# Introduction to Technical Programming

1.1 Different types of hardware of a common system

**Content:**

• System Unit

• Motherboard

• CPU

• Memory

• Input and Output hardware

Learning Outcomes:

Students should be able to:

1.1.1 Identify the components of the system unit

1.1.2 Explain the term CPU and its purpose

1.1.3 Describe the term CPU and explain the impact of using various types of CPU’s

1.1.4 Describe different types of CPUs regarding use, and power

1.1.5 Define the term computer memory

1.1.6 Discuss the **primary purpose of memory**

**1.1.7 Differentiate between different types of memory and their purpose (Range: RAM, ROM, CMOS, Cache memory, Flash Memory)**

**1.1.8 Explain how data is stored** on memory

1.1.9 Define the purpose of the motherboard and its components

1.1.10 Describe different types of input hardware regarding use and classification e.g., direct, and indirect entry

1.1.11 Discuss how data is transferred between memory i.e., primary, and secondary and the CPU

1.1.12 Describe the Flow/transfer of data between components. (Range: USB – PnP, U3, Point-to- point connections)

* + 1. Describe the factors to consider when choosing an input device. (Range: Ergonomic considerations, Wireless vs cables)
    2. Describe different types of output hardware
    3. Describe the purpose and use of devices such as docking stations for mobile and laptop computers.

1.2 **Purpose of software**

Content:

* Software as a component of a computerised system
* Categories of software

Learning Outcomes:

*Students should be able to:*

* + 1. Describe the term software
    2. Describe the purpose and function of software
    3. Describe the basic concepts of software. (Range: Software as programs. Identify software components. Concept of a graphical user interface (GUI).)
    4. Contrast: System software vs application software
    5. Differentiate between: Shareware, Freeware, Open-Source Software and Proprietary software, Firmware
    6. Discuss the process of how software is obtained and installed.
    7. Differentiate between online software and installed software.
    8. Discuss the following terms in relation to software. (Range: Compatibility issues, Versions, patches, and service packs, Updating software)

1.3 **The Linux shell**

Content:

• Working with the Linux Bash terminal

Learning Outcomes:

Students should be able to:

* + 1. Launch a new Linux terminal on the Raspberry Pi
    2. Use the man command to get help
    3. Expand a Linux file path and explain each element
    4. List the contents of the current folder using the ls command
    5. Change directly location using the cd command
    6. Create a new folder using the mkdir command
    7. Remove a folder using the rmdir command
    8. Remove a file using the rm command
    9. Rename a file using the mv command
    10. Copy a file using the cp command
    11. Clear the command prompts screen using the cls command
    12. Run an executable file from the command line

**Problem solving in computer programming**

2.1 **Problem solving process and concepts**

Content:

* Problem solving
* Application of problem-solving constructs
* Developing solutions

*Learning Outcomes:*

*Students should be able to:*

2.1.1 Define the term problem solving

2.1.2 Define the term computational thinking

2.1.3 Describe the phases of the PLDC (Program Development Life Cycle)

2.1.4 Describe the purpose of problem solving leading to solutions

2.1.5 Explain and apply various problem-solving steps. Polya, G., 1957) (Range:Understand the problem (task/problem description or scenario/user stories) State in own words Clarity on what needs to be done

What is known or given?

What is missing or needed?

Devise a plan/algorithm (storyboard – visual or textual)

Look for patterns

Look at related problems, known solutions

Examine simpler or special cases

Make a table, create diagram, use guess and check, work backwards, identify sub-goal

Carry out the plan/implement the algorithm (write the code)

Look back/test (see if it works)

Check results against original problem. Does it make sense? Is there another solution?)

2.1.6 Use appropriate tools and techniques to present a solution. Range:

User stories (written by the client and provide the requirements)

Noun-verb analysis of user stories

List of nouns provides identification of objects and state

List of verbs provides identification of behaviour

Acceptance tests (does the program meet the requirements?)

2.2 **Construct an algorithm and present a solution to a given problem**

Content:

* Problem solving
* Algorithm design
* Flowcharts

*Learning Outcomes:*

*Students should be able to:*

2.2.1 Define the term algorithm and its purpose in the problem-solving process. (Range: Basic concepts of an algorithm. What is an algorithm? Develop a clear understanding of the problem presented.)

2.2.2 Implement and understand the basic algorithmic constructs used to create a **flowchart.** Range: Input, Output, Processing and Calculations, Selection Iteration

2.2.3 Create a flowchart to present a particular algorithm and its associated tasks

2.2.4 Interpret a basic flow chart and describe its intended operation / function

1.1 Different types of hardware of a common system

After you have completed this module, you should be able to :

* Identify the components of the system unit
* Explain the term CPU and its purpose
* Describe the term CPU and explain the impact of using various different types of CPU’s
* Describe different types of CPUs with regard to use, and power
* Define the term computer memory
* Discuss the primary purpose of memory
* Differentiate between different types of memory and their purpose (Range: RAM, ROM, CMOS, Cache memory, Flash Memory)
* Explain how data is stored on memory
* Define the purpose of the motherboard and its components
* Describe different types of input hardware with regard to use and classification e.g. direct and indirect entry
* Discuss how data is transferred between memory i.e. primary and secondary and the CPU
* Describe the Flow/transfer of data between components. (Range: USB – PnP, U3, Point-to-point connections)
* Describe the factors to consider when choosing an input device. (Range: Ergonomic considerations, Wireless vs cables)
* Describe different types of output hardware
* Describe the purpose and use of devices such as docking stations for mobile and laptop computers.
* Describe the term software
* Describe the purpose and function of software
* Describe the basic concepts of software. (Range: Software as programs. Identify software components. Concept of a graphical user interface (GUI).)
* Contrast: System software vs application software
* Differentiate between: Shareware, Freeware, Open Source Software and Proprietary software, Firmware
* Discuss the process of how software is obtained and installed.
* Differentiate between online software and installed software.
* Discuss the following terms in relation to software. (Range: Compatibility issues, Versions, patches and service packs, Updating software)
* Launch a new Linux terminal on the Raspberry Pi
* Use the man command to get help
* Expand a Linux file path and explain each element
* List the contents of the current folder using the ls command
* Change directly location using the cd command
* Create a new folder using the mkdir command
* Remove a folder using the rmdir command
* Remove a file using the rm command
* Rename a file using the mv command
* Copy a file using the cp command
* Clear the command prompts screen using the cls command
* Run an executable file from the command line

# 1.1 Different types of hardware of a common system

### Introduction

Our daily lives have become increasingly reliant on computers for everyday tasks. As a result, computer programming has become an essential skill. Even though not everyone has the skill, those who choose to be programmers must understand how computers work. Knowledge of computer architecture increases the understanding of some programming concepts, not simply knowing them. This book introduces the types of hardware to begin with a basic understanding of how a system functions.

### Identify the components of the system unit

**Vocabulary**

A system unit is the part of a computer that houses the primary devices that perform operations and produce results for complex calculations.

In this section, we will explain what a system unit is and its purpose on a computer. The system unit contains components such as the motherboard, the central processing unit (CPU), random access memory (RAM), and other devices. This unit performs most of the tasks that a computer is required to perform. In general, system units are used to differentiate between the computer itself and its peripheral devices, such as the monitor, keyboard, and mouse. In layman's terms, a system unit is also called a chassis or a tower. Figure 1.1 illustrates some components making up a system unit. Some of the components are too small to show diagrammatically and we will try to explain them in detail.

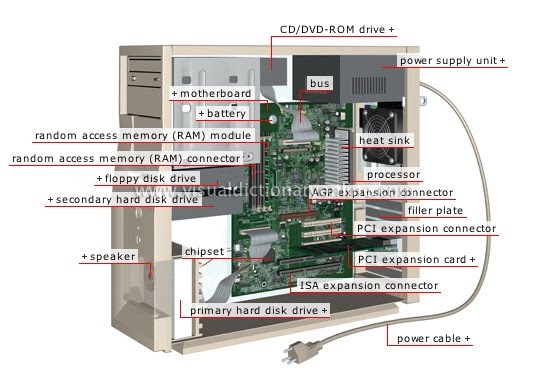


Figure 1. 1: System Unit components

**General functions of components**

Providing proper system functioning is dependent on each component's unique functions. Here are some of the general functions of the main components:

* **Motherboard**. Since connecting all computer nodes to one another is this device's primary function, it is essentially just a series of cables connecting the processor and memory modules.
* **Processor** The computer's "brain," or core, or the element that handles the majority of the work is called the processor. The processor receives data from other parts of the computer and uses it to make computations. The data is returned to the many devices that utilise it after mathematical calculations.
* **RAM** Data storage and quick access are goals of RAM. After the power is cut off, all data on it is lost. Therefore, RAM is referred to being volatile. Volatile memory is a type of memory that maintains its data only while the device is powered. If the power is interrupted for any reason, the data is lost.
* Hard Drive- The hard drive is intended to serve as the computer's permanent storage for files such as music, movies, pictures, and documents.
* **Video Card** converts the image in the computer’s memory into a video signal for the monitor.
* **Power Supply** provides power to the PC. It also performs the functions of stabilization and protection against minor interference in the electrical network.

### 1.1.2 Explain the term CPU and its purpose

**VOCABULARY**

The computer's central processing unit (CPU) is the portion of a computer that retrieves and executes instructions. The CPU is essentially the brain of a computer system.

The acronym CPU stands for Central Processing Unit. The CPU is placed into a specific square-shaped socket found on all motherboards by inserting its metallic connectors or pins found on the underside. Each socket is built with a specific pin layout to support only a specific type of processor.

