Computer – It’s an Electronic Device.

Difference between Electrical & Electronics Device ---

Name of Electrical Devices

1. Fan
2. Tube Light
3. Mixie
4. Grinder
5. Motors (A/C & D/C)
6. TV

Name of few Electronic Devices

1. Computer
2. Calculator
3. Smart Watches
4. Smart TV
5. Smart Plugs
6. Smart Fans

Concentrating/Studying/Analyzing the electron flow in a conductor --- Electrical

Concentrating/Studying/Analyzing the electron flow in a semi-conductor --- Electronics

Computer

1. Input Device (Keyborad, mouse, magnetic pen, pen drive)
2. CPU – Central Processing Unit [Memory, ALU, CU]
3. Output Device [Monitor, Printer, Plotter, Projector]

Memory – It’s helping us to save all the data in digital format

Types of Memory

1. Optical Memory (Laser light is used for read/write operation) [CD/DVD/BRD]
2. Magnetic Memory (Electro magnetic signals are used for read/write operation) [HDD/Floppy/Zip Drive]
3. Solid State Drives (SSD) – Storage will be on Semi-conductor devices

Father of Computer – Charles Babbage – He invented a mechanical device called “Analytical Engine”

i/o device, cpu – Analytical engine

RAM, ROM

RAM – Random Access Memory (Read/Write Operation) – Costly, Volatile (Memory is depends on electrical power) – 2GB- 32GB , SSD Memory

ROM – Read Only Memory – BIOS, Chip

EPROM, EEPROM

EPROM – Erasable/Programmable ROM – UV Light for Erase & Write

EEPROM – Electronically Erasable & Programmable ROM –

Human – Computer

|  |  |  |
| --- | --- | --- |
|  | Human | computer |
| i/p | Food, Drinks | Electricity, Keyboard, Mouse |
| o/p | Mouth, ear, eyes, sense | Speaker, mic, webcam, finger print reader, touch screen |
| Processing | Brain | CPU |
| memory | Brain | Optical, magnetical & SSD |
| Hardware | Parts of our body | Mother Board, monitor, keyboard, cpu, processors |
| Software | Soul | System (OS, Compilers) & Application softwares |

Hardware Configuration

1. Processors
2. Memory
3. I/O Devices

Softwares

Types of Softwares

1. System Software – Operating System (Windows, Linux, Mac, Solaris), Compilers & Interpreters
2. Application Software - MS-Office, AutoCAD, Adobe, Maya, Unity

DOS – Disk Operator System – Microsoft – (Character User Interface – CUI) – commands

Unix – It’s highly secured & powerful, multi-user operating system. (CUI)

Chrome – ChromeBook – Web based Operating System

GUI – Windows

Unix – Linux /MacOS – Using C programming Lang

Solaris – Written using JAVA

6000 – 512mb RAM, Pentium ii processor, 64GB HDD, 3.5’ Floppy disk drive, CD drive, 15.6” CRT colour Monitor

Punched cards are still in use in hand-weaving ( A technology used to create cloths from silk threads)

|  |  |  |
| --- | --- | --- |
|  | DOS | Unix |
| Create a File | Copy con <filename.ext> |  |
| List all files & Folders | dir |  |
| Copy file content |  |  |
| Copy folder content | Copy | cp |
| Changing Date | date | date |
| Changing Time | time | time |
| Renaming a file/folder |  |  |
|  |  |  |

DOS – File System (FAT- File Allocation Table)

Boot Loader –

POST – Power On Self Test (Check I/O Devices – Load the Boot options) F8/F12/F2 / Break – Go into BIOS.

Enable Hardware level virtualization, hyper-V, boot order (hdd, cd/dvd, floppy disk drive, network, usb devices)

Unix – File System - / (root)

Dos – C:/ drive (Boot folder)

Git-SCM = Software configuration mgmt. tool. It’s linux/unix based

Cygwin – Software.

