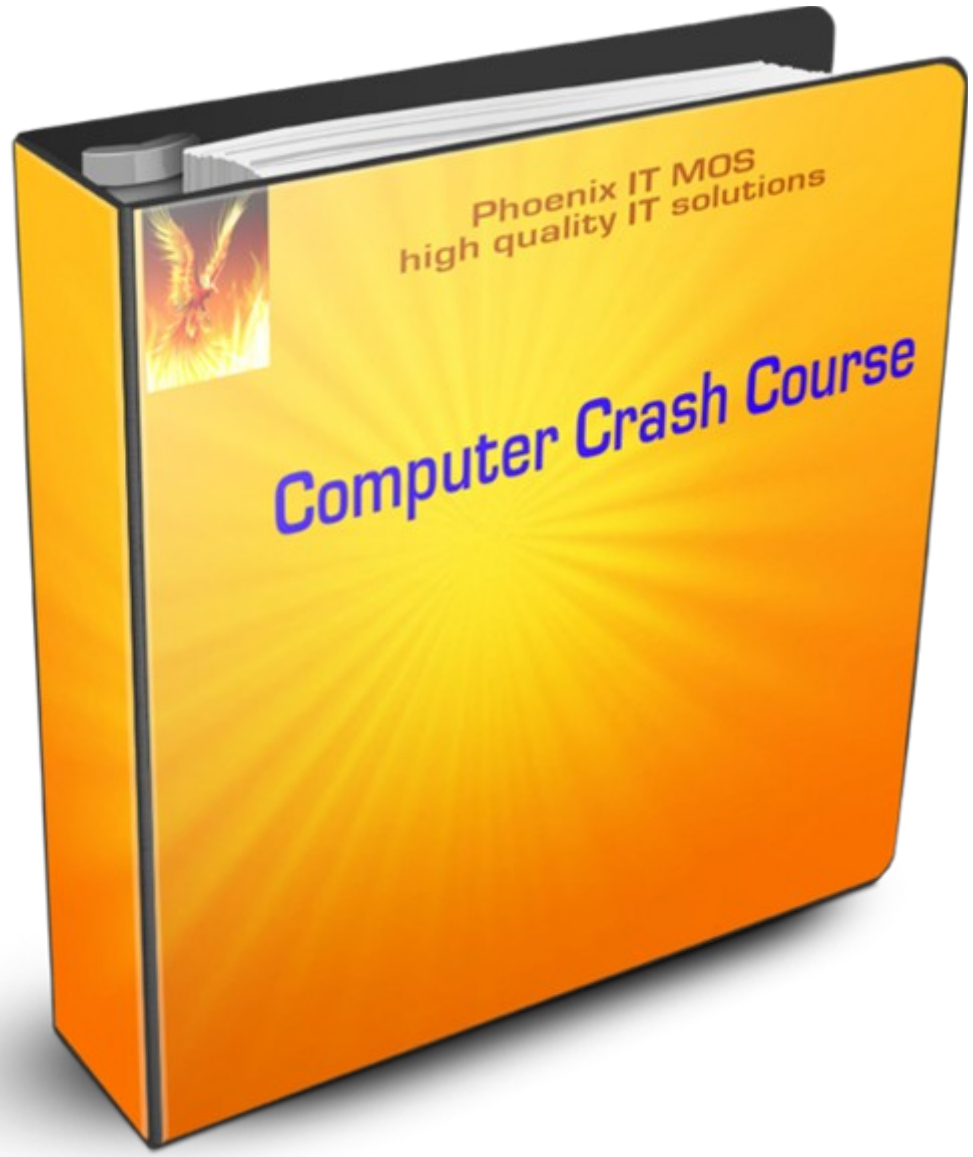

Phoenix IT MOS
high quality IT solutions
Computer Crash Course

Background of this course , what it is + how to use

I put this information together in order to have a supplement to the school program in computer science for my son.

This course is basically a guideline for teachers in basic computer knowledge.