CPU is the part of a computer that handles the data and activities of the various physical components of the computer. It transfers instructions between a computer's hardware and software. It is also known as a processor, microprocessor, or central processor. Inputs enter a computer and travel to the CPU

### 1.1.3 Describe the term CPU and explain the impact of using various types of CPU’s

The CPU executes the instructions and delivers the results to the associated output. Basically, the CPU is the heart of a computer. It takes in the necessary information and processes it, thus allowing the computer to function. A CPU has three main parts: arithmetic logic unit (ALU), control unit (CU), and memory unit. Based on the von Neumann architecture, the CPU is equalled to the brain of a computer since it controls everything in the computer system. John von Nuemann a Hungarian-American [mathematician](https://en.wikipedia.org/wiki/Mathematician), [physicist](https://en.wikipedia.org/wiki/Physicist), [computer scientist](https://en.wikipedia.org/wiki/Computer_scientist), [engineer](https://en.wikipedia.org/wiki/Engineer). He made great contribution in computer architecture.

**Instruction Processing cycle**

When a program is being carried out, the CPU implements the fetch – decode – execute cycle, which recurs repeatedly until arriving at the STOP instruction. Here are the common five stages of the fetch decode execute cycle.

1. **Fetch- Instruction from Memory (Instruction Fetch, IF)**

Each instruction is stored in memory and has its own address. The processor takes this address number from the program counter, which is responsible for tracking which instructions the CPU should execute next. Once the instruction is fetched, the program counter (PC) will have the address of the next instruction to be executed. A program counter is also known as an instruction counter, instruction pointer, instruction address register or sequence control register.

**VOCABULARY**

A program counter (PC) is a CPU register in the computer processor which has the address of the next instruction to be executed from memory.

1. Decode the instructions into binary (Instruction Decode, ID).

All programs to be executed are translated into Assembly instructions. Assembly code must be decoded into binary instructions, which are understandable to your CPU. This step is called decoding.

1. Execute action and move to next step or calculate address (EXE).

While executing instructions, the CPU can do one of three things: Do calculations with its ALU, move data from one memory location to another, or jump to a different address.

1. Access memory operand (MEM).

The CPU must give feedback after executing an instruction, and the output data is written to the memory.

1. Write back result to register (WB).

If we need to store the result in the destination location, it is done during the writeback stage, and the register file is updated

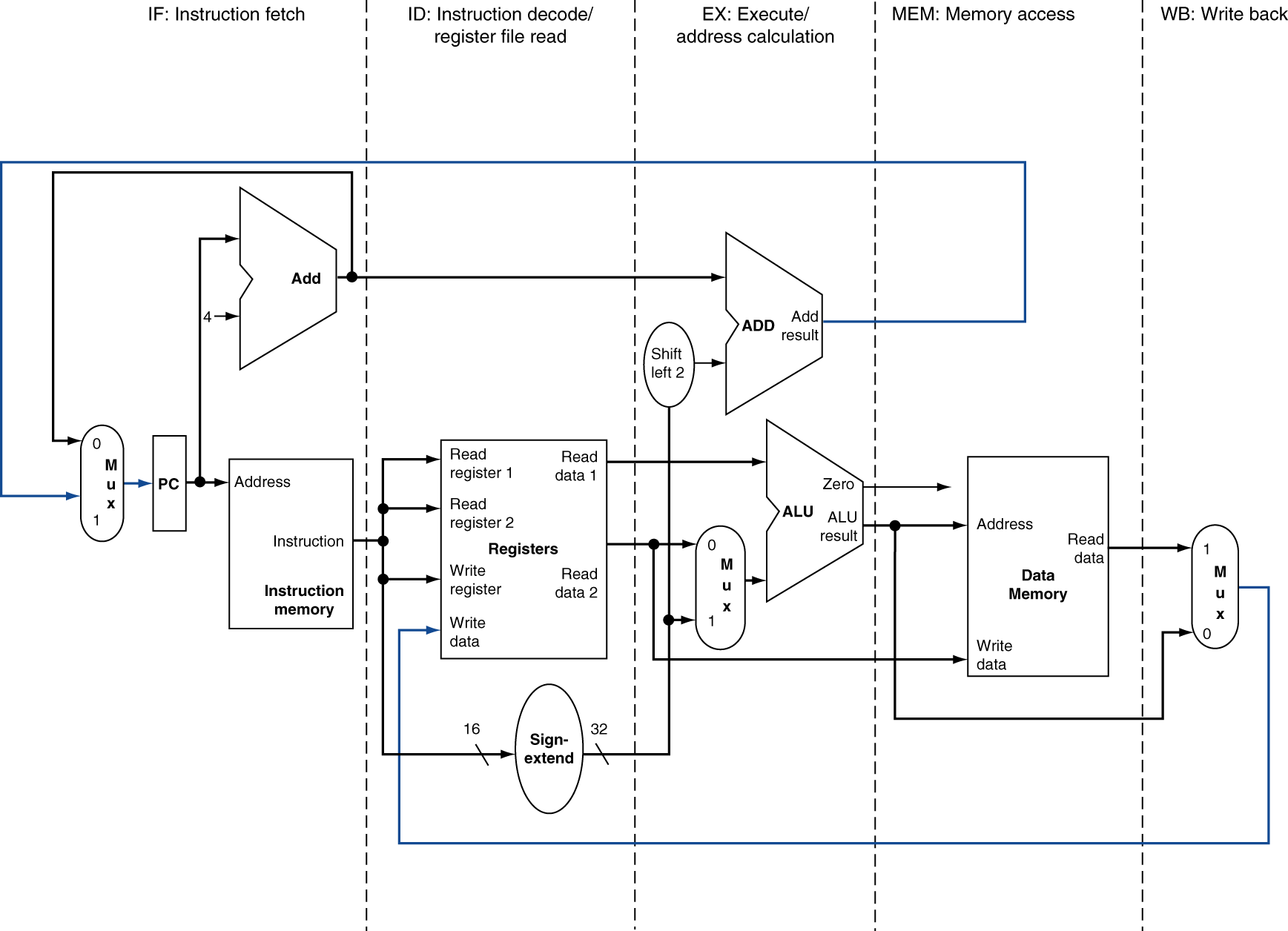


Figure 1. 2: 5 stages of processing

Each instruction must be completed in one cycle. Logically, instruction execute one at a time but physically instructions execute in parallel through a technique called pipelining. Pipelining is an implementation technique where multiple instructions get overlapped during execution. Pipelining enhances computer throughput.

A CPU's speed determines how many operations it can carry out. The rate at which one operation is completed in a second is measured in hertz. A computer's speed is typically expressed in gigahertz. The CPU can complete one million simple jobs at 1 GHz speed. The simplest actions a processor can take are considered "simple tasks."

The more instructions your CPU can process in a second, the faster it is, but do not be deceived by this figure. The performance of your computer is impacted by a variety of factors, not only the CPU's speed. To obtain independent results, numerous additional variables, like CPU architecture, cache capacity, and bus speed, must be considered.

The central processing unit (CPU) has three components:

* Control Unit
* Arithmetic Logic Unit
* Registers

**Control Unit**

The control unit reads instructions from memory, decodes them, and then puts them into action. The control unit serves as a middleman, decoding the instructions provided to the processor, directing the other units, such as the Arithmetic Logic Unit (below), using control signals, and sending the processed data back to memory.

**Arithmetic Logic Unit (ALU)**

Digital circuitry inside the processor called an arithmetic logic unit (ALU) handles arithmetic and logical operations by loading data from input registers. The ALU completes the required operations by connecting many transistors after receiving the necessary instructions from the control unit, and then it records the outcomes in an output register. The system clock, memory, secondary storage, and data and address buses are all necessary for the CPU to operate effectively. Some of the arithmetic operations include addition subtraction, multiplications, and divisions. The result of arithmetic computations is put into the accumulator.

**Register** – saves the most frequently used instructions and data.  Processor registers are a common name for the registers used by the CPU. An instruction, a storage address, or any other data, including bit sequences or single characters, may be stored in a processor register.

There are 6 types of central processing units Single Core, Dual Core, Quad Core, Hexa Core , Octa Core, and Deca Core.

Single Core CPU- consists of a single CPU and can execute one instruction at a time.- An example is the Intel 4004 released in 1971.

Dual Core CPU- Consists of two cpu’s that act like one CPU meaning the CPU can multitask e.g Pentium D released in 2005.

Quad-Core CPU-A quad-core CPU has **four cores on a single CPU processor**. The CPU is the greatest choice for multitasking since it equally distributes the workload among its cores. An example of quad core CPU is Athlon II X4 which was released in 2009.

Some computers utilize two or more processors. These consist of separate physical microprocessors located side by side on the same board or on separate boards. Each CPU has an independent interface, separate cache, and individual paths to the system front-side bus. As you will notice from Figure 1.3, the name depicts the number of cpus e.g. Hexa core has 6, octa core has 8 and deca core has 10 cpu’s.



Figure 1. 3: Types of CPU's

Intel and AMD are the two key manufacturers of CPU’s.  These manufacturers are both great, though they tend to be used in different situations.

### Describe different types of CPUs with regard to use, and power

Section 1.1.3 outlined the types of processors based on number of cores. Processors can also be categorised based on:

* Data bus width e.g., 32 bit and 64 bit processors
* Brand name and manufactures e.g., Intel and AMD
* Architecture- e.g., Reduced Instruction Set Computing RISC and Complex Instruction Set Computing (CISC)
* General purpose processors based on architecture- in this category we have 5 types which are explained below

Let us differentiate the RISC and CISC processors.

