INDEX

TOPICS

- **Certificates……**
- **Acknowledgement…….....**

CHAPTER 1: INTRODUCTION

1.1 Introduction of the project ……

1.2 Project overview……...

1.3 Thesis……

CHAPTER 2: EMBEDDED SYSTEMS

2.1 Introduction to embedded systems……

2.2 Need of embedded systems……...

2.3 Explanation of embedded systems……...

2.4 Applications of embedded systems……

CHAPTER 3: HARDWARE DESCRIPTION

3.1 Introduction with block diagram……

3.2 ARM-11 processor…...…….

3.3 Adapter……...…..……..……...

3.4 Wi-Fi ……..……

3.5 USB camera……..…..……..……...

3.6 Servomotor……..…..……..……...

3.7 Bluetooth module……..…..……..……

3.8 DC motors……

CHAPTER 4: SOFTWARE DESCRIPTION

4.1 Express PCB……

4.2 Linux operating system…….

4.3 Proteus software……

4.4 Procedural steps for compilation, simulation and dumping……..

CHAPTER 5: PROJECT DESCRIPTION

CHAPTER 6: ADVANTAGES, DISADVANTAGES AND APPLICATIONS

CHAPTER 7: RESULTS, CONCLUSION, FUTURE PROSPECTS

REFERENCES

CHAPTER 1: INTRODUCTION

1.1 Introduction:

The main of the project to design a robot for virtual telepresence using Raspberry pi. In project the smart phone reads the accelerometer and magnetometer data of the direction in which the user turns his head, say, right or left. This data is sent to the modem over Wi-Fi and to the Raspberry Pi board, which, in turn, provides these values as inputs to the servo motors.

The Raspberry Pi is a low cost, **credit-card sized computer** that can be used in electronics projects. It is a capable little device that enables people of all ages to explore computing, and to learn how to program in languages like Scratch and Python. It's capable of doing everything you'd expect a desktop computer to do, from browsing the internet and playing high-definition video, to making spreadsheets, word-processing, and playing games.

Arduino is an open source, computer hardware and software company, project, and user community that designs and manufactures microcontroller kits for building digital devices and interactive objects that can sense and control objects in the physical world

The task of the project is achieved by using Raspberry pi. This robot with a Pi camera is placed in a remote location to capture the environment in visual form using Raspberry Pi. The captured visuals are displayed on the user's virtual reality (VR) headset along with obstacle detection using image processing. The robot can also be moved in any direction (in which the user turns his head, say, right or left) through an app installed in the user's smartphone.

Two servo motors are used to move the camera—one for the vertical movement and the other for the horizontal movement. So, when you turn your head along with VR headset to the right side, the Raspberry Pi camera will also turn to the right direction. The smartphone also provides input to the Arduino Mega (Board2) for the purpose of navigation or movement of the robot. The motor driver IC and geared motors are connected at the end of the navigation circuit. The commands to run the robot can be sent via Bluetooth from the smartphone. In this example, Bluetooth is used.

The main objectives of the project:

1. To design a robot for virtual telepresence.
2. Using Raspberry pi to achieve the task.

3. To send data to the modem using Wi-Fi.
4. This robot can send the live visuals along with obstacle detection to the VR headset.
5. To receives input from the smartphone via Wi-Fi, and then sends controlling pulse to the servo motors to move the Raspberry Pi camera.

1.2 Project Overview:

An embedded system is a combination of software and hardware to perform a dedicated task.Some of the main devices used in embedded products are Microprocessors and Microcontrollers. Microprocessors are commonly referred to as general purpose processors as they simply accept the inputs, process it and give the output.In contrast, a microcontroller not only accepts the data as inputs but also manipulates it, interfaces the data with various devices, controls the data and thus finally gives the result.

The project "**Virtual Telepresence Robot Using Raspberry Pi**" is an exclusive in project which the smartphone reads the accelerometer and magnetometer data of the direction in which the user turns his head, say, right or left. This data is sent to the modem over Wi-Fi and to the Raspberry Pi board, which, in turn, provides these values as inputs to the servo motors. This robot will send the visual data along with obstacle detection to the user head set though wi-fi.

1.3 Thesis:

The thesis explains the implementation of "**Virtual Telepresence Robot Using Raspberry Pi"**. The organization of the thesis is explained here with:

Chapter 1 Presents introduction to the overall thesis and the overview of the project. In the project overview, a brief introduction of **Virtual Telepresence Robot Using Raspberry Pi** is discussed.

Chapter 2 Presents the topic embedded systems. It explains the about what is embedded systems, need for embedded systems, explanation of it along with its applications.

Chapter 3 Presents the hardware description. It deals with the block diagram of the project and explains the purpose of each block. In the same chapter the explanation of Pi camera, Raspberry Pi processor, DC motors, Bluetooth module and Servomotor is given.

Chapter 4 Presents the software description. It explains the implementation of the project using Linux operating system.

Chapter 5 Presents the project description along with Pi camera, Bluetooth module, DC motors, VR head set and Servomotor interfacing to Raspberry Pi

Chapter 6 Presents the results, conclusion and future scope of the project.

CHAPTER 2: EMBEDDED SYSTEMS

2.1 Embedded Systems:

An embedded system is a computer system designed to perform one or a few dedicated functions often with real-time computing constraints. It is embedded as part of a complete device often including hardware and mechanical parts. By contrast, a general-purpose computer, such as a personal computer (PC), is designed to be flexible and to meet a wide range of end-user needs. Embedded systems control many devices in common use today.

Embedded systems are controlled by one or more main processing cores that are typically either microcontrollers or digital signal processors (DSP). The key characteristic, however, is being dedicated to handle a particular task, which may require very powerful processors. For example, air traffic control systems may usefully be viewed as embedded, even though they involve mainframe computers and dedicated regional and national networks between airports and radar sites. (Each radar probably includes one or more embedded systems of its own.)

Since the embedded system is dedicated to specific tasks, design engineers can optimize it to reduce the size and cost of the product and increase the reliability and performance. Some embedded systems are mass-produced, benefiting from economies of scale.

Physically embedded systems range from portable devices such as digital watches and MP3 players, to large stationary installations like traffic lights, factory controllers, or the systems controlling nuclear power plants. Complexity varies from low, with a single microcontroller chip, to very high with multiple units, peripherals and networks mounted inside a large chassis or enclosure.

In general, "embedded system" is not a strictly definable term, as most systems have some element of extensibility or programmability. For example, handheld computers share some elements with embedded systems such as the operating systems and microprocessors which power them, but they allow different applications to be loaded and peripherals to be connected. Moreover, even systems which don't expose programmability as a primary feature generally need to support software updates. On a continuum from "general purpose" to "embedded", large application systems will have subcomponents at most points even if the system as a whole is "designed to perform one or a few dedicated functions", and is thus appropriate to call "embedded". A modern example of embedded system is shown in fig: 2.1.

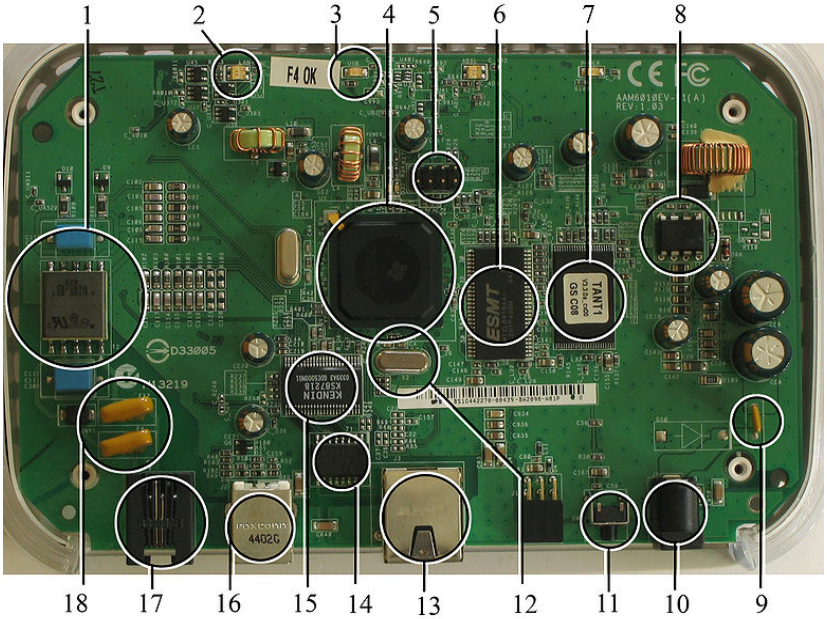


Fig 2.1:A modern example of embedded system

Labeled parts include microprocessor (4), RAM (6), flash memory (7).Embedded systems programming is not like normal PC programming. In many ways, programming for an embedded system is like programming PC 15 years ago. The hardware for the system is usually chosen to make the device as cheap as possible. Spending an extra dollar a unit in order to make things easier to program can cost millions. Hiring a programmer for an extra month is cheap in

comparison. This means the programmer must make do with slow processors and low memory, while at the same time battling a need for efficiency not seen in most PC applications. Below is a list of issues specific to the embedded field.