It can also be used as an introduction for beginners. You have then to follow the links and read additional information concerning the subjects mentioned in this document, otherwise you won't understand this course.

Warning : this course is not a HOW TO DO, which tells you to use a specific software or click on this and that.

Prerequisites

None !

Further reading

Programming language basic elements

The Programming language basic elements course gives an general introduction to the basic elements of a high level programming language. I use ONLY pseudo code in this document !

Programming introduction

The Programming introduction is a very short "jump" into programming at a basic level. This document will give you an idea of writing a script and do programming in a high level programming language.

Python examples

The Python examples should help non programmers to understand some Python programming language basics. This document is a supplement to the Programming introduction document, it describes not the whole language.

Build a Website easily

The Build a Website easily document explains you which steps you can take to create a Website. This is an umbrella document for the following courses : HTML introduction, CSS introduction and SEO introduction.

Table of Contents

Hardware

Abstract

Hardware is all the "ware" you can touch, this means all pieces of the computer you can see.
For example the screen, the mouse, the keyboard, the interfaces and so on.

Software on the other hand is all the "ware" you cant't touch, Software signifies all the programs which can run on the computer, for example the operating system (ex Windows, Linux...), a printer driver, a text editor (ex Word, Open Office) and so on.

A simplified computer system includes a :

CPU (Central Processing Unit, or the boss of your computer)
The CPU contains the :
ALU, control unit, instruction decoder, registers, program counter, CPU-bus.
Memory (to store data and programs temporarily)
Interfaces (connects the bus to I/O devices, hard disk, etc)
Bus (connects each element, like CPU, memory, interfaces)
I/O devices (screen, mouse, keyboard...)
Hard disk (stores permanently programs, data, operating system...)
Power supply (transforms the alternative current to direct current - AC to DC)

Some details about :

- ALU

The Arithmetic and Logic Unit is a digital circuit used to perform arithmetic and logic operations. It represents the fundamental building block of the central processing unit (CPU)

- Control unit

The CU can be designed in two ways : hardwired or microprogram based.
The CU does the following :

- Controls sequential instruction execution
- Interprets instructions
- Guides data flow through different computer areas
- Regulates and controls processor timing
- Sends and receives control signals from other computer devices
- Handles multiple tasks, such as fetching, decoding, execution handling and storing results

- Registers

  Registers are the most important components of CPU. Each register performs a specific function. They are used to quickly accept, store, and transfer data and instructions that are being used immediately by the CPU. There are various types of registers those are used for various purpose. Some mostly used registers are Accumulator(AC), Data Register(DR), Address Register(AR), Program Counter(PC), Memory Data Register (MDR), Index Register(IR), Memory Buffer Register(MBR).

- Bus

  - Data bus : is used to transfer data within CPU, memory and input/output devices. It is bidirectional.
  - Address bus : It is a group of wires that are used to address a location of the memory or to address I/O devices. It is unidirectional.
  - Control bus : CPUs uses the control bus to process data. Some control signals are read, write and opcode fetch, etc. Various operations are performed by the CPU with the help of the control bus.

History

The first "real" computer was Konrad Zuses	Z3	1941
The first commercial computer was the	Univac 1	1951
The first **P**ersonal **C**omputer (PC) was the	Altair 8800	1975

Further reading *:* *Computer history on Wikipedia*

Mainframe

The mainframe is a huge machine for government and big companies, to compute large amounts of data.

Further reading : Mainframe computer on Wikipedia

Super Computer

Super Computer is a broad term for the fastest computers currently available. They are very expensive and are employed for specialized applications that require immense amounts of mathematical calculations. They are used for weather forecasting, nuclear energy research, analysis of geological data, scientific simulations, etc.

Further reading : Super Computer on Wikipedia

Workstation

A Workstation is more powerful than a PC and is connected to a local area network. It is used for professional engineering application like CAD and CAM.

Further reading : Workstation on Wikidepia

Minicomputer

A minicomputer was a computer of a size between a mainframe and a microcomputer. The mini was a centralized computer with multiuser capabilities, the user communicated via a terminal. Today the use of the term minicomputer has diminished and has merged with ***servers***.