Key Difference between RISC and CISC processor

* In RISC, the instruction set is reduced, and most of these instructions are very primitive, while in CISC, the instruction set is very large that can be used for complex operations.
* RISC computer’s execution time is very less, whereas CISC computer’s execution time is very high.
* RISC code expansion may create a problem, while CISC code expansion is not a problem.
* In RISC, the decoding of instructions is simple, whereas, in CISC, the decoding of instructions is complex.
* RISC doesn’t require external memory for calculations, but CISC requires external memory for calculations.
* RISC has multiple registers sets present, while CISC has only a single register set.

There are five types of general-purpose processors they are, Microcontroller, Microprocessor, Embedded Processor, DSP and Media Processor.

Microcontroller

A microcontroller is basically a type of computer that comes in different packages and sizes. Feedback on reading input and output is the main function of this Microcontroller. It is also commonly referred to as General Purpose Input Output (GPIO).



Figure 1. 4: Example of a microcontroller (  
PIC32MX Series Microcontrollers)

Microprocessor

The general-purpose processors are represented by the microprocessor in embedded systems. There are different varieties of microprocessors available in the market from different companies. The microprocessor is also a general-purpose processor that consists of a control unit, ALU, a bunch of registers also called scratchpad registers, control registers and status registers.



Figure 1. 5:Example of Microprocessor

Microprocessors are cheap, offer high speed, small in size, consumes less power and generate less heat.

Microprocessors uses a clock signal to control the rate at which instructions are executed, synchronize other internal components and to control the data transfer between them.

**Embedded Processor**

An embedded processor is one type of processor which is designed to control mechanical functions and electrical functions. It consists of several blocks such as the processor, timer, an interrupt controller, program memory and data memory, power supply, reset and clock oscillator circuits, system application-specific circuits, ports and interfacing circuits. Embedded processors are designed especially for coping with the wishes of an embedded gadget.

Embedded processors may be observed in transportable devices like virtual watches, PDAS, digital cameras, GPS gadgets and MP3 players. They also can be located in large structures such as visitor’s lights, systems controlling energy vegetation and factory controllers.

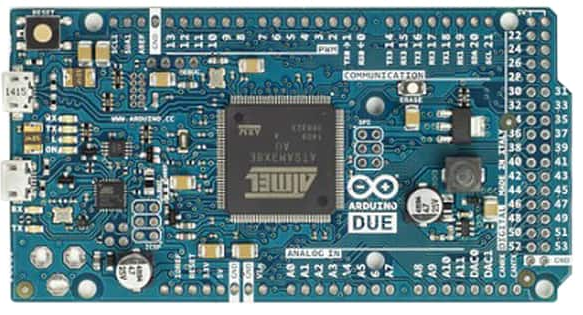


Figure 1. 6:Embedded processors

Digital Signal Processor

The digital signal processor is one type of processor used for measuring, filtering and/or compress digital or analog signals. Digital Signal Processors (DSP) take real-world signals like voice, audio, video, temperature, pressure, or position that have been digitized and then mathematically manipulate them. A DSP is designed for performing mathematical functions like "add", "subtract", "multiply" and "divide" very quickly.

The signal processing means analysis and manipulation of signal and can be done via computer or Application Specific Integrated Circuits (ASIC), Field Programmable Gate Array (FPGA) or Digital Signal Processor (DSP) to obtain the clear signal. The DSP processors are used in an oscilloscope, barcode scanners, mobile phones, printers, etc.

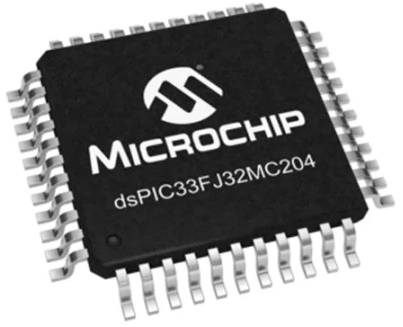


Figure 1. 7: Example of 16bit Digital Signal Processor 40MIPS

### Define the term computer memory

VOCABULORY

Computer memory is the storage space in the computer, where data is to be processed and instructions required for processing are stored.

The memory is divided into large number of small parts called cells. Each location or cell has a unique address, which varies from zero to memory size minus one. For example, if the computer has 64k words, then this memory unit has 64 \* 1024 = 65536 memory locations. The address of these locations varies from 0 to 65535. The CPU is responsible for selecting memory cells to read or write data.

The concept of memory and storage can be easily conflated as the same concept; however, there are some distinct and important differences. Put succinctly, memory is primary memory, while storage is secondary memory. We will explain these terms in detail in section 1.1.7.

### 1.1.6 Primary purpose of memory

Memory is central to a computer's operation because it forms the critical link between software and the CPU. The basic function of computer memory is essentially to store data. Depending on the type of data it stores and the role it plays in computer operation, however, memory performs several different functions. Although all of these functions involve data storage, RAM, ROM, flash memory and hard drives each perform a different and necessary function to keep a computer and its peripherals working. Computer memory likewise decides the size and number of programs that can be run all the while, and helps to optimize the capabilities of capable microprocessors

Once the data stored in computer memory, the data will remain there forever. Each time memory is full, then the data can be deleted in part or in whole to be replaced with new data. The performance of a computer depends on memory and CPU. CPU can not store programs or a large set of data permanently. They are only capable of storing basic instructions required to operate the computer. Therefore it is mandatory to have the memory to run a computer system properly.

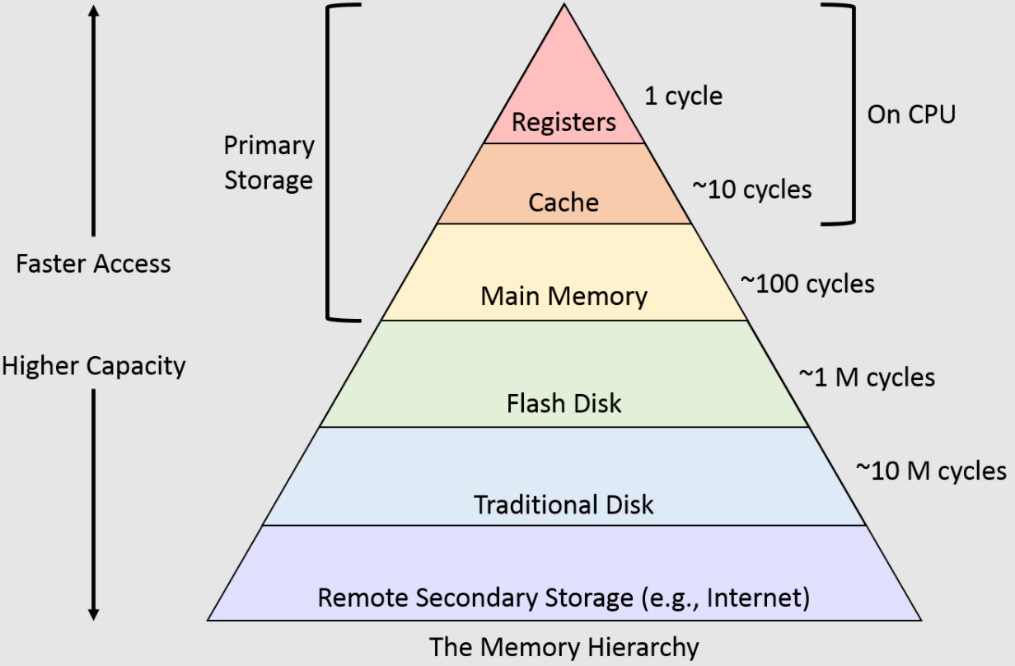
### 1.1.7 The different types of memory and their purpose (Range: RAM, ROM, CMOS, Cache memory, Flash Memory)

There are basically 2 broad categories of memory.



Figure 1. 8:Categories of Computer Memory

To further assist us in understanding computer memory, we will use a computer hierarchy chart and then explain each of these types.



In computer architecture, the memory hierarchy is an improvement of computer storage into a hierarchy based modal on response time. It affects performance in a computer algorithm, predictions, architectural design, and lower-level programming constructs involving the locality of reference.

**Registers**

Register memory is the smallest and fastest memory in a computer. A register temporarily holds frequently used data, instructions, and memory address that are to be used by CPU. They hold instructions that are currently processed by the CPU. All data is required to pass through registers before it can be processed.  Some of the widely used registers include Accumulator or AC, Data Register or DR, the Address Register or AR, Program Counter (PC), I/O Address Register, and these are explained in Table 1.1.

|  |  |
| --- | --- |
| **Type of register** | **Uses** |
| Data register | stores data, which is being transmitted to or received from a peripheral device. |
| Program Counter | It holds the address of the memory location of the next instruction to be fetched. |
| **Instructor Register** | hold instruction codes and passes them to the Control Unit to be decoded. |
| **Accumulator Register** | is a type of register for short-term, intermediate storage of arithmetic and logic data in a computer's central processing unit (CPU). |
| Address Register | it is a 12-bit register that stores the address of a memory location where instructions or data is stored in the memory. |

Table 1. 1: Common registers

**Cache Memory**

Cache memory is a high-speed memory, which is small but faster than the main memory (RAM). The CPU can access it more quickly than the primary memory. So, it is used to synchronize with high-speed CPU and to improve its performance.

When CPU needs the data, first, it looks inside the L1 cache. If it does not find anything in L1, it looks inside the L2 cache. If again, it does not find the data in L2 cache, it looks into the L3 cache. If data is found in the cache memory, then it is known as a cache hit. On the contrary, if data is not found inside the cache, it is called a cache miss or miss penalty. The solution to reducing cache miss is through implementing several or multilevel caches. **Multilevel Caches** is one of the techniques to improve cache performance by reducing the *“MISS PENALTY”*.