DOS – Command Prompt / Power Shell

Unix – Bash (Shell)/Terminal

<https://www.cygwin.com/>

**Unix**

Unix is a family of operating systems that derive from one built in the 1970s at Bell Labs by AT&T. The current owner of the Unix license is The Open Group, however, Novell, Inc currently holds the copyrights. Linux operating systems and OS X operating systems are based on Unix. One of Unix's standard features is the command-line shell that is used to interface with the operating system. In Linux, this shell is commonly Bash.

**Bash**

The Bourne Again Shell or Bash is based on the original Unix shell, which was called sh. Bash is compatible with sh and incorporates other features as well. Bash was created in 1989.

OS – Is capable of handling files & Folders

Pls download Git from the

Cygwin – Is a tool, using which we can download all the unix/& linux based tools and that can be executed from windows itself.

Bash – downloaded using Cygwin

# Commands

In this section, we'll be covering some basic Unix commands. Before we begin, we should cover the basic aliases for locations in your file system. A file system is generally comprised of a directory, or folder, that itself can contain files and directories in a tree structure. The top-level directory is known as the root directory and it is the folder that contains all of the other folders on the drive partition you are currently accessing. In Unix, your root directory is represented by the / character. For example, if you wished to change directory to the root directory from anywhere on the partition, you would issue the following command: cd /

That's all well and good, but what about the directory you're currently in? The . character represents the current directory. Furthermore, .. represents the parent directory, so if you wanted to copy everything from the current directory up to the directory immediately above it, you would issue the following command: cp -r . ..

In Unix, we also have a directory called the home directory. This directory is usually the one that our terminal starts in and it is where our personal files are generally stored. This directory is represented by the ~ character.

Now, let's talk about some Unix commands:

## Arguments and Flags

In Unix, commands can be modified in two ways.

1. Commands can have arguments. Arguments given to a command take the form of strings written after the command. ex. command arg1 arg2 arg3
2. Commands can have flags. Flags are special arguments given to a command. There are two kinds of flags in Unix, short-hand or character flags, a single character (or group of characters), prefixed by a single dash -c, and full flags, the full name of the flag, prefixed by a double dash --flag.

Arguments to your command usually represent variables or targets for your command, and flags usually represent options you wish to enable for your command.

Ex. The following two commands do the same thing:

cp -r hi bye

cp --recursive hi bye

The above commands copy the contents of the hi directory to the bye directory recursively, using the -r and --recursive flags. This means that the cp command has been told to copy the contents of a directory instead of it's default mode which copies a specific file.

hi and bye are arguments to the command.

## The Most Important Command

man - The manual command will print to the terminal the manual for using a particular command. If you are unsure what flags or arguments a command takes, you simply type man command. For example, if you wished to see the manual for the copy command, you would issue the command: man cp

## Directory Commands

* cd - The change directory command allows us to navigate to a different directory on the drive.
  + go to root directory: cd /
  + go to home directory: cd or cd ~
  + navigate one directory up: cd ..
  + navigate into the hi directory, which is inside the bye directory: cd ./bye/hi
  + change to the previous directory: cd -
* ls - The list directory command allows us to see the contents of a particular directory. When given no arguments, it lists the contents of the current directory. The -a flag allows you to see hidden items in the directory.
  + list the contents of the current directory: ls
  + list the contents of the hi directory: ls hi or ls ./hi
  + list the contents of the directory including the "hidden" contents: ls -a
* mkdir - The make directory command allows us to create a new directory. mkdir takes an argument representing the name of the directory you wish to create.
  + create a directory named hi: mkdir hi
* pwd - The print working directory command prints the full name of the directory you are currently working in. For example, if you were working in the home directory inside of the root directory the output of pwd might be /home.