2.1.1 History:

In the earliest years of computers in the 1930–40s, computers were sometimes dedicated to a single task, but were far too large and expensive for most kinds of tasks performed by embedded computers of today. Over time however, the concept of programmable controllers evolved from traditional electromechanical sequencers, via solid state devices, to the use of computer technology.

One of the first recognizably modern embedded systems was the Apollo Guidance Computer, developed by Charles Stark Draper at the MIT Instrumentation Laboratory. At the project's inception, the Apollo guidance computer was considered the riskiest item in the Apollo project as it employed the then newly developed monolithic integrated circuits to reduce the size and weight. An early mass-produced embedded system was the Autonetics D-17 guidance computer for the Minuteman missile, released in 1961. It was built from transistor logic and had ahard disk for main memory. When the Minuteman II went into production in 1966, the D-17 was replaced with a new computer that was the first high-volume use of integrated circuits.

2.1.2 Tools:

Embedded development makes up a small fraction of total programming. There's also a large number of embedded architectures, unlike the PC world where 1 instruction set rules, and the Unix world where there's only 3 or 4 major ones. This means that the tools are more expensive. It also means that they're lowering featured, and less developed. On a major embedded project, at some point you will almost always find a compiler bug of some sort.

Debugging tools are another issue. Since you can't always run general programs on your embedded processor, you can't always run a debugger on it. This makes fixing your program difficult. Special hardware such as JTAG ports can overcome this issue in part. However, if you stop on a breakpoint when your system is controlling real world hardware (such as a motor), permanent equipment damage can occur. As a result, people doing embedded programming quickly become masters at using serial IO channels and error message style debugging.

2.1.3 Resources:

To save costs, embedded systems frequently have the cheapest processors that can do the job. This means your programs need to be written as efficiently as possible. When dealing with large data sets, issues like memory cache misses that never matter in PC programming can hurt you. Luckily, this won't happen too often- use reasonably efficient algorithms to start, and optimize only when necessary. Of course, normal profilers won't work well, due to the same reason debuggers don't work well.

Memory is also an issue. For the same cost savings reasons, embedded systems usually have the least memory they can get away with. That means their algorithms must be memory efficient (unlike in PC programs, you will frequently sacrifice processor time for memory, rather than the reverse). It also means you can't afford to leak memory. Embedded applications generally use deterministic memory techniques and avoid the default "new" and "malloc" functions, so that leaks can be found and eliminated more easily.Other resources programmers expect may not even exist. For example, most embedded processors do not have hardware FPUs (Floating-Point Processing Unit). These resources either need to be emulated in software, or avoided altogether.

2.1.4 Real Time Issues:

Embedded systems frequently control hardware, and must be able to respond to them in real time. Failure to do so could cause inaccuracy in measurements, or even damage hardware such as motors. This is made even more difficult by the lack of resources available. Almost all embedded systems need to be able to prioritize some tasks over others, and to be able to put off/skip low priority tasks such as UI in favor of high priority tasks like hardware control.

2.2 Need For Embedded Systems:

The uses of embedded systems are virtually limitless, because every day new products are introduced to the market that utilizes embedded computers in novel ways. In recent years, hardware such as microprocessors, microcontrollers, and FPGA chips have become much cheaper. So when implementing a new form of control, it's wiser to just buy the generic chip and write your own custom software for it. Producing a custom-made chip to handle a particular task or set of tasks costs far more time and money. Many embedded computers even come with extensive libraries, so that "writing your own software" becomes a very trivial task indeed.From an implementation viewpoint, there is a major difference between a computer and an embedded system. Embedded systems are often required to provide Real-Time response. The main elements that make embedded systems unique are its reliability and ease in debugging.

2.2.1 Debugging:

Embedded debugging may be performed at different levels, depending on the facilities available. From simplest to most sophisticate they can be roughly grouped into the following areas:

- Interactive resident debugging, using the simple shell provided by the embedded operating system (e.g. Forth and Basic)
- External debugging using logging or serial port output to trace operation using either a monitor in flash or using a debug server like the Remedy Debugger which even works for heterogeneous multicore systems.
- An in-circuit debugger (ICD), a hardware device that connects to the microprocessor via a JTAG or Nexus interface. This allows the operation of the microprocessor to be controlled externally, but is typically restricted to specific debugging capabilities in the processor.
- An in-circuit emulator replaces the microprocessor with a simulated equivalent, providing full control over all aspects of the microprocessor.
- A complete emulator provides a simulation of all aspects of the hardware, allowing all of it to be controlled and modified and allowing debugging on a normal PC.
- Unless restricted to external debugging, the programmer can typically load and run software through the tools, view the code running in the processor, and start or stop its operation. The view of the code may be as assembly code or source-code.

Because an embedded system is often composed of a wide variety of elements, the debugging strategy may vary. For instance, debugging a software(and microprocessor) centric embedded system is different from debugging an embedded system where most of the processing is performed by peripherals (DSP, FPGA, co-processor). An increasing number of embedded systems today use more than one single processor core. A common problem with multi-core development is the proper synchronization of software execution. In such a case, the embedded system design may wish to check the data traffic on the busses between the processor cores, which requires very low-level debugging, at signal/bus level, with a logic analyzer, for instance.

2.2.2 Reliability:

Embedded systems often reside in machines that are expected to run continuously for years without errors and in some cases recover by them if an error occurs. Therefore the software is usually developed and tested more carefully than that for personal computers, and unreliable mechanical moving parts such as disk drives, switches or buttons are avoided.

Specific reliability issues may include:

- The system cannot safely be shut down for repair, or it is too inaccessible to repair. Examples include space systems, undersea cables, navigational beacons, bore-hole systems, and automobiles.
- The system must be kept running for safety reasons. "Limp modes" are less tolerable. Often backups are selected by an operator. Examples include aircraft navigation, reactor control systems, safety-critical chemical factory controls, train signals, engines on single-engine aircraft.
- The system will lose large amounts of money when shut down: Telephone switches, factory controls, bridge and elevator controls, funds transfer and market making, automated sales and service.

A variety of techniques are used, sometimes in combination, to recover from errors—both software bugs such as memory leaks, and also soft errors in the hardware:

- Watchdog timer that resets the computer unless the software periodically notifies the watchdog
- Subsystems with redundant spares that can be switched over to
- software "limp modes" that provide partial function
- Designing with a Trusted Computing Base (TCB) architecture[6] ensures a highly secure & reliable system environment
- An Embedded Hypervisor is able to provide secure encapsulation for any subsystem component, so that a compromised software component cannot interfere with other subsystems, or privileged-level system software. This encapsulation keeps faults from propagating from one subsystem to another, improving reliability. This may also allow a subsystem to be automatically shut down and restarted on fault detection.
- Immunity Aware Programming

2.3 Explanation of Embedded Systems:

2.3.1 Software Architecture:

There are several different types of software architecture in common use.

- Simple Control Loop:

In this design, the software simply has a loop. The loop calls subroutines, each of which manages a part of the hardware or software.

- Interrupt Controlled System:

Some embedded systems are predominantly interrupt controlled. This means that tasks performed by the system are triggered by different kinds of events. An interrupt could be generated for example by a timer in a predefined frequency, or by a serial port controller receiving a byte. These kinds of systems are used if event handlers need low latency and the event handlers are short and simple.

Usually these kinds of systems run a simple task in a main loop also, but this task is not very sensitive to unexpected delays. Sometimes the interrupt handler will add longer tasks to a queue structure. Later, after the interrupt handler has finished, these tasks are executed by the main loop. This method brings the system close to a multitasking kernel with discrete processes.

- Cooperative Multitasking:

A non-preemptive multitasking system is very similar to the simple control loop scheme, except that the loop is hidden in an API. The programmer defines a series of tasks, and each task gets its own environment to “run” in. When a task is idle, it calls an idle routine, usually called “pause”, “wait”, “yield”, “nop” (stands for no operation), etc.The advantages and disadvantages are very similar to the control loop, except that adding new software is easier, by simply writing a new task, or adding to the queue-interpreter.