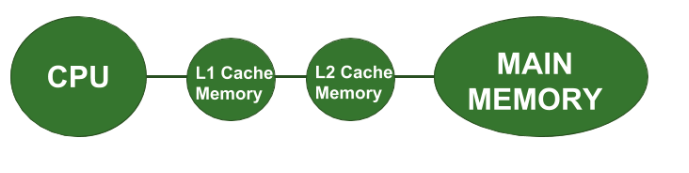


Figure 1. 9: Multilevel caches

**Primary Memory**

Primary Memory is of two types: RAM and ROM.

**Random Access Memory (RAM)**

It is a volatile memory meaning it does not store data or instructions permanently. When you switch on the computer the data and instructions from the hard disk are stored in RAM.

**Advantages of RAM**

1. It is a faster type of memory in a computer.
2. It requires less power to operate.
3. Program loads much faster
4. More RAM increases the performance of a system and can multitask.
5. Perform read and write operations.
6. The processor can read information faster than a hard disc, floppy, USB, etc.

**Disadvantages of RAM**

1. Less RAM reduces the speed and performance of a computer.
2. Due to volatile, it requires electricity to preserve the data.
3. It is expensive than ROM
4. It is unreliable as compared to ROM
5. The Size of RAM is limited.

There are two types of RAM : Static RAM SRAM) and Dynamic RAM (DRAM)

|  |  |  |
| --- | --- | --- |
| parameter | SRAM | DRAM |
| Read & Write | Faster | Slower than SRAM |
| Storage component | Uses transistor to store single bit of information | Uses separate capacitor to store each bit of data |
| Price | Expensive than DRAM | Cheaper than SRAM |
| Power consumption | More | Less |
| Uses | Cache memory | Main memory |
| Density | Less dense | Highly dense |
| Storage per bit | Can store many bits per chip | Cannot store many bits per chip |

**Read-Only Memory (ROM)**

ROM is a memory device or storage medium that is used to **permanently** store information inside a chip. It is a read-only memory that can only read stored information, data or programs, but we cannot write or modify anything. A ROM contains some important instructions or program data that are required to start or boot a computer. It is a **non-volatile** memory.

There are five types of Read Only Memory:

**MROM (Masked Read Only Memory):-** program or data is pre-configured by the integrated circuit manufacture at the time of manufacturing. Its permanent.

**PROM (Programmable Read Only Memory):** It is a type of digital read-only memory, in which the user can write any type of information or program only once.

**EPROM (Erasable and Programmable Read Only Memory)-** stored data can be erased and re-programmed only once in the EPROM memory.

**EEPROM (Electrically Erasable and Programmable Read Only Memory)-** **A high voltage electrical charge is used to erase recorded data from EEROM.**

**Flash ROM: data or instructions** can be written or programmed in small units called block or sector.

**Advantages of ROM**

1. It is a non-volatile memory
2. It is static, so it does not require constant refreshing every time.
3. These cannot be changed accidently
4. It is cheaper than RAM.
5. Simple and reliable

**Disadvantages of ROM**

1. Store data cannot be updated or modify except to read the existing data.
2. It is a slower memory than RAM to access the stored data.

**CMOS- complementary metal-oxide semiconductor**

Alternatively referred to as a RTC (real-time clock), NVRAM (non-volatile RAM) or CMOS RAM, CMOS is short for complementary metal-oxide semiconductor. CMOS is an onboard, battery powered semiconductor chip inside computers that stores information. This information ranges from the system time and date to system hardware settings for your computer.

**Which devices use CMOS?**

* [Microprocessors](https://www.computerhope.com/jargon/c/cpu.htm)
* Microcontrollers
* [Digital logic circuits](https://www.computerhope.com/jargon/l/logicirc.htm)
* [SRAM](https://www.computerhope.com/jargon/s/sram.htm) (Static RAM)

**Secondary Memory**

The secondary storage devices which are built into the computer or connected to the computer are known as a secondary memory of the computer. It is also known as external memory or auxiliary storage. The secondary memory is accessed indirectly via input/output operations. It is non-volatile and can not be accessed directly by the CPU.

**Examples of secondary memory**

1. Hard Disk Drives
2. Solid State Drive
3. Pen Drive/USB flash drive
4. Secure Digital Card -SD Cards
5. CD/ DVDs

**Cloud Storage**

A service paradigm known as "cloud storage" involves sending and storing data on remote storage systems, where it is then maintained, managed, backed up, and made accessible to users across a network, most often the internet. Users often pay a monthly, per-consumption fee for the storage of their cloud data. Some of the providers are AWS.

### 1.1.8 Explain how data is stored on memory

Data is first transformed into straightforward numbers (Binary Digits-Bits) that a computer can easily store. Bit is the basic unit of memory. At a time, it can be either on or off. Generally, bits are represented using electrical voltage. Voltage presence indicates that the bit is in ON state. Voltage absence indicates that the bit is in OFF state. Here, OFF state is considered as 0. ON state is considered as 1. Computer memory is the collection of several bits. Group of 8 bits are called byte. Second, circuitry within the computer records the numbers. Third, software or programs are used to organize, move, and manipulate the numbers.

**Memory Units**

The smallest unit is a bit

I Byte=8 bits

1 Kilobyte=1024 bytes

1 Megabyte=1024 Kilobytes

1 GigaByte=1024 MegaBytes

1 TeraByte=1024 GigaBytes

### 1.1.9 Define the purpose of the motherboard and its components

A motherboard, often referred to as an mboard, mainboard, base board, system board, planar board, or main circuit board, is the main board and the building block of a computer. A logic board is what it is known as on Apple computers. The motherboard serves as a single platform to connect all of the parts of a computer together. It connects the CPU, memory, hard drives, optical drives, video card, sound card, and other ports and expansion cards directly or via cables. The motherboard of the computer houses the CPU, RAM expansion slots, ROM, USB ports, and PCI slots. It enables communication between the RAM, CPU, and every other piece of hardware. Figure 1.10 shows an example of a motherboard.



Figure 1. 10: i7 Gigabyte Q87M Motherboard

Apart from the other components- the motherboard has a Northbridge and Southbridge

**Northbridge:** An integrated circuit in the motherboard's chipset is in charge of establishing a connection between the AGP, CPU interface, and memory. It has direct connections to the CPU interface, AGP, and memory, unlike southbridge. Northbridge's main responsibility is to provide bus-based communication between the CPU and external devices.

**Southbridge:** This motherboard component is an integrated circuit with a single purpose for which it was produced. I/O controllers, hard drive controllers, and integrated hardware all depend on it.

Some of the popular manufacturers of motherboards includes Intel, ASUS, ABIT, BioStar, Gigabyte and MSI.

**Functions of the Motherboard**

**The functions of a computer motherboard are as follows:**

1. The motherboard acts as the central backbone of a computer on which other modular parts are installed such as the CPU, RAM and hard disks.
2. The motherboard also acts as the platform on which various expansion slots are available to install other devices / interfaces.
3. The motherboard is also responsible to distribute power to the various components of the computer.
4. They are also used in the coordination of the various devices in the computer and maintain an interface among them.

Below are the six different types of Motherboards:

* **AT & Baby AT**
* **Advanced Technology eXtended ATX**
* **BTX (Balanced Technology Extended):**
* **DTX Discontinuation Transmission**
* **LPX (Low Profile eXtension)**
* **microATX**
* **NLX-  New Low Profile Extended**

### 1.1.10 Describe different types of input hardware with regard to use and classification e.g. direct and indirect entry

Hardware- Hardware (sometimes abbreviated to HW) can be defined as the physical components that a computer system needs to function.

Software-   consists of written, machine-readable instructions or [programs](https://www.techtarget.com/searchsoftwarequality/definition/program) that tell physical components what to do and when to execute the instructions.

Hardware can be split into input devices, processing devices, output devices and storage devices.

Input devices are used to get data and instructions into the computer. Processing devices are used in the processing of data once it gets into the computer. Input devices can further be split into direct and indirect input hardware.

Direct data entry devices are specific purpose devices designed to automate or speed up the entry of data into the system by minimising human data entry. They have a wide range of uses including in education, retail and in business. They consist of either specialist hardware, software (or both) and come in a number of different forms. Examples includes:

Magnetic stripe reader-– used to read data from magnetic stripes on mostly banking cards, membership cards or hotel door cards. The stripe on the cards holds data such as membership information.



Figure 1. 11: Magnetic Stripe Reader

Chip readers-read data from the chip on bank cards or shopping cards. The chip and pin reader works by inserting the card into a slot and then entering a PIN (personal identification number)

A picture containing person, indoor, remote, game

Description automatically generated

Figure 1. 12: Chip reader

PIN pads- use to enter data into Automated Teller Machines (ATM), EFTPOS system, entry doors and handheld devices.

Optical mark reader- used to read and input information from a form made in pen or pencil. Typically used to read multiple choice questions. The OMR shines on to the form and less light is reflected where a pencil mark has been made.



Figure 1. 13: Optical Mark Reader

Barcode reader- used to scan codes directly from the products, books and membership cards. Figure 1.14 shows a picture of a Barcode Reader.



Figure 1. 14: Barcode Reader

If the data is in human readable form, it must be converted into machine readable form so that a computer can process it. This process of data conversion is time consuming and error prone that causes a major bottleneck in data processing. Some examples of indirect input devices are: keyboard, mouse and joystick. When you press any key on keyboard, it converts that character into series of electronic pulses and sends to CPU.

**Exercise**

Categorise the following devices into direct and indirect input hardware.

* Mouse
* Joy Stick
* Light pen
* Track Ball
* Scanner
* Graphic Tablet
* Microphone
* Magnetic Ink Card Reader(MICR)
* Optical Character Reader(OCR)
* Bar Code Reader
* Optical Mark Reader(OMR)

### 1.1.11 How data is transferred between memory and the CPU

The connections between the CPU and Memory are shown in Figure 1.15.