Further reading : Minicomputer on Wikipedia

Personal Computer

A Personal Computer or PC is a **microcomputer** designed to be used by one person at a time.

Further reading : Personal Computer on Wikipedia

Binary System

All computer functions are based on the binary system. The numbers in the binary system are "0" and "1", that's it!
The representation of decimal 3 in binary is 011, decimal 7 is 111, decimal 5 is 101 and so on.

Further reading : Binary system on Wikipedia

Computer architectures

Von Neuman

This architecture was proposed by the mathematician John von Neumann (1945). It describes the design of an computer with its basic elements : the **CPU** (including the arithmetic logic unit, control unit, registers and program counter), the **memory** (for data and instructions), the **input/output** interface, the external mass **storage** and the **bus** (connecting the elements).

Instruction set related

- RISC (Reduced Instruction Set Computer) - Harvard Architecture

RISC is a CPU design strategy based on the insight that simplified instruction set gives higher performance. It is a type of microprocessor architecture that uses highly-optimized set of instructions. RISC does reducing the cycles per instruction at the cost of the number of instructions per program. Pipelining is one of the unique feature of RISC. It is performed by overlapping the execution of several instructions in a pipelined fashion. It has a high performance advantage over CISC.

Examples : Apple iPods, iPhone, some mobile phones

Characteristics : Simple instructions taking one cycles
hardwired instructions
few instructions
highly pipelined

- CISC *(Complex Instruction Set Computer)*

The CISC approach attempts to minimize the number of instructions per program, sacrificing the number of cycles per instruction. The number of instructions per program can be reduced by embedding the number of operations in a single instruction, thereby making the instructions more complex. CISC has the ability to execute addressing modes or multi-step operations within one instruction set.

Examples : IBM 370, VAX 11/780, PDP-11, the 68000 family, the x86 based processors

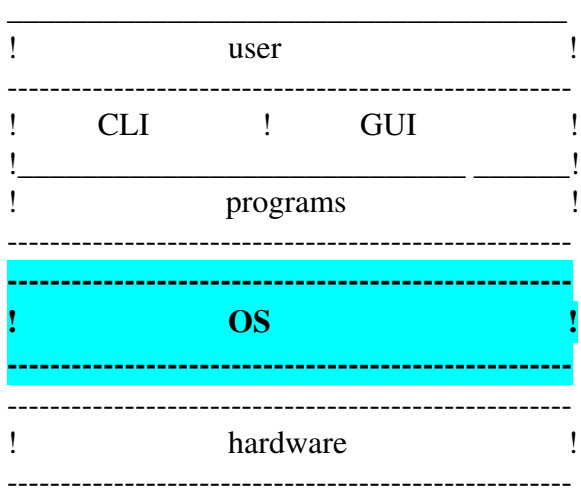
Characteristics : Complex instructions taking multiple cycles
instructions are executed by microprogram
many instruction
not, or less pipelined

Operating System

Abstract

The **O**perating **S**ystem (OS) is a piece of software which connects the hardware to the programs/user. The operating system translates the needs from the user/programs to the hardware and vice versa. For example : store a text on a hard disk, show a graphic on the screen, list a directory. The OS also manages, coordinates and shares all the computer hardware resources.

Place of the **OS** in a simplified computer system :



Further reading : Operating system on Wikipedia

History

Early computers had no operating system! The first (Batch) OS was developed in the 1960's.

Further reading : Operating system history on Wikipedia

Types

- Batch Processing (first OS for mainframes)

All jobs are in the queue. One job is executed after the other (1,2,3,4,5,6,7....). Job results are printed or saved in a file. There is no interaction with the user while executing a job.

- Time Sharing (second OS for mainframes)

Severals users behind their terminals have access to a single computer at the same time. Every user gets small slices of computer time at a rate which makes him believe that he is working with his own computer.

- Real Time

This OS is for time critical purposes, for example computers in cars.