- Primitive Multitasking:

In this type of system, a low-level piece of code switches between tasks or threads based on a timer (connected to an interrupt). This is the level at which the system is generally considered to have an "operating system" kernel. Depending on how much functionality is required, it introduces more or less of the complexities of managing multiple tasks running conceptually in parallel.

As any code can potentially damage the data of another task (except in larger systems using an MMU) programs must be carefully designed and tested, and access to shared data must be controlled by some synchronization strategy, such as message queues, semaphores or a non-blocking synchronization scheme.

Because of these complexities, it is common for organizations to buy a real-time operating system, allowing the application programmers to concentrate on device functionality rather than operating system services, at least for large systems; smaller systems often cannot afford the overhead associated with a generic real time system, due to limitations regarding memory size, performance, and/or battery life.

- Microkernels And Exokernels:

A microkernel is a logical step up from a real-time OS. The usual arrangement is that the operating system kernel allocates memory and switches the CPU to different threads of execution. User mode processes implement major functions such as file systems, network interfaces, etc.

In general, microkernels succeed when the task switching and intertask communication is fast, and fail when they are slow. Exokernels communicate efficiently by normal subroutine calls. The hardware and all the software in the system are available to, and extensible by application programmers. Based on performance, functionality, requirement the embedded systems are divided into three categories:

2.3.2 Stand Alone Embedded System:

These systems takes the input in the form of electrical signals from transducers or commands from human beings such as pressing of a button etc.., process them and produces desired output. This entire process of taking input, processing it and giving output is done in standalone mode. Such embedded systems comes under stand alone embedded systems

Eg: microwave oven, air conditioner etc...

2.3.3 Real-time embedded systems:

Embedded systems which are used to perform a specific task or operation in a specific time period those systems are called as real-time embedded systems. There are two types of real-time embedded systems.

- Hard Real-time embedded systems:

These embedded systems follow an absolute dead line time period i.e.., if the tasking is not done in a particular time period then there is a cause of damage to the entire equipment.

Eg: consider a system in which we have to open a valve within 30 milliseconds. If this valve is not opened in 30 ms this may cause damage to the entire equipment. So in such cases we use embedded systems for doing automatic operations.

- Soft Real Time embedded systems:

These embedded systems follow a relative dead line time period i.e.., if the task is not done in a particular time that will not cause damage to the equipment.

Eg: Consider a TV remote control system, if the remote control takes a few milliseconds delay it will not cause damage either to the TV or to the remote control. These systems which will not cause damage when they are not operated at considerable time period those systems comes under soft real-time embedded systems.

2.3.4 Network communication embedded systems:

A wide range network interfacing communication is provided by using embedded systems.

Eg:

- Consider a web camera that is connected to the computer with internet can be used to spread communication like sending pictures, images, videos etc.., to another computer with internet connection throughout anywhere in the world.
- Consider a web camera that is connected at the door lock.

Whenever a person comes near the door, it captures the image of a person and sends to the desktop of your computer which is connected to internet. This gives an alerting message with image on to the desktop of your computer, and then you can open the door lock just by clicking the mouse. Fig: 2.2 show the network communications in embedded systems.

Fig 2.2: Network communication embedded systems

2.3.5 Different types of processing units:

The central processing unit (c.p.u) can be any one of the following microprocessor, microcontroller, digital signal processing.

- Among these Microcontroller is of low cost processor and one of the main advantage of microcontrollers is, the components such as memory, serial communication interfaces, analog to digital converters etc.., all these are built on a single chip. The numbers of external components that are connected to it are very less according to the application.
- Microprocessors are more powerful than microcontrollers. They are used in major applications with a number of tasking requirements. But the microprocessor requires many external components like memory, serial communication, hard disk, input output ports etc.., so the power consumption is also very high when compared to microcontrollers.
- Digital signal processing is used mainly for the applications that particularly involved with processing of signals

2.4 APPLICATIONS OF EMBEDDED SYSTEMS:

2.4.1 Consumer applications:

At home we use a number of embedded systems which include microwave oven, remote control, vcd players, dvd players, camera etc….

Fig2.3: Automatic coffee makes equipment

2.4.2 Office automation:

We use systems like fax machine, modem, printer etc…

Fig2.4: Fax machine **Fig2.5: Printing machine**

2.4.3. Industrial automation:

Today a lot of industries are using embedded systems for process control. In industries we design the embedded systems to perform a specific operation like monitoring temperature, pressure,humidity ,voltage, current etc.., and basing on these monitored levels we do control other devices, we can send information to a centralized monitoring station.

Fig2.6: Robot

In critical industries where human presence is avoided there we can use robot**s** which are programmed to do a specific operation.

2.4.5 Computer networking:

Embedded systems are used as bridges routers etc..

Fig2.7: Computer networking

2.4.6 Tele communications:

Cell phones, web cameras etc.

Fig2.8: Cell PhoneFig2.9: Web camera

CHAPTER 3: HARDWARE DESCRIPTION

3.1 Introduction:

In this chapter the block diagram of the project and design aspect of independent modules are considered. Block diagram is shown in fig: 3.1:

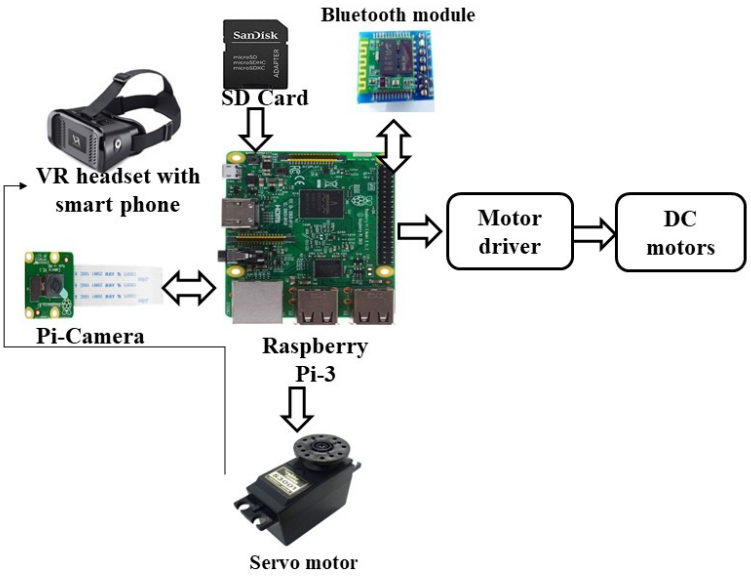


FIG 3.1: Virtual Telepresence Robot Using Raspberry Pi

The main blocks of this project are:

1. Power Supply.
2. Raspberry pi processor.
3. SD card.
4. Pi camera.
5. VR head set.
6. Servo motor.
7. Bluetooth module.
8. Smartphone with internet connection.

3.2 ARM-11 Processor (Raspberry Pi Processor)

Differences between microprocessor and microcontroller:

Microprocessor:

1. The c.p.u, memories, timers, Input/output ports, serial communication, interrupts etc.., all these are located on different chips and are externally connected to the processing unit.
2. Different chips occupy more space; more power consumption is required and more cost.
3. Microprocessors are used in products like general purpose computers.
4. Different kinds of software applications can be loaded and can be used simultaneously.
5. Multitasking can be done using microprocessor.
6. The memory size, number of ports etc.., can be configured according to the requirement in our application.
7. A large instruction set can be used for the applications using microprocessor.
8. The clock rates are faster when compared to micro controller.
   They are in gega hertz for microprocessor.

Microcontroller:

1. The central processing unit, serial communication, timers,

Memories, interrupts, input/output ports etc.., are equipped on the same single chip.

2. It occupies less space, so it consumes less power, and also the cost also very low when compared to microprocessor.

3. Used for products that performs only a specified task.

eg: air conditioner, microwave oven, remote controls etc..,

4. Only a single software application is generally used.

5. Using microcontroller only a specified task can be done basing on specified time periods.

6. The memory size, number of ports etc.., are very limited.

7. A compact instruction or a reduced instruction set is generally used for the applications when we use microcontroller.

8. The clock rates are slower when compared to microprocessor. They are in mega hertz.

ARM PROCESSOR

- ARM Processor was developed at Acorn computer limited of Cambridge, England between 1983 and 1985.
- This was after RISC concept came out at Stanford and Berkeley universities in 1980.
- ARM uses Enhanced RISC Architecture.
- ARM (Acorn RISC machine) limited was found in 1990.
- ARM designed basic core structure and licensed it to many partners who develop and fabricate new Micro Controllers and different chips.
- ARM processor is mainly intended in the development of embedded applications which involve complex computations (High-end applications).