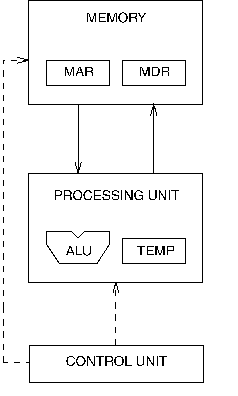


Figure 1. 15:CPU interaction with memory

Communication between memory and processing unit consists of two registers:

* Memory Address Register (MAR).

The Memory Data Register (MDR) keeps the data which is transferred between the Memory and the CPU.

* Memory Data Register (MDR).

The Memory Address Register (MAR) holds the memory location of data that needs to be accessed.

Reading Data

1. The address of the location is put in MAR.
2. The memory is enabled for a read.
3. The value is put in MDR by the memory.

To write,

* 1. The address of the location is put in MAR.
  2. The data is put in MDR.
  3. The **Write Enable** signal is asserted.
  4. The value in MDR is written to the location specified.
* Buses are the means by which data is transmitted from one part of a computer to another, connecting all major internal components to the CPU and memory. Address bus Carries the addresses of data (but not the data) between the processor and memory.
* Data bus carries data between the processor, the memory unit, and the input/output devices.
* Control bus carries control signals/commands from the CPU (and status signals from other devices) to control and coordinate all the activities within the computer.

### 1.1.12 Describe the Flow/transfer of data between components. (Range: USB – PnP, U3, Point-to-point connections)

The pursuit of the fastest speed and throughput has been one of the key goals in the development of computer architecture. By utilizing technological potential in the design of the computer components and by giving the computer an appropriate structure and organization, this goal was and is still achieved in two ways. Due to rapid technological advancement in the manufacture of integrated circuits, where component speed and density are continuously rising while costs are falling, the advancement of computer components is quite rapid.

**VOCABULARY**

Data Bus- In computer terminology, a bus is a communication system that allows the transfer of data between components within a computer, or between separate computers.

Computer bus can be in the form of wired cables or electrical wires embedded in the computer motherboard PCB (Printed Circuit Board).

The function of a data bus is to either allow these components to communicate with each other or with the outside world. A data bus can transfer data to and from the memory of a computer, or into or out of the central processing unit (CPU) that acts as the device's "engine." A data bus can also transfer information between two computers. The amount of data that can be transferred by a data bus is referred to as **bandwidth.** If we consider the speed of bus or bit transfer, then one wire or bus transfers millions of bits per second.

Most modern computers today use both parallel and serial buses. • Parallel data buses: o Carry data on many wires simultaneously o Each wire carries one bit of data o Most common parallel buses: Advanced Technology Attachment (ATA), PC card, Small Computer System Interface (SCSI) • Serial data buses: o Has one wire that carries all the bits of data o Most common serial data buses: Universal Serial Bus (USB), FireWire, Serial ATA, Serial Attached SCSI. Figures 16 and 17 illustrates a data bus and a computer bus respectively.



Figure 1. 16: Data Bus



Figure 1. 17: Computer Bus

Apart from data bus, there is also an address bus and a control bus.

* The address bus is used to transfer address bits to the memory.
* The control bus is used to transfer control bits from control units to other components of the computer.

Data transfer from Universal Serial Bus

When the software requires data transfer to occur between itself and the USB, it sends a block of data called an *I/O Request Packet (IRP)* to the appropriate pipe, and the software is later notified when this request is completed successfully or terminated by error.

The actual data is sent across the bus in packets. Each *packet* is a bundle of data along with information concerning the source, destination and length of the data, and also error detection information.

Each packet is made up of a set of components called *fields* including the following, summarised in figure 4 :

Table 1: A typical data packet. numbers represent size of field in bits, unless otherwise indicated.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sync (8) | PID(8) | Address | Endpoint | Data (0-123 bytes) |

* An eight bit "*SYNC*" synchronisation field used by inputs to correct their timing for accepting data. Marks the start of a data packet.
* The 8 bit Packet Identifier (*PID*) which uses 4 bits to determine the type, and hence format, of the packet data. The remaining 4 bits are a 1's complement of this, acting as check bits. Part of this field determines which of the four groups (token, data, handshake, and special) that the packet belongs to, and specifies an input, output or setup instruction.
* An *address field* which gives the address of the function on the end of the pipe to be used
* The 4 bit *endpoint field*, giving the appropriate endpoint which sends or receives the packet.
* A *data field* consisting of 0-1023 bytes

### 1.1.13 Describe the factors to consider when choosing an input device. (Range: Ergonomic considerations, Wireless vs cables)

Input devices are peripherals from which computers receive data. Keyboards, mice, scanners, and webcams are some examples. If you select the right accessories, you can extend the capabilities of your computer and increase productivity. Make the most of your next computer accessory purchase by understanding the factors to consider when selecting input devices. Here are some of the factors to consider:

1. **User Needs**- this is the urgency of use of the device in the computer room by users also whether it will satisfy the needs of the user.
2. **Initial cost-**the amount it can cost when buying/purchasing the devices
3. **Maintenance Cost**-the amount that can be used to maintain the servicing of these devises should be considered.
4. **Mode Of Transmission**- how will you transport them to the computer room should be considered
5. **Compatibility With Available Hardware**- will the devices fit to other devices in the room already. A device that fits your needs but doesn't work with your computer is useless.
6. **User- Friendliness**- whether the devices will be used to solve problems and easy to be used by users
7. **Wireless vs wired connection**-Cables is typically faster than a Wi-Fi connection, and it offers other advantages as well. Cable connection is more secure and stable than Wi-Fi. You can test your computer's speeds on Wi-Fi versus an Ethernet connection easily.

### 1.1.14 Describe different types of output hardware

**VOCABULARY**

Any peripheral that accepts data from a computer and prints, projects, or reproduces it is known as an output device.

Output devices are used to provide results of the computer to the users. Examples includes **Monitor, Printer, Projectors and Plotters**. The output may be audio, video, hard copy – printed paper, etc. Output devices convert the computer data to human understandable form. We give input to the computer using input devices and the computer performs operations on the data and displays the output to the user using the output device.

**Monitor** - A computer’s principal output device is a monitor, often known as a Visual Display Unit (VDU). It displays the processed data like text, images, videos, audios, etc. There are two types of monitor viewing screens:

* Cathode-Ray Tube (CRT)
* Display on a Flat Panel Monitor with a Cathode-Ray Tube (CRT)
* Plasma Monitor

**Printer**- it is an output device that creates a hard copy of the processed data or information. Printers are divided into two categories:

* Impact Printer
* Non-Impact Printers

1. **Impact Printers**- In impact printers, characters are printed on the ribbon, which is then smashed on the paper. Here, the characters are printed on to the paper by striking an ink ribbon against it with a hammer or print head. Impact printers are relatively cheap making them ideal for large scale printing. Most of these are used in cooperates. Common examples of impact printers include Dot matrix printer, Daisy wheel printer, Line printer and Chain printer.
2. **Non-Impact Printers:**These printers print characters without the use of a ribbon. These printers are fast because they print one full page at a time producing quality printout. The common examples are laser and inkjet printers.

**Plotter**

A plotter is a [printer](https://www.techtarget.com/whatis/definition/printer) that interprets commands from a computer to make line drawings on paper with one or more automated pens. Some of the examples of plotters include drum plotters, flatbed plotters, electrostatic plotters and inkjet plotters.

**Projector**

A projector is an output device that reproduces images by projecting them onto a screen, wall, or other surface using images created by a computer or Blu-ray player.

### 1.1.14 Describe the purpose and use of devices such as docking stations for mobile and laptop computers.

A docking station is a cradle for a portable media player that serves to charge and connect the unit to a receiving device. Docking stations enable users with a [laptop computer](https://www.computerhope.com/jargon/l/laptop.htm) to convert it into a [desktop computer](https://www.computerhope.com/jargon/d/desktopc.htm) when at the office or at home. For example, a business user could use a laptop on the road to create a document. When they return to the office, they could attach the laptop to the docking station to use their monitor, speakers, and office [printer](https://www.computerhope.com/jargon/p/printer.htm). Docking stations are now commonly used by developers in increasing the number of displays to reduce number of times of switching through tabs if they are to use one display.



Figure 1. 18: Ultraslim Docking station

## 1.2 Purpose of software

A computer system consists of

In section 1.1 our discussion mainly centred on hardware components. We must remember that for computers and devices to function, there must be instructions or programs controlling them. Without software, the hardware will not work as expected. We defined software as consisting of written, machine-readable instructions or [programs](https://www.techtarget.com/searchsoftwarequality/definition/program) that tell physical components what to do and when to execute the instructions. In this section we are going to dive deeper into the different software categories and their purposes.

* + 1. Describe the term software

VOCABULARY

Software-Software – set of instructions, data or programs used to operate computers and execute specific tasks.

Software can be thought of as the variable part of a computer, while hardware is the invariable part. The two main categories of software are [application](https://www.techtarget.com/searchsoftwarequality/definition/application) software and [system software](https://www.techtarget.com/whatis/definition/system-software). Application software is software that fulfils a specific need or performs tasks. System software is designed to run a computer's hardware and provides a platform for applications to run on top of.