- Single User / Single Tasking

One user can run one program, for example MS DOS for PC's.

- Multi User / Multi Tasking (multi programming)

Multiple users can run multiple programs, for example Unix.

Basic Elements

Bootstrap, for example the PC's BIOS or UEFI (firmware)
Boot loader
---------- Kernel ----------

Task/process management
Scheduler (timing)
Dispatcher (priority)

CPU management
Memory management, including DMA and virtual memory
I/O management (spooler...)
Interrupt handler (HW, SW)
Device drivers
File system (storage)
Network system
Application **P**rogramming **I**nterface (for system call services) API / library

---------- Kernel ----------

CLI(**C**ommand **L**ine **I**nterpreter) or shell -> programs, utilities, programming software
GUI (**G**raphical **U**ser **I**nterface) -> programs, utilities, programming software

Further reading : Operating System Kernel on Wikipedia

PC Boot Sequence

Reset or power on - BIOS (ROM) - boot loader (hard disk) - OS kernel (hard disk) - shell.
The PC is now ready to execute programs (for example the GUI) or to execute commands.

PC Installations

- Classic

You have to partition and format the disk, then you can install a single OS on the disk.

- Multi Boot

First you have to partition and format the disk for each OS, then you can install the operating systems on the disk. Second you have to install a multi boot loader. With the multi boot loader you can select one of the multiple OS to start.

- Emulation

You have to partition and format the disk, then you can install a single OS on the disk.
The emulator runs as a program on the installed OS. In the emulator you can run programs of an other OS. The emulator is "simulating" (emulating) an OS.
Example : under Linux-OS in the emulation "Wine" you can run Windows programs.

- Loop Mounting (loop mounted devices)

You have to partition and format the disk, then you can install a single OS on the disk. The second OS is installed in a subdirectory of the actual OS file system. An additional entry for the second OS is added to the original boot loader. With the modified boot loader you can select one or the other OS to start.
Example : under Windows in "Wubi" you can run Ubuntu-Linux.

Further reading : Wubi on sourceforge

- Virtual Machines

There are different types of virtual machines.

Hardware virtualization

This is a software implementation of a specific machine (hardware). This allows the execution of software just like the real machine would do.

Software virtualization

The virtual machine allows you to run a operating system (guest) inside of another operating system (host). For example with Parallels you can run Windows inside of OS X, or with VMWare you can run Linux inside of Windows.

Most commonly the virtual machine software, also known as a hypervisor or virtual machine monitor, connects the hardware to multiple operating systems. With that you can run different applications on different operating systems simultaneously.

Further reading : Virtual Machine on Wikipedia

Live Medias

- Live CD

A live CD is a CD with a bootable OS on it. With the live CD you can test an OS without the inconvenience of an installation on the hard disk.

Further reading : Live CD on Wikipedia

- Live USB stick

A live USB is a USB stick with a bootable OS on it. With the live USB stick you can test an OS without the inconvenience of an installation on the hard disk. The USB stick can function as a "computer on a stick". This means that users can store not only the operating system, but also applications and data files.

Further reading : Live USB on Wikipedia

Application Programs

Word Processor

Examples : Word for Windows / Libre Office Writer for Linux-OS's

Further reading : Word processor on Wikipedia

Spreadsheet

Examples : Excel for Windows / Libre Office Spreadsheet for Linux-OS's

Further reading : Spreadsheet on Wikipedia

Vector Graphic Editor

Examples : Paint for Windows / Libre Office Graphics for Linux-OS's

Further reading : Vector graphics on Wikipedia

Database

Examples : Oracle for Windows / Libre Office Database for Linux-OS's

Further reading : Database on Wikipedia

Graphics Software (bitmap)

Examples : Photoshop for Windows / Gimp for Linux-OS's

Further reading : Graphics software on Wikipedia

Media Player

Examples : VLC for Windows / Totem for Linux-OS's

Further reading : Media player on Wikipedia

Software Development

Abstract

Software development signifies the whole process to find a solution for a given problem in using appropriate methods and tools to create a software application.