## General Purpose Commands

* su - The substitute user command allows you to switch users. With no argument, this defaults to the root user, which has higher priveleges. This can be useful if you need to perform multiple commands with elevated priveleges but is generally considered to be bad practice in preference to sudo, for administrative logging purposes.
* sudo - the sudo command allows you to run a particular command as the root user.
* clear - the clear command usually prints a number of blank lines such that all previous commands are no longer on the screen. There is a shortcut for this command, ctrl-l
* echo - the echo command will print a string or the result of a command to the console.
* > - The > operator will redirect the output of a command to a file. The file will be created or overwritten if it already exists. ex. ls . > log.txt
* >> - The >> operator acts the same way as the > operator but appends output to the file instead of overwriting if it exists.
* grep - the grep command prints any lines in a file or files that match a given pattern. By default, grep interprets the pattern as a basic regular expression.
  + Print all lines in hello.txt that contain the word goodbye: grep goodbye hello.txt

## File Commands

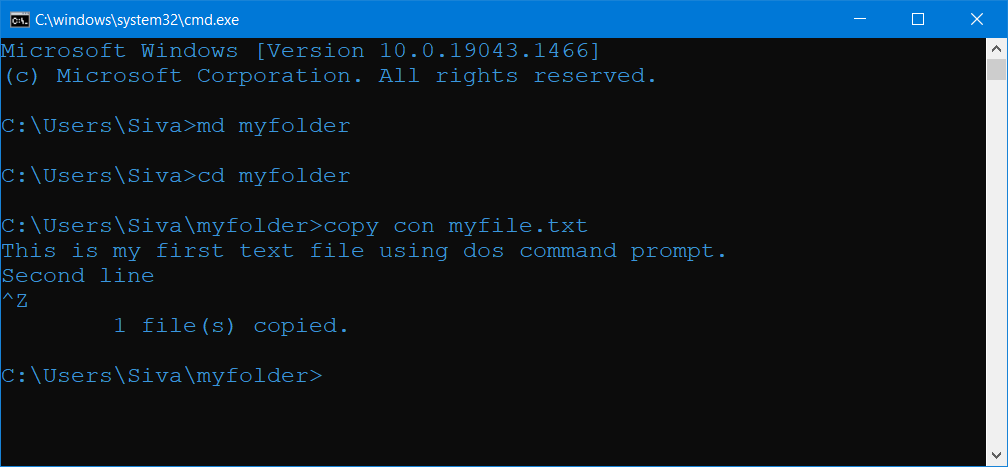
* cat - the concatenate command prints the contents of a file to the console. cat hello.txt
* head - the head command prints the first ten lines of a file to the console. head hello.txt
* tail - the tail command prints the last ten lines of a file to the console. tail hello.txt
* touch - the touch command allows you to modify the timestamp of a file. This command is usually used to create empty files, as an empty file is created if touch is given a file name that does not exist. touch hello.txt
* cp - the copy command creates a copy of the specified file at the location specified. If the recursive glag is used, it will operate on directories.
  + copy a hello.txt to goodbye.txt: cp hello.txt goodby.txt
  + copy the hello directory to the goodbye directory: cp -r hello goodbye
* mv - the move command will rename or move a file or entire directory with the recursive flag.
  + rename a hello.txt to goodbye.txt: mv hello.txt goodbye.txt
  + move hello.txt to the goodbye directory: mv hello.txt goodbye/.
  + rename the hello directory to goodbye: mv -r hello goodbye
* rm - the remove command will delete a file. If you use the recursive flag, it can delete a directory. The force flag will cause the command to delete files without prompting the user if there are warnings. The command rm -rf . is extremely dangerous.
  + remove hello.txt: rm hello.txt
  + remove the hello directory: rm -r hello
* wc - the word count command will print the number of words in a file. This command has several flags available
  + -c, --bytes - prints the byte count
  + -m, --chars - prints the character count
  + -l, --lines - prints the lines
  + -w, --words - prints the word count (default)
* ln - the link command creates a link between files. This allows you to make a shortcut to a file in one location without copying it over.