ARM ARCHITECTURE:

- ARM architecture is based on Enhanced RISC architecture (deviates from classic RISC architecture).
- Embedded applications need to have :
    - High code density
    - Low power consumption rate
    - Small silicon foot print
- A large uniform register file (bank).
- Load-Store architecture, where data processing operations involve only registers but not memory locations.
- Uniform and Fixed length instructions.
- Good speed/power consumption ratio.
- High code density.

Enhancements to Classic RISC architecture:

- Control over ALU and Shifter (Barrel Shifter) which helps maximum usage of hardware on the chip.
- Auto increment and Auto decrement of addressing modes to optimize program loops.
- Load and Store multiple data elements through a single instruction, which increases data throughput.
- A lot of branch instructions which can be used in conjunction with a number of instructions, which maximizes execution throughput.

VERSIONS:

- V1 (1983-85) 26 bit addressing mode, no multiply or co-processor.
- V2 32 bit addressing mode, multiply, co-processor.
- V3 32 bit addressing.
- V4 Add signed and unsigned words.
- V4T Thumb instruction set (16-bit instruction set)
- V5T Superset of 4T with new instructions
- V5TE Add Signal processing

Examples:

ARM 6 V3

ARM 7 V3

ARM7TDMI V4T

Strong ARM V4

ARM 9E-S V5TE

ARM11 is a family of ARM architecture 32-bit RISC microprocessor cores.

Overview

Announced	
Year	**Core**
2002	ARM1136J(F)-S
2003	ARM1156T2(F)-S

2003	ARM1176JZ(F)-S

The ARM11 micro architecture (announced 29 April 2002) introduced the ARMv6 architectural additions which had been announced in October 2001. These include SIMD media instructions, multiprocessor support and a new cache architecture. The implementation included a significantly improved instruction processing pipeline, compared to previous ARM9 or ARM10 families, and is used in smartphones from Apple, Nokia, and others. The initial ARM11 core (ARM1136) was released to licensees in October 2002.

The ARM11 family is currently the only ARMv6-architecture cores. There are however ARMv6-M cores (Cortex-M0 and Cortex-M1), addressingmicrocontroller applications;[1] ARM11 cores target more demanding applications.

Differences from ARM9

In terms of instruction set, the ARM11 builds on the preceding ARM9 generation. It incorporates all ARM926EJ-S features and adds the ARMv6 instructions for media support (SIMD) and accelerating IRQ response.

Microarchitecture improvements in ARM11 cores[2] include:

- SIMD instructions which can double MPEG-4 and audio digital signal processing algorithm speed
- Cache is physically addressed, solving many cache aliasing problems and reducing context switch overhead.
- Unaligned and mixed-endian data access is supported.
- Reduced heat production and lower overheating risk
- Redesigned pipeline, supporting faster clock speeds (target up to 1 GHz)
  - Longer: 8 (vs 5) stages
  - Out-of-order completion for some operations (e.g. stores)
  - Dynamic branch prediction/folding (like XScale)
  - Cache misses don't block execution of non-dependent instructions.
  - Load/store parallelism
  - ALU parallelism

- 64-bit data paths

JTAG debug support (for halting, stepping, breakpoints, and watchpoints) was simplified. The EmbeddedICE module was replaced with an interface which became part of the ARMv7 architecture. The hardware tracing modules (ETM and ETB) are compatible, but updated, versions of those used in the ARM9. In particular, trace semantics were updated to address parallel instruction execution and data transfers.

ARM makes an effort to promote good Verilog coding styles and techniques. This ensures semantically rigorous designs, preserving identical semantics throughout the chip design flow, which included extensive use of formal verification techniques. Without such attention, integrating an ARM11 with third party designs could risk exposing hard-to-find latent bugs. Due to ARM cores being integrated into many different designs, using a variety of logic synthesis tools and chip manufacturing processes, the impact of its register-transfer level (RTL) quality is magnified many times.[3] The ARM11 generation focused more on synthesis than previous generations, making such concerns be more of an issue.

Cores

There are four ARM11 cores:

- ARM1136
- ARM1156, introduced Thumb2 instructions
- ARM1176, introduced security extensions
- ARM11MPcore, introduced multicore support

Chips

This list is incomplete; you can help by expanding it.

- Ambarella A5s, A7, A7L
- Broadcom BCM2835 (Raspberry Pi), BCM21553
- Cavium ECONA CNS3000 series [6]
- CSR Quatro 4230, Quatro 4500 series, Quatro 5300 series
- Freescale Semiconductor i.MX3x series, such as i.MX31, i.MX35
- Nintendo 1048 0H
- Infotmic IMAPX200, IMAPX210, IMAPX220

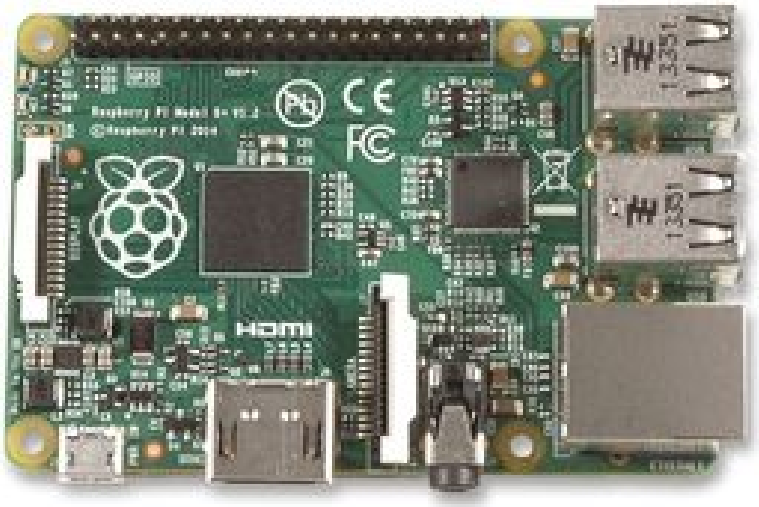
- Nvidia Tegra
- PLX Technology NAS7820, NAS7821, NAS7825
- MediaTek MTK6573_S01 / MTK6573_S00
- Qualcomm MSM720x, MSM7x27
- Qualcomm Atheros AR7400
- Samsung S3C64x0, S5P6422
- Telechips [tcc8902]
- Texas Instruments OMAP2 series, with a TMS320 C55x or C64x DSP as a second core

The **Raspberry Pi** is a credit-card-sized single-board computer developed in the UK by the Raspberry Pi Foundation with the intention of promoting the teaching of basic computer science in schools.

The Raspberry Pi is manufactured through licensed manufacturing deals with Newark element14 (Premier Farnell), RS Components and Egoman. All of these companies sell the Raspberry Pi online. Egoman produces a version for distribution solely in China and Taiwan, which can be distinguished from other Pi by their red coloring and lack of FCC/CE marks. The hardware is the same across all manufacturers.

Product Information

- RASPBERRY PI 3 BOARD
- Silicon Manufacturer: Broadcom
- Core Architecture: ARM
- Core Sub-Architecture: ARM1176JZF-S
- Silicon Core Number: BCM2837
- Silicon Family Name: BCM2xxx
- No. of Bits: 64 bit
- Kit Contents: Raspberry Pi 3
- Features: Credit card size computer, HDMI, Ethernet & 4 USB ports, 1GB RAM, Micro SD Socket, 40 GPIO