Further reading : Software development on Wikipedia

The Software Development Live Cycle *(SDLC)*

- Analysis

Methods : HIPO
SADT
Data flow diagram

- Design

Methods : SADT
Jackson method
Nassi-Shneiderman diagram
Pseudo code

Top down, bottom up
Program flow
Programming paradigm : Structured programming, OOP, ...
Data structures
Algorithms

- Implementation (coding)

Use of modules, libraries, frameworks, etc.

- Test an debugging

Methods : Reviews
Walk-troughs
Dynamic testing
A/B testing

- Deployment

- **Maintenance**

Don't forget the documentation and the control cycle of each step!

Programming

Programming is the implementation (coding) of a problem/solution in a programming language.

Machine Language

Machine languages are the only languages understood by computers. Every CPU has its own unique machine language. The machine language or machine code is represented by 0's and 1's like that "001 011 111" (this is the "jump" code for the DEC PDP11)

Further reading : Machine language on Wikipedia

Assembly Language

The assembly language represents the next abstraction level for the machine language. This language is also for a specific CPU. A program written in an assembly language must be translated by the program called "assembler" into the machine language.
Example for an x86 assembly code : mov ah,9

Further reading : Assembly language on Wikipedia

High Level Language

- Abstract

The **H**igh **L**evel **L**anguage (HLL) is the next abstraction layer for the assembly language. The high level programming language allows to solve a problem quicker, because this language is close to the language of human beings. Knowing details of the internal structure of a particular CPU is not necessary. The HLL code has to be translated into machine code (executable code) by a Compiler and a Linker or an Interpreter.

Example : Print "Hello world" --> this will write *Hello world* on the screen.

The programming environment is a set of tools around a Compiler or Interpreter.
In the Compiler environment you will find : the Compiler, the Linker, Libraries, the Debugger.
In the Interpreter environment you will find : the Interpreter, Libraries, the Editor.

Further reading : High level programming language on Wikipedia

- History

In the 1950's the HLL's appeared namely Fortran (1954), Lisp (1958), Algol (1958) and Cobol (1959).
Most popular/used languages today are Java (1991), the C family (1972), Python (1989) and PHP (1994).

Further reading : History of programming languages on Wikipedia

Programming Tools

- Source Code Editor

The Source Code Editor is a text editor for writing HLL source code. Source Code Editors have built in functionalities to help the programmer to write the code more efficient and error free.

Further reading : SourceCode Editor on Wikipedia

- Compiler

The Compiler is a program which translates the source code (written in a HLL) into the object code.

Further reading : Compiler on Wikipedia

- Linker

The Linker is a program which translates the object code into the executable code. The Linker can link multiple (object code) files into one file and then translate this file into the executable code.

Further reading : Linker on Wikipedia

- Interpreter

The Interpreter is a program that translates the source code and runs the program at the same time. It converts one program statement (line) into machine code, executes it and then proceeds to the next statement.

Further reading : Interpreter on Wikipedia

- Debugger

The Debugger helps programmers to test and find errors (bugs) in their program and fix them.

Further reading : Debugger on Wikipedia

Internet

Abstract

The Internet is a worldwide collection of computer networks, cooperating with each other to exchange data using a common software standard. Through telephone wires and satellite links, Internet users can share information in a variety of forms.

A PC with a modem connected to the telephone line will give you the basic access to the Internet. The modem is communicating with your **I**nternet **S**ervice **P**rovider (ISP) . The ISP is a gate to the Internet. The browser software on your PC shows you the content of a chosen Website.

Data flow in both directions : browser <-> modem <-> telephone line <-> ISP <-> Website.

Further reading : Internet on Wikipedia

History

1958 The Internet began as ARPANET an U.S. department of defense project. The idea was to create a nationwide computer network that would continue to function, even if a large portion of it would be destroyed.

1969 The first test was launched with the four node network : UC Los Angeles (UCLA), Stanford Research Institute, UC Santa Barbara, University of Utah.