File – Flat file (filename.ext)

Copy con <filename.txt>

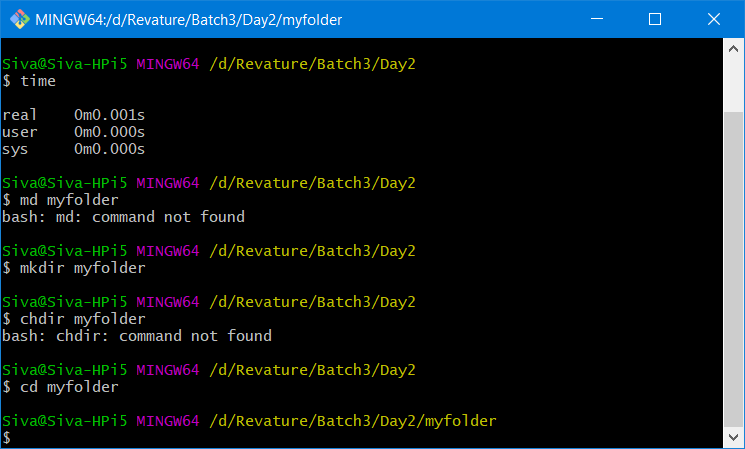
This is my first file using command prompt.

1. Open Command Prompt
2. Create a folder with the name myfolder by using the command “md myfolder”
3. Change the current folder as myfolder using the following command “cd myfolder”
4. Create a new text file using the command “copy con myfile.txt”
5. Type few words in that file and press ctrl+z



For Unix (Either open Cygwin bash or git bash)

* + 1. Create a folder using “mkdir myfolder”
    2. Change to that folder using “cd myfolder”
    3. To create a file, cat myfile.txt



Number System

1. Binary number system (0-1) base value 2
2. Octal number system (0-7) base value 8
3. Decimal number system (0-9) base value 10
4. Hexa decimal number system (0-9, A-F) base 16

SIVA –

ASCII value – A – 65

Simple Quiz

1. I’m giving you 1000 coins (1 cent)
2. I’m also giving you 10 boxes
3. You need to fill the 10 boxes using this 1000 coins
4. You can fill the box in any order.
5. I will ask particular value, I will ask you the whole amount (not fraction)
6. You have to give me the requested amount/quantity only (no extra/ no less value)
7. You can give one or more box
8. You should not take any coin from the boxes partially
9. How you will divide and store 1000 (1 cent ) coins in 10 boxes??

Answer

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 (2^0) | 2 (2^1) | 4 (2^2) | 8 | 16 | 32 | 64 | 128 | 256 | 489 |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |

1000 – 511 = 489

|  |  |  |  |
| --- | --- | --- | --- |
| Binary | Octal | Decimal | Hexa-decimal |
| 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 |
| 10 | 2 | 2 | 2 |
| 11 | 3 | 3 | 3 |
| 100 | 4 | 4 | 4 |
| 101 | 5 | 5 | 5 |
| 110 | 6 | 6 | 6 |
| 111 | 7 | 7 | 7 |
| 1000 | 10 | 8 | 8 |
| 1001 | 11 | 9 | 9 |
| 1010 | 12 | 10 | 10 |
| 1011 | 13 | 11 | A |
| 1100 | 14 | 12 | B |
| 1101 | 15 | 13 | C |
| 1110 | 16 | 14 | D |
| 1111 | 17 | 15 | E |
| 10000 | 18 | 16 | F |
| 10001 | 19 | 17 | 10 |

Memory address use hexa decimal number system –

O – off

1 – On

Switches – On/Off , gates

Toggle button -

Computer Programming Languages

* + 1. Low Level Language (Machine Language/ Binary Language/ Assembly level lang)
    2. Middle Level Language (C, BASIC, FORTRAN, FOXPRO)
    3. High Level Language (Java, Python, Ruby, Scala)

Computer can understand 0,1 only (Binary language)

Learn a new Language (French)

1. Learn the alphabets (vowels, consonants) how many alphabets
2. Learn words, phrase (pre-defined words)
3. Learn sentences using the learned words. (Grammer)
4. Read Poems, Novel, stories, watching to movies in French
5. Interact with your French speaking friends

A Reference Language (Learn French using English)