Raspberry Pi 3 Featuring the ARM1176JZF-S Running at 1.2 Ghz, with 1 GB of RAM

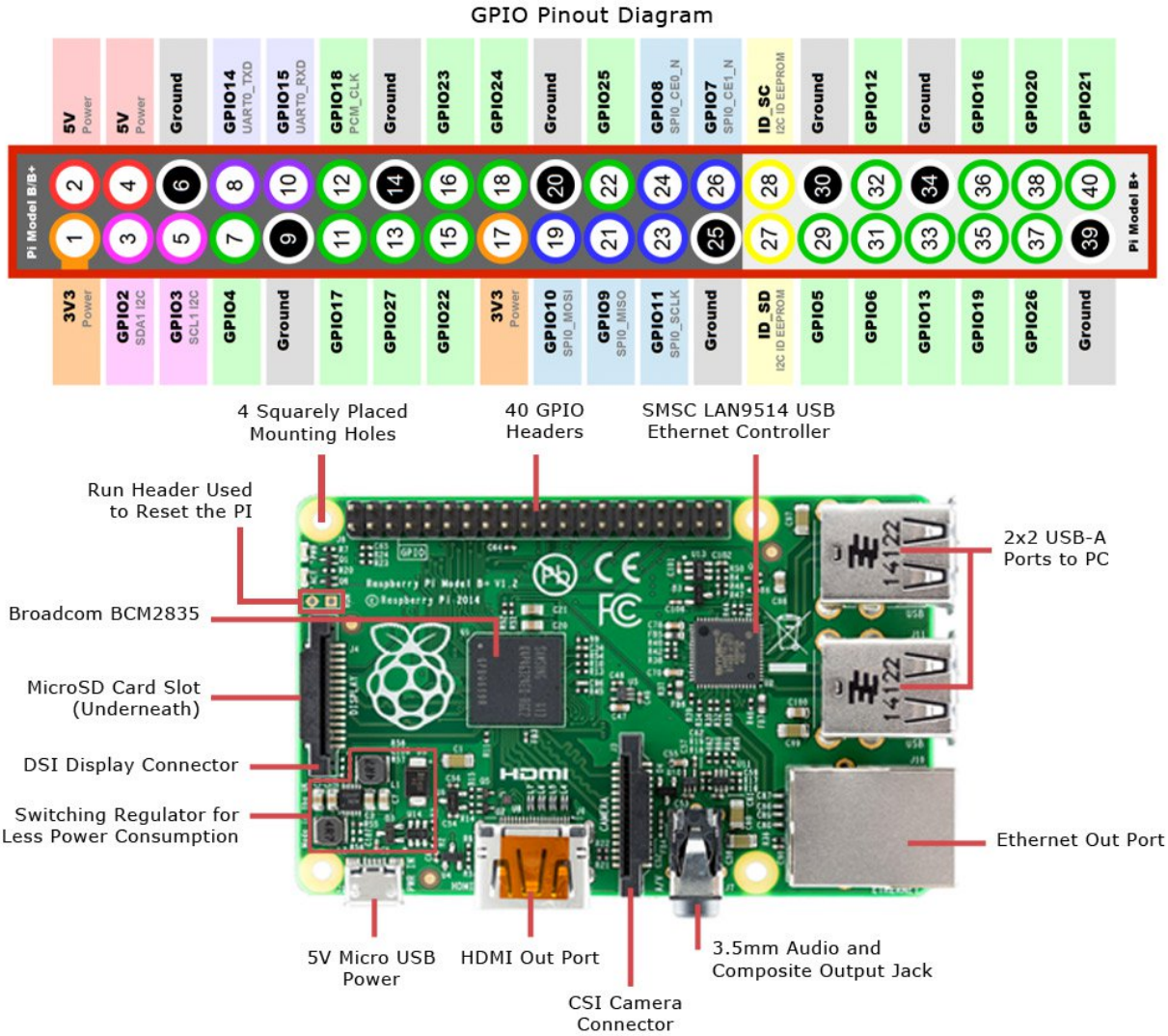
The RASPBERRY Pi 3 is a credit card sized computer that plugs into your TV and a keyboard, its like a little PC which can be used for many of the things that your desktop PC does, like spreadsheets, word processing and games. It also plays high definition video. The design is based around a Broadcom BCM2837 SoC, which includes an ARM1176JZF-S 1.2 Ghz processor, VideoCore IV GPU and 1 GB of RAM. The design does not include a built in hard disk or solid state drive, instead relying on a microSD card for booting and long term storage. This board is intended to run Linux kernel based operating systems.

- Integrated videocore 4 GPU capable of playing full 1080p high definition Blu-Ray quality video
- Free, versatile, and highly developer friendly Debian GNU/Linux operating system
- HDMI video output and RCA video output
- Four USB ports
- 4-pole 3.5mm stereo audio jack with composite video output
- 5V micro USB power input jack
- microSD, MMC, SDIO flash memory card slot
- 40 pin 2.54mm header expansion slot

- Hobby & Hardware Hacking
- Imaging, Video & Vision

- Consumer Electronics
- Communications & Networking

Pin description of Raspberry Pi processor 3



Raspberry Pi processor 3

Alternatives

Order Code	Manufacturer Part No	Manufacturer / Description
2431427	RASPBERRY-MODB+/8GB-USD	RASPBERRY-PI RASPBERRY PI MODEL B+ & 8GB NOOBS SDCARD

	Silicon Manufacturer:Broadcom; Core Architecture:ARM; Core Sub-Architecture:ARM1176JZF-S; Silicon Core Number:BCM2835; Silicon Family Name:BCM2xxx;

Accessories

Order Code	Manufacturer Part No	Manufacturer / Description
2426744		**MULTICOMP** **ENCLOSURE, RASPBERRY PI MODEL B+, CLEAR** Enclosure Type:Raspberry Pi B+ Box; Enclosure Material:Acrylonitrile Butadiene Styrene (ABS); Body Colour:Transparent; External Height - Imperial:1.18"; External Height - Metric:3; More Details
	MC-RP002-CLR	2426745 **MULTICOMP** **ENCLOSURE, RASPBERRY PI MODEL B+, BLACK** Enclosure Type:Raspberry Pi B+ Box; Enclosure Material:Acrylonitrile Butadiene Styrene (ABS); Body Colour:Black; External Height - Imperial:1.18"; External Height - Metric:31mm; More Details
	MC-RP002-BLK	

2427499 T5582DV	**STONTRONICS** **PSU, RASPBERRY PI, 5V, 2A, UK EURO** Power Supply Output Type:Fixed; Input Voltage AC Min:90V; Input Voltage AC Max:264V; Output Voltage Nom.:5V; Output Current Max:2A; More Details

1098522 1658621-9 3D	**TE CONNECTIVITY / AMP** **SOCKET, NOVOLATCH, 40WAY** Product Range:TE - AMP-LATCH Novo Series; Contact Termination:IDC / IDT; Gender:Receptacle; No. of Contacts:40; No. of Rows:2; More Details

1164295 1658622-9 3D	**TE CONNECTIVITY / AMP** **WIRE-BOARD CONNECTOR RECEPTACLE 40POS, 2.54MM** Product Range:TE - AMP-LATCH Novo Series; Contact Termination:IDC / IDT; Gender:Receptacle; No. of Contacts:40; No. of Rows:2; More Details

1494729 PSG90000	**PRO SIGNAL** **LEAD, USB1.1 A PLUG - A PLUG, 1.8M** Connector Type A:USB 1.1 A Plug; Connector Type B:USB 1.1 A Plug; Cable Length - Imperial:5.9ft; Cable Length - Metric:1.8m; Available until stocks are exhausted Alternatives available; More Details
1494729 PSG90000	**PRO SIGNAL** **LEAD, USB1.1 A PLUG - A PLUG, 1.8M** Connector Type A:USB 1.1 A Plug; Connector Type B:USB 1.1 A Plug; Cable Length - Imperial:5.9ft; Cable Length - Metric:1.8m; Available until stocks are exhausted Alternatives available; More Details
1494735 PSG90006	**PRO SIGNAL** **LEAD, USB AM - AF, 1.8M** Connector Type A:USB 1.1 A Socket; Connector Type B:USB 1.1 A Plug; Cable Length - Imperial:5.9ft; Cable Length - Metric:1.8m; Jacket ; More Details

Advantages and disadvantages:

Advantages:

The Raspberry Pi has found its way in to the hobbyist market for computing, but it is also very capable for other business and personal use as well. An extremely low power draw, small form factor, no noise, solid state storage, and other features make it an attractive solution for a small and lightweight server.

Although Raspberry Pi is as small as the size of a credit card, it works as if a normal computer at a relatively low price. it is possible to work as a low-cost server to handle light internal or web traffic. Grouping a set of Raspberry Pi to work as a server is more cost-effective than a normal server. If all light traffic servers are changed into Raspberry Pi, it can certainly minimize an enterprise's budget.
Recently it has been experimenting with Raspberry Pi (revision B) running different GNU/Linux distributions.

1. **Power consumption** - The Pi draws about five to seven watts of electricity. This is about one tenth of what a comparable full-size box can use. Since servers are running constantly night and day, the electrical savings can really add up. I have calculated that the basic Pi kit (Pi board, case, and power supply) will pay for itself with about one year's worth of electricity savings if it's left to run 24x7x365. I ended up with the CanaKit Basic Kit (ASIN # B00DG9D6IK) which is very affordable and good quality.
2. **No moving parts** - The Pi uses an SD card for storage, which is fast and has no moving parts. There are also no fans and other things to worry about. A Class 10 SD card is usually the best performing compared to lower class cards, but this will mainly only affect boot time where there is the most I/O. There is a compatibility chart for SD cards here, results may vary but overall I've had very good luck with Transcend cards which provide a good value.
3. **Small form factor** - The Pi (with a case) can be held in your hand. A comparable full-size box cannot. This means the Pi can be integrated inside of devices, too.
4. **No noise** - The Pi is completely silent.
5. **Status lights** - There are several status lights on the Pi's motherboard. With a clear case you can see NIC activity, disk I/O, power status, etc.
6. **Expansion capabilities** - There are numerous devices available for the Pi, all at very affordable prices. Everything from an I/O board (GPIO) to a camera. The Pi has two USB ports, however by hooking up a powered USB hub, more devices can be added.
7. **Built-in HDMI capable graphics** - The display port on the Pi is HDMI and can handle resolutions up to 1920×1200, which is nice for making the Pi in to a video player box for example. There are some converters that can convert to VGA for backwards compatibility. Alist of HDMI to VGA

converters can be found here. I ended up using the Sanoxy HDMI to VGA cable (ASIN # B0088K7QUQ) which has worked well so far.