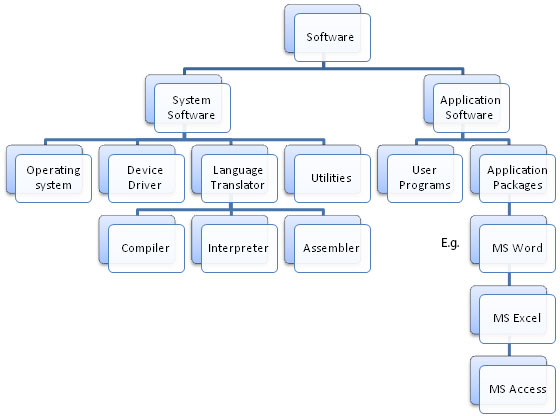


Figure 1.2. 1: Classification of software

System Software

These software packages are made to run the hardware and application software on a computer. The actions and features of the hardware and software are coordinated by the system software. Additionally, it manages how the computer hardware functions and offers a setting or platform in which all other software can operate.

**Operating Software**-An operating system acts as an intermediary between the user of a computer and computer hardware. The purpose of an operating system is to provide an environment in which a user can execute programs conveniently and efficiently. Key functions are file management, memory management, resource management, security management among others. The three basic types of operating systems are stand-alone, server, and embedded.

1. Stand-alone operating system- This is an operating system that runs on a desktop computer, a laptop computer, or a handheld computer is called a stand-alone operating system. Examples are Microsoft Windows, Mac Os, and Linux.
2. Server operating system- these are specifically designed to support the network. Examples includes include windows server, UNIX, Solaris, and Netware.
3. Embedded Operating System-An embedded operating system uses on mobile devices and a wide variety of consumer electronics. It resides on a ROM chip. Common examples are windows mobile Plam OS, iPhone OS, BlackBerry, Google Android, Windows mobile, and Symbian OS.

**Device Drivers**-this software is often considered a type of system software. Device drivers control the devices and peripherals connected to a computer, enabling them to perform their specific tasks.

**Utility software**- This software analyses and maintain a computer. This software is focused on how OS works on that basis it performs task to enable smooth functioning of computer. Examples of utility software includes antivirus, backup software, uninstaller, screen saver, file compression, virus scanner, file manager, disk compression tool all are utility software. Some of the utility software come along with operating software for example windows defender, disk cleanup, disk optimisation tools.

**Language Translators**-Translators are computer programs that translates program written in each programming language into a functionally equivalent program in a different language. There are mainly three Types of translators which are used to translate different programming languages into machine equivalent code: **Compiler, Interpreter, Assembler.**

1. **Compiler** is a computer program that translates code written in a high-level language to a low-level language, object/machine code. Examples of languages using compilers are C, C++, C#, Java VB etc.

**HOW DOES A COMPILER WORK?**

**Here are the stages involved in compiling a program.**

**Lexical Analysis** -Convert a program into sequence of of tokens.

Syntax Analysis - Recover the structure portrayed by utilizing arrangement of tokens from past scanner.

**Semantic Analysis-**Ensures program has a well-defined meaning.

**IR Generation-**Compilers create an explicit low-level or machine-like intermediate representation that should be straightforward to deliver and straightforward for the target machine to understand.

**Code Generation-**The compiler converts the optimized intermediate code to the machine code dedicated to the target machine.

1. An interpreter program executes other programs directly, running through program code and executing it line-by-line. Examples of interpreted languages are Ruby, Perl, Python, PHP etc.

**HOW DOES INTERPRETER WORK?**

Four things happen inside an interpreter

* + 1. **Lexing** — The lexer breaks the line of code into tokens.
    2. **ii. Parsing**- The parser employments these tokens to produce a structure, here, an Unique Language structure Tree, to portray the relationship between these tokens.
    3. **Compiling**- The compiler turns this AST into code object(s).
    4. **Translating**- The translator executes each code.

1. **Assembler**-An assembler translates assembly language into machine code. An assembler takes basic computer instructions and changes them into a design of bits that the computer processor can utilize to perform its fundamental operations.

Compilers take less execution time compared to an interpreter whereas assemblers take more time than the compiler. An interpreter analyses every line; hence is slower than running compiled code but it can take less time to interpret program code than to compile and then run it. Interpreters are written for multiple platforms; this means code written once can be run immediately on different systems without having to recompile for each.

Application software consists of application packages and user programs. Applications packages are always bought as a suite of programs for instance, Microsoft Office, LibreOffice and they provide common end-user needs. Examples of application programs include word processors e.g. Microsoft Word, WordPad, LibreOffice Writer, presentation programs e.g., PowerPoint, databases e.g. Microsoft Access, LibreOffice Base etc. User programs could be any example of programming languages such as C, C#, C++, Java, Kotlin, Go, R, Perl etc.

### Describe the purpose and function of software

**Functions of computer software**

* **Software** can make the computer compare data, make logical decisions, do mathematical calculations, store and retrieve data and instructions from primary or secondary sources, and carry out sequences of tasks.
* **The** **software** enables computer users to obtain what they need from the computer.
* **Software makes** the computer work towards giving the outputs in the manner the user wants them to, such as output on a screen, printouts, sounds, sending emails, etc.
* **Software** is used to translate programs written in different languages into machine languages, e.g., compilers, interpreters, and assemblers.

### 1.2.3 Basic concepts of software.

Operating systems serve as interfaces between users and computer hardware. Figure 1.2.2 illustrates how the operating systems interacts with hardware and providing interface for users. Interfaces are separated into two categories: command-line and graphical.

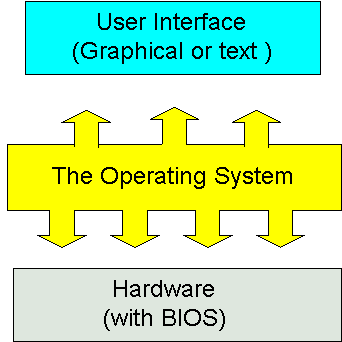


Figure 1.2. 2: Interfacing for users

* **Command-line Interface**: In a command-line interface, users interact with the computer through commands. Command line interface is difficult to work with as it requires knowledge of command language.
* **Graphical User Interface (GUI):** As a graphical user interface does not need you to memorize the command language, it is easier to understand. Users interact with the computer through icons and menus.

### 1.2.4 Contrast: System software vs application software

Table 1.2.1 compares system software and application software.

Table 1.2. 1:Comparison of system software and application software

|  |  |  |
| --- | --- | --- |
| System Software | | Application Software |
| System software is mainly designed for managing system resources. | Application software is designed to accomplish specific tasks. | |
| Programming of system software is complex. | Programming of application software is comparatively easy. | |
| A computer can not run without system software. | A computer can run without application software | |
| System software do not depend on application software | Application software depend on system software and can not run without system software. | |
| System software are typically written or programmed in a low-level language such as machine language or assembly language. | Application software are mostly written in high level languages such as Java, C++ etc. | |

Figure 1.2.3 illustrates an overview of computer hardware and software.

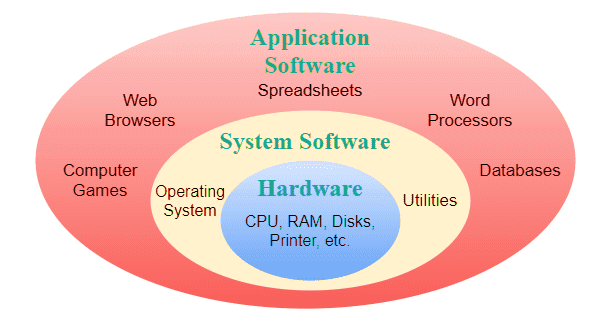
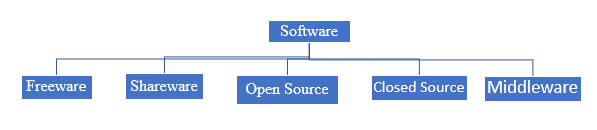


Figure 1.2. 3: Overview of computer

* + 1. Differentiate between: Shareware, Freeware, Open-Source Software and Proprietary software, Firmware

Computer software can also be classified according to availability and shareability and the categories are freeware, shareware, open source closed source and midlleware.



**Freeware**

Freeware is the software that is available to use for free of cost without any limitations. Example- Some of the google software is freeware.

**Shareware**

The software is copyrighted and distributed for free only for testing purposes. After the trial period ends, you must pay. For example, Microsoft Office (if you download MS-Office from the Microsoft website they give a one-month free trial after that you must pay for it yearly or monthly).

**Open Source**

This is provided for use, modification, and redistribution. Open-source software is downloaded from the internet at no cost. Example- Google play store apps, etc.

**Closed source software**

Closed source software is software for which the source code is not freely available.  It is developed and provided to the user as a fully compiled, executable set of files. The developer often provides support to users after purchase and ensures that the software works as expected. This is usually developed if you can not find the specific software you want off the shelf and you approach software development houses to custom make for your specific needs.

**Middleware**

Middleware is software that bridges gaps between other applications, tools, and databases to provide unified services to users. It is commonly characterized as the glue that connects different software platforms and devices together.

**Proprietary**

Proprietary software is computer software where the source codes are publicly not available only the company that has created can modify it. A good example is Microsoft products where users must get a licence key to install them on to their devices.

**Firmware**

Firmware is a type of software that is embedded directly in a piece of hardware to make the hardware work as intended. Firmware is programmed by the manufacturer and is installed on a digital device right in the factory. All computing devices have firmware.

### 1.2.4 Discuss the process of how software is obtained and installed

Computer software can be installed from a compact disk or any other portable devices. Software which comes loaded on a compact disc is autoplay allowing to start the installation process immediately. If not, you will need to load the storage device and locate the installation file with .exe extension. Another alternative would be to install from the command line. In this case, you will need to navigate to the folder containing the installation file and run the executable setup file. Many times, this can be done by typing setup or install at the prompt to start the installation.

