presencia en las computadoras, de qué modo se instala y por qué es preocupante su presencia.
13- Describa brevemente las pautas sugeridas por el autor para prevenir posibles problemas relacionados con el "malware".

Parte II

1- El autor define "malware" como "malicious software", lo que podríamos interpretar como software malicioso, maligno, dañino, etc. Siguiendo la misma lógica cómo se interpretarían los términos "spyware" y "shareware"?
2- Identifique y subraye en el texto dos ejemplos que señalen comparaciones, ejemplificaciones y condición. ¿Qué conectores le ayudaron a identificarlos? Explique brevemente las ideas que conectan.
3- Indique a qué hacen referencia las siguientes expresiones:
"This" (line 2)
"Them" (line 5)
"us" (line 6)
"it" (line 7)
"itself" (line 11)
"what" (line 13)
"their" (line 15)
"it" (line 21)
"you"- "your" (line 23)
"I" (line 27)
"which" (line 32)
"my" (line 37)
"I" (line 38)
"some" (line 53)
"that" (line 61)
"others" (line 65)

4- El uso de verbos modales es otro recurso lingüístico que destaca una variedad de significados acompañando a un verbo principal (ver anexo teórico). En el texto predomina el uso de **"can".** Justifique esta elección.
5- Finalmente, incluya las expresiones "malware, virus, worm, trojan horse, cookies, spyware" y realice un resumen indicando los distintos tipos de "malware" y cómo pueden prevenirse.

Texto 10

COMPUTER PROGRAMMING

HOME WHY LEARN? REQUIREMENTS LANGUAGES JOBS CONTACT US

LANGUAGES

Selecting a ***programming*** language depends on the task at hand. It is possible to accomplish most programming tasks with most programming languages. Some programming languages are better suited for general file manipulations, whilst others are more suitable for text ***filtering*** and ***processing***, and others are better at systems tasks or mathematical operations. Every programming language has its plus points. Good programmers will have a good command of a few languages, ***enabling*** them to select the language best suited to the particular programming challenge presented to them.

There are many programming languages and all vary in complexity. Below are a few *of* the most popular, along with advantages and disadvantages of each:

C

C is a structured programming language created in the 1970's. It was designed to be small and simple, suited *towards* systems programming and operating system design.

Advantages: Excellent for writing small fast programs. Easily interfaced with assembly language.

Disadvantages: Doesn't easily lends itself towards object-oriented programming design. Learning the syntax can be tricky.

C++

C++ is an object-oriented successor to C. Object-oriented programs are the next logical step after structured programming. Object-orientated programs are built out of objects. Objects are small packages of reusable functions and code. There are many publicly and commercially available libraries of objects available that should make programming a lot simpler. You just pull together the objects you need like building blocks - or at least that's the theory.

Advantages: Far better than traditional C for the organisation of large programs. Supports the object-oriented approach of program design very well indeed.

Disadvantages: A vast and complex language with difficult syntax. Not as quick as C in many applications.

Visual Basic

Based on the orignal BASIC programming language. Visual Basic, by Microsoft, allows for rapid development of GUI based programs and is considered to have good database support.

Advantages: Not difficult to learn compared to other programming languages. Almost Instant compiling makes for extremely fast and easy development. Lots of bolt-ons available.

Disadvantages: Applications developed tend to very large and need several large runtime DLL's in order to run. Whilst easy to develop simple window based programs, it is not easy to write complex graphical application.

Java

Java was designed by Sun Microsystems and was designed to be a portable incarnation of C++" and was designed with embedded applications in mind.

Advantages: The binaries produced are portable to other platforms. Applications can be run within web pages. Very robust - virtually no resource leeks.

Disadvantages: A virtual machine is used to run the Java programs which makes Java applications slower than true compiled applications. As it's a high level language, low level system interaction is very difficult.

http://www.allan-home.co.uk/languages.html

Consignas de lectura

1) Recorra el texto e indique: fuente y tipo de texto.
2) Observe los elementos para-textuales y elabore una hipótesis del contenido del texto.
3) Lea la primera oración de cada párrafo y complete o rectifique su hipótesis.
4) Lea detenidamente el texto y desarrolle las siguientes consignas:

- ¿De qué depende la selección de un lenguaje de programación determinado?
- ¿Cómo describe el autor a un 'buen programador?
- El lenguaje C++ se plantea como el sucesor del lenguaje C. ¿En qué se relacionan y/o diferencian?
- ¿A qué se denomina 'object oriented language'?
- ¿Por qué se define al lenguaje Java como 'a portable incarnation of C++?
- Identifique estructuras comparativas. De ejemplos.
- ¿Qué función cumple el conector *indeed*? (line 20)
- Identifique el uso de expresiones negativas. Cite ejemplos.
- ¿Qué indica el conector *in order to*? (lines 27-28).
- Las siguientes oraciones extraídas del texto contienen palabras terminadas en ***–ing.*** Dichas palabras adquirirán una función determinada, es decir referirán a un objeto, acción o característica de un objeto, de acuerdo al lugar que ocupan en un enunciado. Lea los ejemplos citados y determine qué función cumplen las palabras resaltadas en negrita:
  -***Selecting*** a ***programming*** language depends on the task at hand.
  -Some programming languages are better suited for general file manipulations, whilst others are more suitable for text ***filtering*** and ***processing***, and others are better at systems tasks or mathematical operations.
  -Good programmers will have a good command of a few languages, ***enabling*** them to select the language best suited to the particular programming challenge presented to them