8. **Affordable** - compared to other similar alternatives, the Pi (revision B) offers the best specs for the price, at least that I've found. It is one of the few devices in its class that offers 512 MB of RAM. The Pi has come down in price since it first arrived, and is finally affordable as a hobby, business use, or whatever need there is.
9. **Huge community support** - The Pi has phenomenal community support. Support can be obtained quite easily for the hardware and/or GNU/Linux software that runs on the Pi mainly in user forums, depending on the GNU/Linux distribution used. A good list of distributions can be found here.
10. **Over clocking capability** - The Pi can be over clocked if there are performance problems with the application used, but it is at the user's risk to do this.
11. **Multiple uses** - Having the storage on an SD card makes it easy to swap with other SD cards running other GNU/Linux distributions to quickly and easily change the functionality of the Pi. If you want to set up the Pi to run as a server to test it out, then later try something else, just swap the SD card and you're done. Using the "dd" command on a GNU/Linux computer, a backup of the SD card can be created and later restored if needed.

Drawbacks of the Pi:

With all of the positive things about the Pi, there are a couple of items that I feel are very minor drawbacks:

Even though Raspberry Pi can perform different tasks, there are some limitations due to its hardware. Because of its processor, it cannot run X86 operating systems. Some common ones like Windows and Linux distros are not compatible. In addition, some applications which require high demands on CPU processing are off-limits. “Model B took 107 ms to complete one calculation of the purely synthetic prime number test; a mid-range desktop Core 2 Duo E8400 took only 0.85ms.” (Collins, 2012) Users must not use normal computer standards to judge Raspberry Pi. It can work as a personal computer, but cannot replace it.

1. **ARM architecture** - While ARM is a highly efficient and low powered architecture, it is not x86 and therefore any binaries that are compiled to run on x86 cannot run on the Pi. The good news is

that entire GNU/Linux distributions have been compiled for the ARM architecture and new ones are appearing all of the time. There are very few applications that absolutely need x86. The only one that I found so far to be a problem is Wine, which runs Windows programs. Unfortunately, Wine does not work on the Pi.

2. **RAM not upgradable** - The main components of the Pi are soldered to the motherboard, including the RAM which is 512 MB. This is not a problem though as GNU/Linux can easily run on this. I've found the Pi uses about 100 MB of RAM while running as a small server (this is without running X11).

Software for the Pi

The Pi runs GNU/Linux and variants of similar operating systems.

A single or multi-purpose server of these types or others will probably not be an issue with memory, with light use. Memory intensive applications generally become an issue with the X11 desktop and running end user applications.

Today virtualization is very popular so some may say that the cost of spinning up a virtual machine is less than running a Raspberry Pi. But, calculate the power consumption for your hypervisor, and weigh out the differences to see which method in fact costs less overall. Sometimes, a physical box or physical segmentation is needed, or avoiding high costs of running a full hypervisor is a factor, and this is where the Pi can step in.

The Raspberry Pi will run a range of OS Distributions and run a variety of software.

- See Software for an overview and OS Distributions for supported operating system and pre-configured 'images'.
- Officially supported OS distributions include Raspbian, Arch Linux and RISC OS Open.
- Many unofficial distributions are available on the Distributions page.
- Advice is also available if you want to compile a kernel, boot from the network using U-Boot, or test the Pi's performance.
- The Raspberry Pi supports a wide range of programming languages, with many tutorials available.
- Information about installing. A Specific application is available through the link.
- Extensive (boot) configuration info (config.txt) is available here.

- Information about various utilities that can be used with your Raspberry Pi can be found here.

Future Use of Raspberry Pi:

Raspberry Pi was designed by the Raspberry Pi Foundation in UK, initially designed to help kids in the school learn basic computer science knowledge, but the very first idea is that everyone should have a personal computer. To make this idea come true, they make this project start from children at school. The education version of this device quickly showed its pros, as a result of applying this technology in the school, a lot of children develop their knowledge and capability about computer science.

So, in the future they should continue develop in this industry, and create different version for teachers and students, for example, make the outfits seems more cute for children so that it can attract more kids' attention.

The Raspberry Pi is using the Linux as basic programming language, and they attempt to come up with their own language that fits this technology better sometime in the future.

The Raspberry Pi is already pre-order out all, and there is a lot individuals order it because it can be built by the customer's own needs. So there may be a possibility it could be used in some special industry like the spy industry, authorized by the government, it credit card size is a big advantage and it can be transferred to any kind of computer that may just fit the government's needs. But the problem is that this technology is so common that may be traced very easily by the enemy

The current version of Raspberry Pi is only 255MB, in the future, this is a main point to break, however, if the enhanced has a RAM boost, its cost will essentially increase, and the sell price may not be able to cover its cost, but if they increase the sell price they may not get many customer like before. So they may want to keep the cost down, if so, they must cut some budget in other features of model B, such as Bluetooth, inbuilt analogy-to-digital audio converters, the whole technology is will not be very valuable. Since this technology is so good, will it be prevalent**?** The answer is no. As the technology is developing everyday, we got a lot high technology device that function likely as Raspberry Pi, like Smartphone and tablet, the very different between Raspberry Pi and other devices is that you can build your own small personal computer using Raspberry Pi, it may seems attractive for a few people, but others as we know are more lazy and less interested in developing IT knowledge, they like to enjoy the high technology that others created for them. In a word, high technology doesn't always become prevalent.

Applications:

Having recently co-authored a book about building things with the Raspberry Pi (Raspberry Pi Hacks), I've spent a lot of the last couple of years talking about this credit-card-sized Linux computer and seeing fun things people have used it for.

Originally designed for education, the Raspberry Pi was inspired by 1981's BBC Micro, also created for education. But when it was released, what makers saw was a very small computer, perfect for putting in all kinds of projects due to the abilities it has crammed into such a small space. Here are a few of my favorites:

1. Encase it in Lego

One of the first things most people want for their Pis is a case, and there are plenty of them on the market. But Lego bricks give you the opportunity to build anything you want! Should your Pi be baking in a Lego oven? How about a tiny TARDIS? Brian Gillespie built this Raspberry Pi Command Center, which he describes as "fully OSHA, ISO 9001, ASME, IEEE and Sarbanes-Oxley compliant." This adorable home for a Pi named the Pihaus.

2. Play video games

There are a lot of options here. You could simply install some Linux-friendly games or emulators. You could use it as your own Steam machine. There's this teeny tiny arcade machine or a cocktail cabinet-style video game machine.

3. Automate your house

This is the promise of *The Jetsons*, right? Automate your pet feeder. Keep the yard a little greener by automating the sprinklers. Turn off the lights without getting up. But, my favorite overlays the *Star Trek* LCARS interface into a multi-functional home automation system.

4. Win your geek girl's love forever

This may be my absolute favorite of all. Lingxiang Xiang built this Pi-powered R2-D2 who is bilingual with facial recognition abilities in addition to motion and distance detection, audio record, and playback.

5. Take photos from the sky

Many groups, including NC Nearspace, have found that the Pi is incredibly well-suited to near-space photography payloads. It has SD storage for all the photos and video, GPIO for the accelerometer, gyroscope and various sensors, the ability to send live video out, and an accessory camera made for the device.

6. Turn the Pi into a radio

This is a great project to introduce kids to both how FM radio works ("well, kids, back in our day, there was no Sirius or Spotify or playlists on your phone...") as well as some introductory electronics and programming. The original instructions for PiFM are pretty simple, but you may want to grab this rewrite, which uses less CPU.

7. Monitor your homebrew

Gotta keep an eye on your fermentation! BrewPi takes care of it for you, logging and controlling temperature and giving you a web interface to keep an eye on it all.

8. Media streamer

Setting up a media centre using the Raspberry Pi is a popular choice, perhaps unsurprisingly given how easy it is to connect the board to a TV via composite or HDMI.