Today, the most common way to get new software is to download it from the Internet. software development houses are allowing customers to access software from their cloud storage after payment. Once a payment is received the user will receive a key which allows them to install the software. Applications like Microsoft Office and Adobe Photoshop can now be purchased and downloaded right to your computer. You can also install free software this way. For example, if you wanted to install the Google Chrome web browser you just download it from their server and install it to your personal computer. The installation file will be saved to your computer in .exe format.

### Differentiate between online software and installed software.

**Installed software** is also referred to as desktop software. Installed software must be purchased in physical form (such as a CD) or downloaded from the internet. In either case, the program is installed using an installation program. Once the software is installed, it is ready to use.

**Benefits of installed software**

1. Installed software can be viewed when offline
2. Installed software is accessible on any device irrespective of connectivity
3. Installed software offers a better User eXperience
4. User has full control of the data

**Online software** is any program accessed over an Internet connection using a Web browser. The website pages act as a user interface. Online software run on a secure, external server, which also stores any data collected.

**Benefits of online software**

1. No software to install
2. Complete flexibility on device type
3. No maintenance required.
4. Connect from anywhere at any time with internet
5. Centralised storage offering safe and secure data

### Discuss the following terms in relation to software. (Range: Compatibility issues, Versions, patches and service packs, Updating software)

**VOCABULARY**

Compatibility is the capacity for two systems to work together without having to be altered to do so.

Compatibility issues come up when users are using the same type of software for a task, such as word processors, that cannot communicate with each other. This could be due to a difference in their versions or because they are made by different companies. There are two types of compatibility: forward compatibility and backward compatibility.

Products that are designed to be compatible with future versions of themselves are referred to as **forward compatible**. Products designed for compatibility with older versions are said to be [**backward compatible**](https://www.techtarget.com/whatis/definition/backward-compatible-backward-compatibility).

Any application which can allow users to revert or rollback to the previous version is said to be backward compatibility.

**Software Version** means a unique build of the software released by a company identified by the alpha-numerical sequence assigned to it together with a corresponding release date e.g. [version 5](https://www.lawinsider.com/dictionary/software-version).9.1 released on 13 October 2022.

**Software Patches**

A software patch is a program that makes changes to software installed on a computer. Software companies issue patches to fix bugs or security problems in their programs or add new functions to the software.

**Service Packs**

A service pack (SP) is a collection of updates and fixes for an operating system or a software program. SPs are usually made up of patches. Many of these patches are often released before a larger SP, but the SP allows for an easy, single installation.

Both system software and application software serve to facilitate user interaction with computer systems and the accomplishment of various activities. There are variations in their designs and intended uses, though. Application software is created to address specific user demands in order to carry out specific tasks, as opposed to system software, which is created to manage system resources or processes.

## 1.3 The Linux shell

The shell, to put it simply, is a program that receives keyboard commands and sends them to the operating system for processing. A program called bash serves as the shell on the majority of Linux computers. You can communicate with the shell by using the terminal emulator, which launches a window. Linux employs a wide range of different terminal emulators.  Among them are gnome-terminal, konsole, xterm, rxvt, kvt, nxterm, and eterm, among others. On Raspberry Pi, terminal we can run Linux commands.

### 1.3.1 Launch a new Linux terminal on the Raspberry Pi

To launch the terminal on Raspberry Pi, you will need to navigate to the Raspberry Icon on the Desktop. Click Accessories and select terminal. Figure 1.3.1 illustrates how the Raspberry terminal looks like.

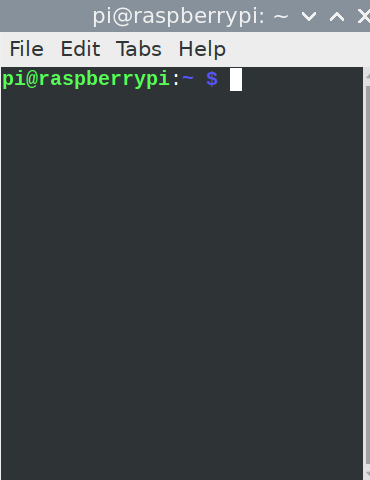


Figure 1.3. 1:Raspberry Terminal

### 1.3.2 Use the man command to get help

**man** is the system's manual pager. Each page argument given to man is normally the name of a program, utility or function. In Linux, man is an interface to view the system's reference manual. A user can request to display a man page by simply typing man followed by a space and then argument. Here its argument can be a command, utility or function. A manual page associated with each of these arguments is displayed. The syntax of man command is as follows:

$ man -options argument

Figure

For example:

man - -usage

Figure 1.3.2 shows the output of man –usage command to print usage syntax.

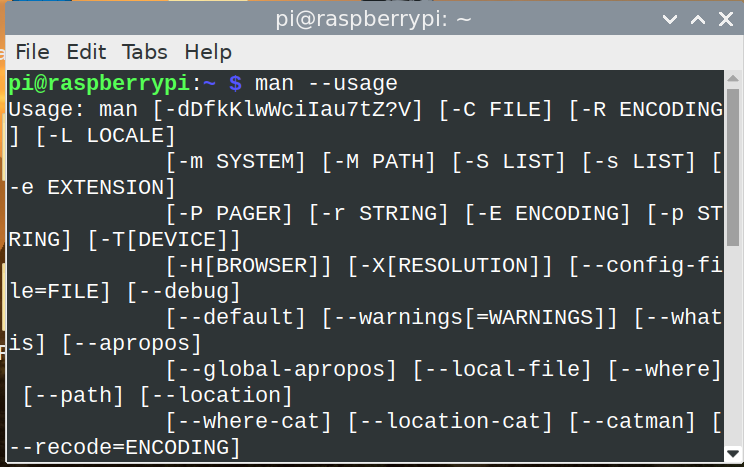


Figure 1.3. 2: Using man command

Another example could be bringing up the help by using man -h command as illustrated in Figure 1.3.3.

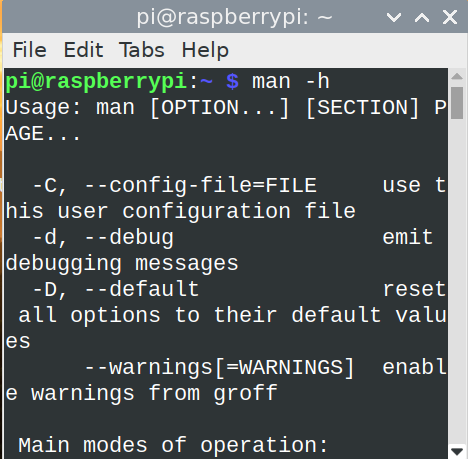


Figure 1.3. 3: using man -h command

So, to avoid buying a lot of books and saving trees- man is your answer. It provides the documentation at your fingertips.

### 1.3.3 Expand a Linux file path and explain each element

A file path is the human-readable representation of a file or folder’s location on a computer system. Files and folders on Linux are given names containing the usual components like the letters, numbers, and other characters on a keyboard. But when a file is inside a folder, or a folder is inside another folder, the / character shows the relationship between them. Figure 1.3.4 illustrates the file structure of the file add2.py. Just like in Windows where the file structure starts with C: on Raspberry Pi home is the root directory. If for and instance you want to see the exact file path, navigate to it on the terminal and type in the command: readlink -f <filename> as shown in Figure 1.3.4, we managed to find out the full path of the file called add2.py which is in /home/pi/Desktop folder

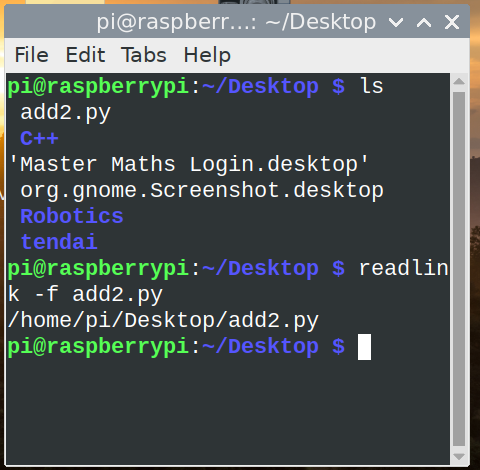


Figure 1.3. 4: File structure

### 1.3.4 List the contents of the current folder using the ls command

To list contents of the directory, the ls commands does the trick. See Figure 1.3.4

### 1.3.5 Change directly location using the cd command

To navigate from one directory to another, Linux uses cd (change directory) just like in Windows. Figure 1.3.5 illustrates how we used the cd command to navigate from Desktop folder to C++ folder.

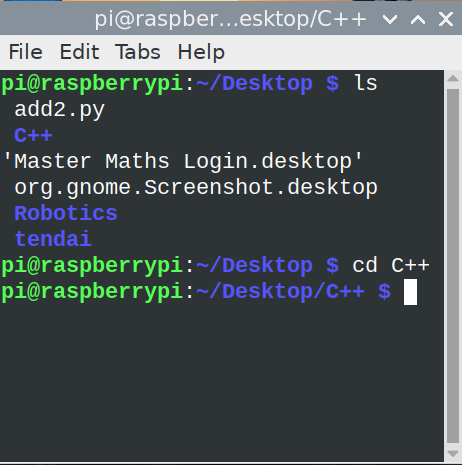


Figure 1.3. 5: Using cd command

If you are in a specific directory, you can navigate straight to the root directory by typing cd /. cd .. will make you move one level less from the current directory.

### 1.3.6 Create a new folder using the mkdir command

**mkdir** command in Linux allows the user to create directories. The user running this command must have sufficient permissions to create a directory in the parent directory; otherwise, a "permission denied" warning may appear.

**Syntax**

mkdir [options...] [directories ...]