Steps to Learn any Programming Lang

1. Learn all characters, symbols, operators that we can use in that programming lang
2. Learn all keywords & it’s usage (pre-defined words)
3. Learn to write simple programs using keywords and operators, symbols and functions/commands
4. Learn various programming aspects, control statements, using various operators, solving expressions
5. Learn to build the application using all previously learned concepts

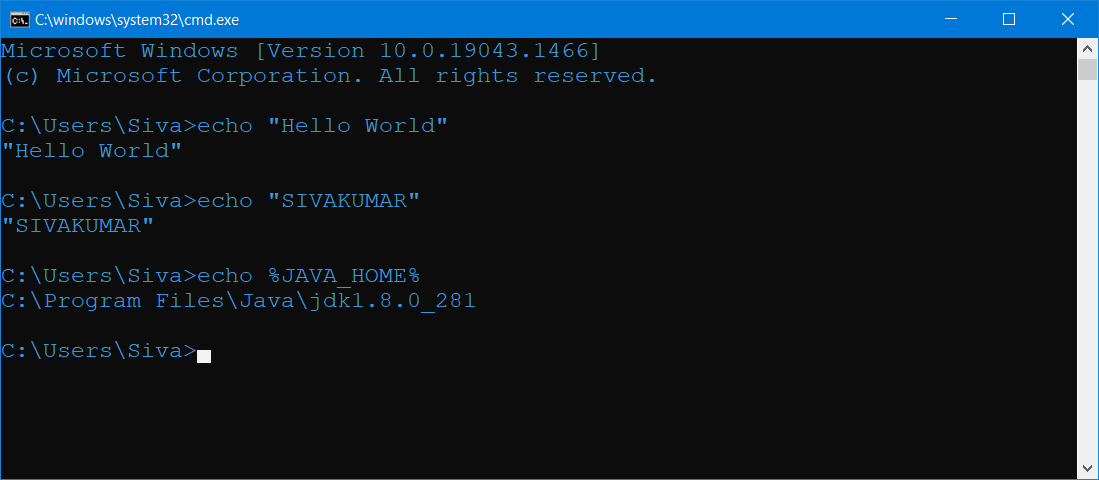
Purpose of Environment variable

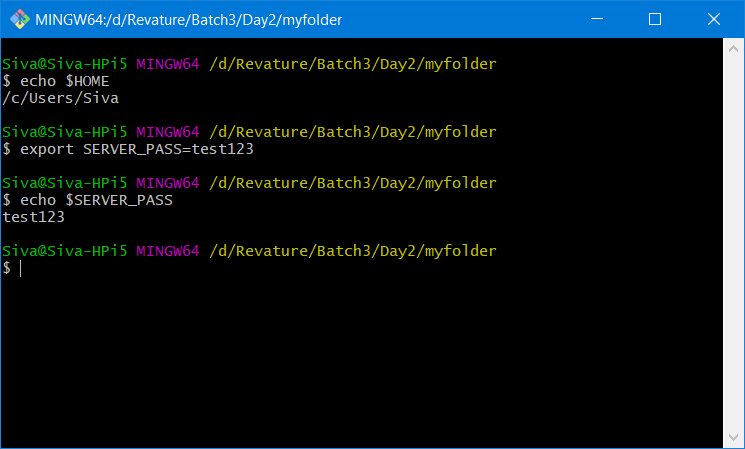
1. I have a Laptop & a printer inside my office room.

When we install a particular software – (exe file), the software could be executed from the folder where it is installed (default behavior) . It can’t be accessible from any other location

Environment variable - path –

Echo %JAVA\_HOME% (dos) === echo $HOME (unix)