Setting up a Pi-based streamer will be relatively straightforward using the free open source media centre software XBMC. The software can stream music and video stored online or locally, and can be configured to work with TV catch-up services like BBC iPlayer, as well as being completely controllable using a remote control.

The team behind XBMC has a video of it playing 1080p video without any difficulty.

However there is one rather large drawback, the on-demand film streaming services offered by Lovefilm and Netflix are both currently incompatible with XBMC running on the Pi. Anyone wanting to stream content wirelessly will also have to add a USB Wi-Fi adapter.

9. Arcade machine

The Raspberry Pi will help greying gamers relive their misspent youth thanks to a number of projects to bring classic arcade game and 1980s computer emulators to the system.

Projects range from recreation of full arcade cabinets, using a Raspberry Pi running MameWAH emulator on a Linux OS under the hood, to running the Pi as the classic Commodore 64 or BBC Micro computer.

If handheld games are more your thing then you could always follow in the footsteps of this more ambitious project, which involves connecting up the Pi to a 3.5-inch TFT screen to create a portable gaming system.

10. Tablet computer

The end result might not give the iPad a run for its money but a number of modders are already working out how to pair the Pi with a touch screen.

While not a straightforward task, modders have already identified LCD touch screens and batteries that could be used. The main challenge seems to be keeping costs down and finding a touchscreen-enabled distro of Linux that works with the Pi's ARM 11 processor.

And while it might lack the sleek lines of a MacBook Air, this idea for a Raspberry Pi laptop, or more accurately a Raspberry Pi computer crossed with a suitcase, is starting to take shape on the Raspberry Pi.org forums.

11. Home automation

Doing the rounds are various Pi-based home automation projects, which hope to use the board as a ZigBee home automation server.

ZigBee systems can be set up to support a range of tasks, including remote controlled air conditioning and lighting, and checking whether doors are open or closed.

12. Carputer

Creating a touchscreen, in-car tablet computer based on the Pi is another popular goal among modders.

Tech hobbyists want a Pi carputer to be a media player and GPS that can run off the cigarette lighter, and have pulled together a list of specs for the device.

There is even talk of further adding a web cam to the device to create a black box that would record the car's journeys or, even more ambitiously, hooking it up to the car's diagnostic system to provide a real time display of the engine's state.

13. Internet radio

Internet radios still cost a pretty penny, so why not pair up the Pi with a low-cost LCD screen, some speakers and create your own.

Various Pi-based internet radio projects already exist and are piecing together the components and code needed to create a Pi-based internet radio, and it seems only a matter of time before this becomes a reality.

14. Controlling robots

Robots and Raspberry Pi seem to be a match made in heaven, if hobbyist projects are anything to go by.

Projects include controlling quadroceptor drones (the jury's still out on this one), an autonomous plane, a voice controlled Star Wars R5-D4 droid and a robot boat.

Messing around with robots and the Raspberry Pi will be helped by expansion boards for the Pi like the forthcoming Gertboard, which will make it easier to hook the computer to a range of motors and sensors.

15. Cosmic computer

As far fetched as it sounds, the Raspberry Pi is even being considered for a trip into space.

Universities are examining whether an array of Pi boards could serve as an onboard computer system for mini satellites, Eben Upton, director of the Raspberry Pi Foundation, told TechRepublic earlier in the year.

The idea is that Raspberry Pi computers could provide a low-cost, off the shelf alternative to the bespoke hardware used normally in spacecraft and satellite systems. Using scores of Raspberry Pi boards would build redundancy into the platform, allowing one board to take over if another fails.

The University of Surrey is already trialing the use of off the shelf computing in the place of bespoke hardware, by sending an Android Smartphone into orbit on board a 4kg nano-satellite.

Another space-bound Pi project in the works is sending a board to the edge of space by weather balloon to use it as an eye in the sky.

16. Hunting for meteorites

Even if the Pi doesn't make it in to space it seems likely that it will be watching the cosmos.

A high school project in Australia is looking to use the Pi to spot meteorites blazing across the Antipodean skies.

Pi boards would analyze feeds from low-cost web cams and grab timestamped images of any suspected meteorite trails.

17. Coffee and Pi

Slightly less ambitious than some of the projects in the pipeline, but no less deserving of a mention is the MoccaPi.

The MoccaPi is a Raspberry Pi controlled coffee machine designed to conjure up the perfect brew via a few typed commands.

All you need is a Python controlled microcontroller, a coffee machine and some relays cables and an SDcard, which all told shouldn't cost much more than $100.

Tell TechRepublic what you plan to do with your Raspberry Pi by emailing nick dot heath at techrepublic.com.

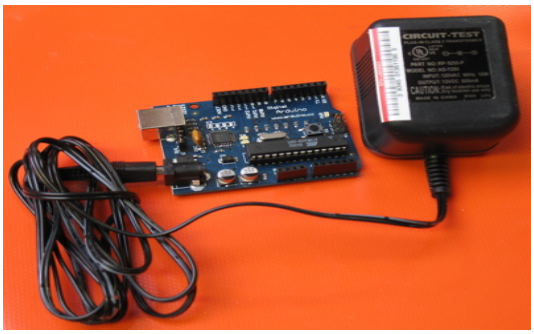
3.3 Adapter:

Adapter

The AC adapter, AC/DC adapter or AC/DC converter is a type of external power supply, often enclosed in a case similar to an AC plug. Other names include plug pack, plug-in adapter, adapter block, domestic mains adapter, line power adapter, wall wart, or power adapter. AC adapters are used with electrical devices that require power but do not contain internal components to derive the required voltage and power from mains power. The internal circuitry of an external power supply is very similar to the design that would be used for a built-in or internal supply.

External power supplies are used both with equipment with no other source of power and with battery-powered equipment, where the supply, when plugged in, can sometimes charge the battery in addition to powering the equipment.

Use of an external power supply allows portability of battery-powered equipment without the added bulk of internal power components and makes it unnecessary to produce equipment for use only with a specified power source



- must be a DC adapter (i.e. it has to put out DC, not AC);
- should be between 9V and 12V DC (see note below);
- must be rated for a minimum of 250mA current output, although you will likely want something more like 500mA or 1A output, as it gives you the current necessary to power a servo or twenty LEDs if you want to.
- must have a 2.1mm power plug on the Arduino end, and

- The plug must be "centre positive", that is, the middle pin of the plug has to be the + connection.

Operation:

Originally, most AC/DC adapters were linear power supplies, containing a transformer to convert the mains electricity voltage to a lower voltage, a rectifier to convert it to pulsating DC, and a filter to smooth the pulsating waveform to DC, with residual ripple variations small enough to leave the powered device unaffected. Size and weight of the device was largely determined by the transformer, which in turn was determined by the power output and mains frequency. Ratings over a few watts made the devices too large and heavy to be physically supported by a wall outlet. The output voltage of these adapters varied with load; for equipment requiring a more stable voltage, linearvoltage regulator circuitry was added. Losses in the transformer and the linear regulator were considerable; efficiency was relatively low, and significant power dissipated as heat even when not driving a load.

In the early twenty-first century, switched-mode power supplies (SMPSs) became almost ubiquitous for this purpose. Mains voltage is rectified to a high direct voltage driving a switching circuit, which contains a transformer operating at a high frequency and outputs direct current at the desired voltage. The high-frequency ripple is more easily filtered out than mains-frequency. The high frequency allows the transformer to be small, which reduces its losses; and the switching regulator can be much more efficient than a linear regulator. The result is a much more efficient, smaller, and lighter device. Safety is ensured, as in the older linear circuit, because there is still a transformer which electrically isolates the output from the mains.

A linear circuit must be designed for a specific, narrow range of input voltages (e.g., 220–240VAC) and must use a transformer appropriate for the frequency (usually 50 or 60 Hz), but a

switched-mode supply can work efficiently over a very wide range of voltages and frequencies; a single 100–240VAC unit will handle almost any mains supply in the world.

However, unless very carefully designed and using suitable components, switching adapters are more likely to fail than the older type, due in part to complex circuitry and the use of semiconductors. Unless designed well, these adapters may be easily damaged by overloads, even transient ones, which can come from lightning, brief mains over voltage (sometimes caused by an incandescent light on the same power circuit failing), component degradation, etc. A very common mode of failure is due to the use of electrolytic capacitors whose equivalent series resistance (ESR) increases with age; switching regulators are very sensitive to high ESR (the older linear circuit also used electrolytic capacitors, but the effect of degradation is much less dramatic). Well-designed circuits pay attention to the ESR, ripple current rating, pulse operation, and temperature rating of capacitors.