5) Una vez finalizada la lectura, complete el siguiente cuadro:

PROGRAMA	UTILIDAD Y PRINCIPALES CARACTERÍSTICAS	VENTAJAS	DESVENTAJAS
C			
C++			
VISUAL BASIC			
JAVA			

Texto 11

About Python

Python is a remarkably powerful dynamic programming language that is used in a wide variety of application domains. Python is often compared to Tcl, Perl, Ruby, Scheme or Java. Some of its key distinguishing features include:

- very clear, readable syntax
- strong introspection capabilities
- intuitive object orientation
- natural expression of procedural code
- full modularity, supporting hierarchical packages
- exception-based error handling
- very high level dynamic data types
- extensive standard libraries and third party modules for virtually every task
- extensions and modules easily written in C, C++ (or Java for Jython, or .NET languages for IronPython)
- embeddable within applications as a scripting interface

Python is powerful... and fast

Fans of Python use the phrase “batteries included” to describe the standard library, which covers everything from asynchronous processing to zip files. The language itself is a flexible powerhouse that can handle practically any problem domain. Build your own web server in three lines of code. Build flexible data-driven code using Python's powerful and dynamic introspection capabilities and advanced language features such as meta-classes, duck typing anddecorators.

Python lets you write the code you need, quickly. And, thanks to a highly optimized byte compiler and support libraries, Python code runs more than fast enough for most applications. The traditional implementation of CPython uses a bytecode virtual machine; PyPy supports just-in-time (JIT) compilation to machine code. Also, Jython and IronPython (see below) support JIT compilation on their respective virtual machine implementations.

Python plays well with others

Python can integrate with COM, .NET, and CORBA objects.
For Java libraries, use Jython, an implementation of Python for the Java Virtual Machine.
For .NET, try IronPython , Microsoft's new implementation of Python for .NET, or Python for .NET.
Python is also supported for the Internet Communications Engine (ICE) and many other integration technologies.
If you find something that Python cannot do, or if you need the performance advantage of low-level code, you can write extension modules in C or C++, or wrap existing code with SWIG or Boost.Python. Wrapped modules appear to your program exactly like native Python code. That's language integration made easy. You can also go the opposite route and embed Python in your own application, providing your users with a language they'll enjoy using.

Python runs everywhere

Python is available for all major operating systems: Windows, Linux/Unix, OS/2, Mac, Amiga, among others. There are even versions that run on .NET, the Java virtual machine, and Nokia Series 60 cell phones. You'll be pleased to know that the same source code will run unchanged across all implementations.

Your favorite system isn't listed here? It may still support Python if there's a C compiler for it. Ask around on news:comp.lang.python - or just try compiling Python yourself.

Python is friendly... and easy to learn

The Python newsgroup is known as one of the friendliest around. The avid developer and user community maintains a wiki, hosts international and local conferences, runs development sprints, and contributes to online code repositories.

Python also comes with complete documentation, both integrated into the language and as separate web pages. Online tutorials target both the seasoned programmer and the newcomer. All are designed to make you productive quickly. The availability of first-rate books completes the learning package.

Python is Open

The Python implementation is under an open source license that makes it **freely usable and distributable, even for commercial use**. The Python licenseis administered by the Python Software Foundation.

Take a look at application domains where Python is used, or try the current download for yourself.

Website maintained by the Python community
hosting by xs4all / design by Tim Parkin

Consignas de trabajo:

Lea el texto y organice su análisis a partir de:
-las estrategias de lectura que considere convenientes con el propósito de anticipar y predecir el texto,
-en qué contexto fue producido (fuente, tipo de texto, etc.) y para quién,
-la estructura del texto teniendo en cuenta secciones, palabras claves, conectores, enumeradores, etc.
A partir de este análisis:
-extraiga la idea principal de cada párrafo y/o sección,
-refleje su comprensión del texto por medio de un resumen que integre los elementos relevados.

TRABAJO PRÁCTICO FINAL

Modalidad de Trabajo

-Seleccione un texto relacionado con un Lenguaje de Programación a su elección.
-Desarrolle su análisis aplicando las estrategias de lectura abordadas durante el año.
-Resuma el texto leído.

Presentación del trabajo:

-Una copia en limpio del texto original.
-Referencias bibliográficas.
-Resumen.
-Glosario Inglés-Español
-Formato: Hoja tamaño A4 o carta, márgenes normales, tipo de letra Arial 11, interlineado $1^{1/2.}$
-Exposición del tema abordado.