1972 First public demonstration of the ARPANET with 40 connected machines.

1991 CERN releases the "www" developed by Tim Berners-Lee.

1992 The U.S. government began to pull out of network management and handed it over to commercial entities which offered Internet access to general public for the first time.

Further reading : History of the Internet on Wikipedia

Protocols

Internet protocols are standards to specify how computers interact and exchange information.

Further reading : Protocols on Wikipedia
Protocols on dmoz

- TCP/IP

Means Transmission Control Protocol / Internet Protocol. This is the accepted and most widely used transport layer protocol.

Further reading : TCP/IP on Wikipedia

- HTTP

Means Hyper Text Transfer Protocol. This is an application layer protocol.

Further reading : HTTP on Wikipedia

- SMTP

Means Simple Mail Transfer Protocol. This is an application layer protocol.

Further reading : SMTP on Wikipedia

- FTP

Means File Transfer Protocol. This is an application layer protocol.

Further reading : FTP on Wikipedia

- NNTP

Means Network News Transfer Protocol. This is an application layer protocol.

Further reading : NNTP on Wikipedia

- Telnet

Telnet is a protocol for accessing remote computers. Through Telnet, an administrator or another user can access someone else's computer remotely. This is an application layer protocol.

Further reading : Telnet on Wikipedia

Layers

- Application layer	(AL)	DHCP / SMTP / FTP / IRC / HTTP / SSH / Telnet ...
- Transport layer	(TL)	TCP / UDP ...
- Internet layer	(IL)	IPv4 / IPv6 / IPsec ...
- Link layer	(LL)	PPP / tunnels / Media Access Control ...

Further reading : Layers on Wikipedia

Security

- Abstract

Internet security means the protection of the resources (session/account, data, programs...) of your machine from attacks by unknown users from the Internet.

Further reading : Security on Wikipedia

- Possible entry points for attacks

- Users which give sensitive information to untrusted sources !
- Mail attachments
- Open ports
- Browser + all Internet related programs (FTP, P2P, instant messaging...).
- Programs (for example on a CD) with a Virus, a Worm (or other malware)
- Infected Websites

- Protection

- Don't communicate sensitive information to untrusted sources !
- Be careful with mail attachments !
- Run a personal firewall in stealth mode
- Make regular updates of all software you have (OS + Internet related programs)
- Configure your Internet related programs very restrictive (browser...)
- Run a good "anti virus" program with automated updates
- New CD's have to be checked by an "anti virus", before you use them
- Make "often" a backup of your data

- Malware

Malware means malicious software, it refers to programs who damage and/or do unwanted actions on a computer.

Some examples of malware :

Virus	attachment to data or program files, copies itself + infects other computers
Worm	independent malware program, copies itself + infects other computers
Trojan	useful program/tool with integrated malware
Spyware	reports sensitive local data to a unknown Internet source
Rootkit	open port(s) for other malware entering via Internet

Further reading : Malware on Wikipedia

Website

- Abstract

A Website is a collection of web pages, hosted on a web server. A Website is accessible through the Internet by using a web browser.

Further reading : Website on Wikipedia

- Website creation steps (a proposition !)

Global tasks
- Define the goal
- Evaluate the costs
- Select a host and a domain name

Design
- General layout
- Number of web pages
- Text
- Images
- Logo
- Animated effects
- Search engine optimization (SEO)

Local implementation
- Create the subdirectories
- Implement the web pages
- Do a local test

On the host
- Create subdirectories
- Upload the files
- Activate the Website monitoring
- Search engine registration

On line test
- All functionalities "by hand" (layout, links, etc.)
- Validation check HTML, CSS, JavaScript, PHP, etc.
- Meta tag check
- Links check (internal + external links)
- Browser compatibility and mobile check
- SEO checks (speed, etc)

Operation / documentation / controlling
- Finishing the documentation
- Check the Website monitoring on a regular basis
- Check the search engine result page (SERP)
- Improve the Website on a regular basis (content, SEO, etc.)