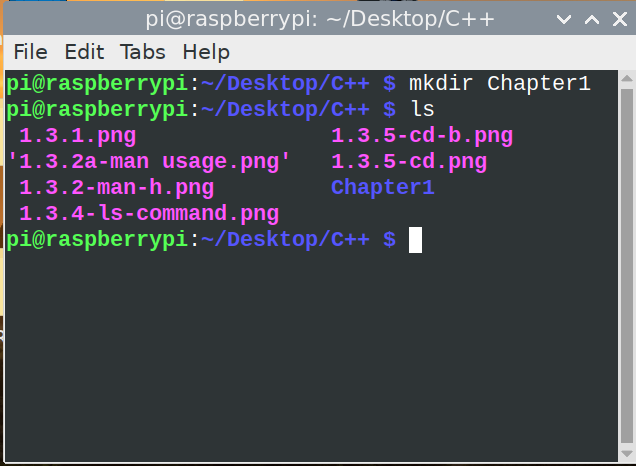


Figure 1.3. 6: Using mkdir command

In Figure 1.3.6, we managed to create a directory called Chapter1 inside the C++ folder. If you want to create two folders at once, just type the mkdir command and folder names and leave space between the names. There are other options which can be used together with mkdir command such as:

**- -v or - -verbose**: It displays a message for every directory created.   
**- -help**: It displays the help related information and exits.

### 1.3.7 Remove a folder using the rmdir command

rmdir command is used remove empty directories from the filesystem in Linux. The rmdir command removes each directory specified in the command line only if these directories are empty. Figure 1.3.7 shows how we removed the directory called Chapter1 from the C++ directory. After listing all files using the ls command, we can see that the directory is not there anymore.

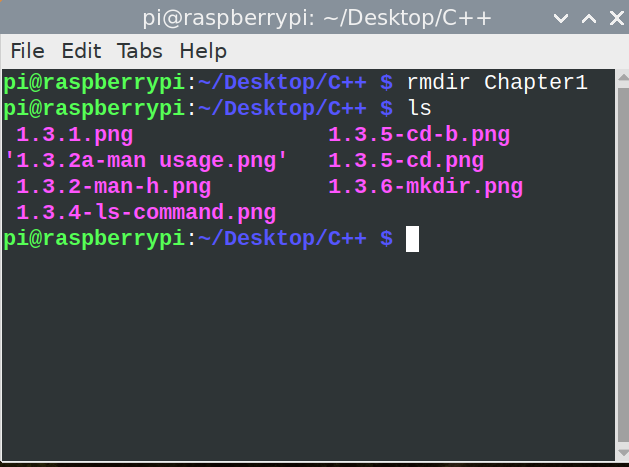


Figure 1.3. 7: Using rmdir command

### 1.3.8 Remove a file using the rm command

The rm command removes the entries for a specified file, group of files, or certain selected files from a list within a directory. When you use the rm command, user confirmation, read permission, and write permission are not required before a file is removed. In Figure 1.3.8, we demonstrate how to remove the file called newfile.py from C++ directory. We also demonstrated how to create a new file using touch command.

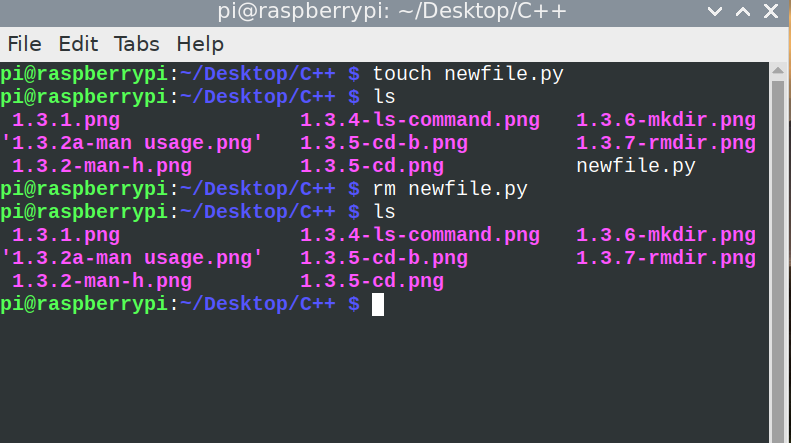


Figure 1.3. 8: Using rm command

### 1.3.9 Rename a file using the mv command

Use the mv command to change the name of a file without moving it to another directory. To rename a file, type the following:

$ mv <currentfilename> <proposedfilename>

Figure 1.3.9 illustrates how we renamed the file called newfile.py to newfile1.py by executing the following command:

$ mv newfile.py newfile1.py

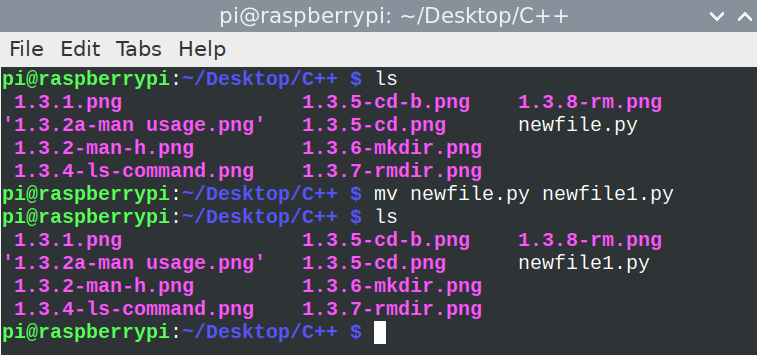


Figure 1.3. 9:Changing filename using mv command

### 1.3.10 Copy a file using the cp command

Use the [cp](https://www.ibm.com/docs/en/ssw_aix_72/c_commands/cp.html) command to create a copy of the contents of the file or directory specified by the sourcefile or sourcedirectory parameters into the file or directory specified by the TargetFile or TargetDirectory parameters. If the file specified as the TargetFile exists, the copy writes over the original contents of the file without warning. If you are copying more than one SourceFile, the target must be a directory. Figure 1.3.10 demostrates the cp command used to copy newfile.py to C++ folder.

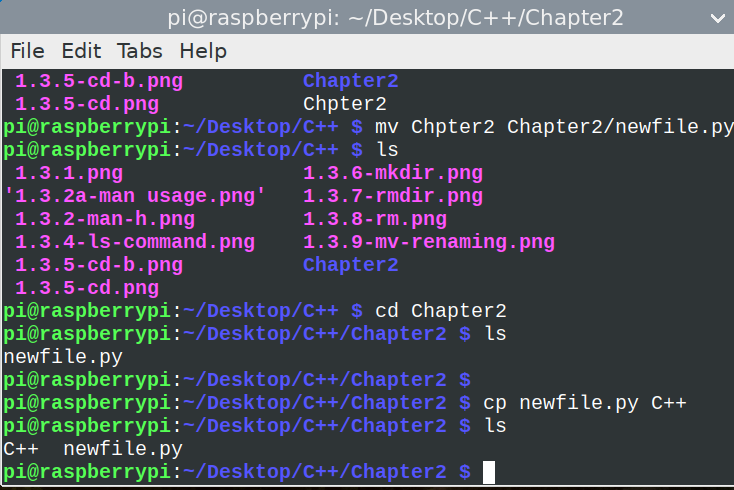


Figure 1.3. 10: Using cp command

### 1.3.11 Clear the command prompts screen using the cls command

clear clears your screen the text on the terminal. It looks in the environment for the terminal type and then in the terminfo database to figure out how to clear the screen. clear ignores any command-line parameters that may be present.

### Run an executable file from the command line

To run an executable file the user rights of those files must be set correct. This can be done by doing the following:

Syntax

chmod +x <filename>.

chmod +x sample.sh.

There are more Linux commands to perform different operations. The link below will give a guide of other Linux commands.

**e-Link**

<https://www.guru99.com/linux-commands-cheat-sheet.html>

# Problem solving in computer programming

## 2.1 Problem solving process and concepts

Content:

* Problem solving
* Application of problem-solving constructs
* Developing solutions

*Learning Outcomes:*

*Students should be able to:*

### 2.1.1 Define the term problem solving

### 2.1.2 Define the term computational thinking

### 2.1.3 Describe the phases of the PLDC (Program Development Life Cycle)

### 2.1.4 Describe the purpose of problem solving leading to solutions

2.1.5 Explain and apply various problem-solving steps. Polya, G., 1957) (Range:Understand the problem (task/problem description or scenario/user stories) State in own words Clarity on what needs to be done

What is known or given?

What is missing or needed?

Devise a plan/algorithm (storyboard – visual or textual)

Look for patterns

Look at related problems, known solutions

Examine simpler or special cases

Make a table, create diagram, use guess and check, work backwards, identify sub-goal

Carry out the plan/implement the algorithm (write the code)

Look back/test (see if it works)

Check results against original problem. Does it make sense? Is there another solution?)

### 2.1.6 Use appropriate tools and techniques to present a solution. Range:

User stories (written by the client and provide the requirements)

Noun-verb analysis of user stories

List of nouns provides identification of objects and state

List of verbs provides identification of behaviour

Acceptance tests (does the program meet the requirements?)

## 2.2 Construct an algorithm and present a solution to a given problem

Content:

* Problem solving
* Algorithm design
* Flowcharts

*Learning Outcomes:*

*Students should be able to:*

2.2.1 Define the term algorithm and its purpose in the problem-solving process. (Range: Basic concepts of an algorithm. What is an algorithm? Develop a clear understanding of the problem presented.)

2.2.2 Implement and understand the basic algorithmic constructs used to create a flowchart**.** Range: Input, Output, Processing and Calculations, Selection Iteration

### 2.2.3 Create a flowchart to present a particular algorithm and its associated tasks

### 2.2.4 Interpret a basic flow chart and describe its intended operation / function