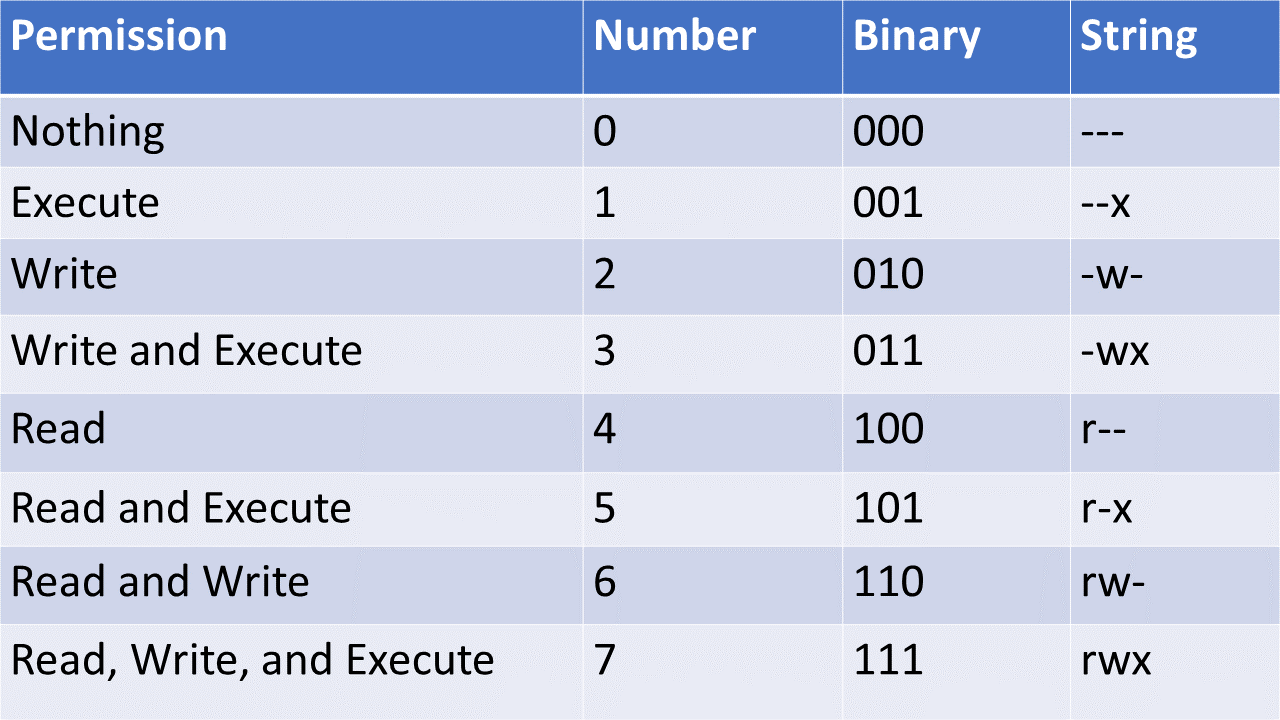


**File Permissions**

In Unix, file permissions can become very important. Every file in Unix has three types of permissions.

1. Owner permissions - What the owner of the file is allowed to do to the file.
2. Group permissions - What the group of users that the file belongs to is allowed to do to the file.
3. Other (world) permissions - What everyone else is allowed to do to the file.

Each type of permission can have any combination of read (r or 4), write (w or 2), and execute (x or 1) permissions. You can represent these permissions as either a number or a string of characters. For example, someone with read and write permissions has 6 or rw permissions.



**Viewing permissions**

You can see the permissions of files in a directory by using the -l flag on the ls command to get it to print the "long listing format"

ls -l .

**Changing permissions**

You can change the permissions on a file using the chmod or change file mode bits command.

If you are using the numbers, it is as simple as specifying the correct permissions for the owner, groups, and other types and issuing the command targeting the file. For example, if I wished the owner to have read, write, and execute, groups to have read and execute, and other to have read permissions on hello.text I would issue the following command:

chmod 754 hello.txt

Otherwise, if you are using the strings, you have to specify which groups you are granting or revoking access to. Owner (u), group (g), and other (o) can be specified or all (a) can be used. The + indicates a granted privilege, - indicates a revoked privilege, and = allows you to set privileges.

Examples:

# Everyone has all permissions

chmod a+rwx hello.txt

# Revoke write access to other

chmod o-w hello.txt

# Set group access to be the same as other

chmod g=o hello.txt