Description:

Efficiency:

The issue of inefficiency of some power supplies has become well known, with US President George W. Bush referring in 2001 to such devices as "Energy Vampires". Legislation is being enacted in the EU and a number of U.S. states, to reduce the level of energy wasted by some of these devices. Such initiatives include standby power and the One Watt Initiative.

But others have argued that these inefficient devices are low powered, e.g., devices that are used for small battery chargers, so even if they have a low efficiency, the amount of energy they waste is less than 1% of household consumption of electric energy.

Considering the total efficiency of power supplies for small electronic equipment, the older mains-frequency linear transformer-based power supply was found in a 2002 report to have efficiencies from 20–75%, and have considerable energy loss even when powered up but not supplying power. Switched-mode power supplies (SMPSs) are much more efficient; a good design can be 80–90% efficient, and is also much smaller and lighter. In 2002 most external plug-in "wall wart" power adapters commonly used for low-power consumer electronics devices were of linear design, as well as supplies built into some equipment.

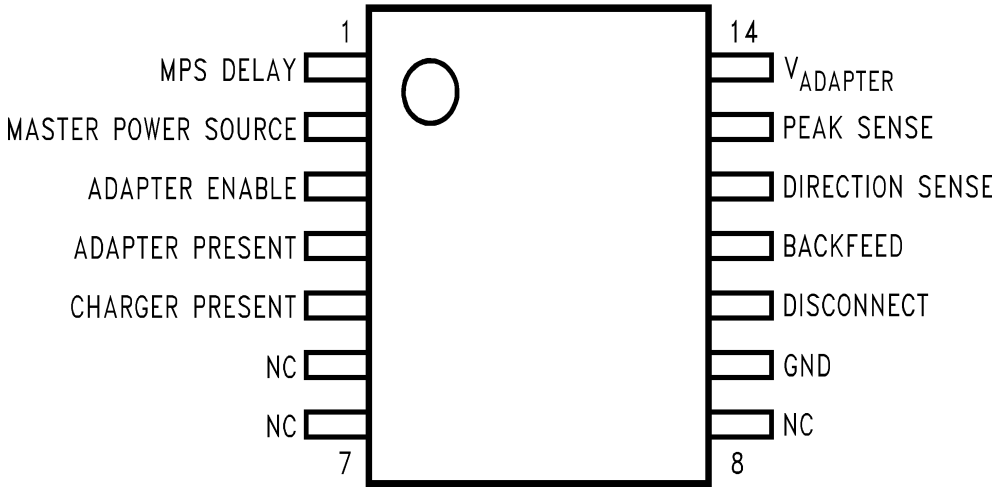
External supplies are usually left plugged in even when not in use, and consume from a few watts to 35 watts of power in that state. The report concluded that about 32 billion kilowatt-hours (kWh) per year, about 1% of total electrical energy consumption, could be saved in the United States by replacing all linear power supplies (average efficiency 40–50%) with advanced switching designs (efficiency 80–90%), by replacing older switching supplies (efficiencies of less than 70%) with advanced designs (efficiency of at least 80%), and by reducing standby consumption of supplies to not more than 1 watt.

Since the report was published, SMPSs have indeed replaced linear supplies to a great extent, even in wall warts. The 2002 report estimated that 6% of electrical energy used in the US "flows through" power supplies (not counting only the wall warts). The website where the report was published said in 2010 that despite the spread of SMPSs, "today's power supplies consume at least 2% of all U.S. electricity production. More efficient power supply designs could cut that usage in half".

Since wasted electrical energy is released as heat, an inefficient power supply is hot to the touch, as is one that wastes power without an electrical load. This waste heat is itself a problem in warm weather, since it may require additional air conditioning to prevent overheating, and even to remove the unwanted heat from large supplies.

Block Diagram:

Pin Configuration**:**



PIN DESCRIPTIONS

Pin No.	Name	Function
1	MPS DELAY	A capacitor between this pin and ground sets the delay of the MPS risetime. See MPS DELAY description in Typical Application section.
2	MASTER POWER SOURCE	Bi-directional logic pin. If driven high by an external source, indicates that a battery is powering the power bus. If driven high by the AIC, indicates the adapter is powering the bus. AIC cannot drive MPS low. If there is no valid adapter voltage present, the pin is not an output but a high impedance logic input. The input is pulled-down via an internal 40k resistor.
3	ADAPTER ENABLE	Logic input pin. Active high. Grants permission to the adapter to drive both the power bus and the MPS signal.
4	ADAPTER PRESENT	Logic output pin. High when 12 volts < $V_{ADAPTER}$ < 17 volts. The output typically has 40k pull-down resistor. The source current is not internally limited and the part can be damaged if the output is shorted to ground when driven HIGH.
5	CHARGER PRESENT	Logic output pin. High when $V_{ADAPTER}$ > 17 volts. The output typically has 40k pull-down resistor. The source current is not internally limited and the part can be damaged if the output is shorted to ground when driven HIGH.
9	GND	IC ground pin.
10	DISCONNECT	Drives the gate of the disconnect P-ch FET.
11	BACKFEED	Drives the gate of the backfeed P-ch FET.
12	DIRECTION SENSE	Connection for current sense resistor to control BACKFEED.
13	PEAK SENSE	Connection for current sense resistor to control DISCONNECT.
14	$V_{ADAPTER}$	Power input pin. Output of AC adapter, auto adapter or airline adapter.

Absolute Maximum Ratings [1][2]

$V_{ADAPTER}$		−0.3V to 28V
$V_{PEAK\ SENSE}$, $V_{DIRECTION\ SENSE}$, $V_{BACKFEED}$, $V_{DISCONNECT}$		−0.3V to $V_{ADAPTER}$
$V_{ADAPTER\ ENABLE}$, $V_{MASTER\ POWER\ SOURCE}$		−0.3V to 5.5V
$V_{MPS\ DELAY}$		−0.3V to 2V
Storage Temperature		−40°C to 125°C
ESD Ratings	Human Body Model[3]	2kV
	Machine Model[4]	200V

Operating Ratings [1]

Supply Voltage ($V_{ADAPTER}$)	5V to 24V
Ambient Temp. Range	−20°C to 70°C
Junction Temp. Range	−20°C to 85°C

Universal power adapters:

One inherent disadvantage of *external* power adapters is that they can get separated from the product they are intended to power. Consequently, there is a market for replacement adapters. In addition, failed power supplies must be replaced. Not only must the replacement match voltage, current, and polarity requirements, but it must also match the connector. Many electrical products are poorly labeled with information concerning the power supply they require, so it is prudent to record the specifications of the original power supply in advance, to ease replacement if the original is later lost. Careful labeling of power adapters can also reduce the likelihood of a disastrous mix-up which could cause equipment damage.

Some "universal" replacement power supplies allow the voltage and polarity to be switched, which can ease the matching problem. In addition, the power connector must be matched.

Four-way X connectors or six-way star connectors, also known as spider connectors, with multiple plug sizes and types are common on generic power supplies. Other replacement power supplies have arrangements for changing the power connector, with four to nine different alternatives available when purchased in a set. RadioShack sells universal AC adapters of various capacities, branded as "Enercell Adaptaplug", and fitted with 2-pin female sockets compatible with their Adaptaplug connector lineup. This allows many different configurations of AC adapters to be put together, without requiring soldering. Phil more and other competing brands offer similar AC adapters with interchangeable connectors.

A suitable power supply for a particular use must have the matching plug dimensions, the matching DC (or AC) voltage and polarity, and the ability to supply the minimum required current. The input voltage must match the wall socket (120/240VAC at 60/50 Hz) or other power source, such as 12VDC automotive battery power.

However, the label on a power supply may not be a reliable guide to the actual voltage it supplies under varying conditions. Most low-cost power supplies are "unregulated", in that their voltage can change considerably with load. If they are lightly loaded, they may put out much more than the nominal "name plate" voltage, which could damage the load. If they are heavily loaded, the output voltage may droop appreciably, in some cases well below the nominal label voltage even within the nominal rated current, causing the equipment being supplied to malfunction or be damaged. Cheap external power supplies of traditional linear design with undersized transformers tend to have poor regulation, whether originally supplied or replacement units.

In general, more modern high-quality switched-mode power supplies (SMPSs) are smaller, more efficient, and put out a much more constant voltage even as the input voltage and the load current may vary. Configurable switched-mode power supplies have come down considerably in price, and they are especially convenient for use when traveling because of their decreased weight and size.

Auto-sensing adapters:

Some universal adapters, for example those by iGo, automatically set their output voltage and maximum current according to which of a range of interchangeable tips is fitted; tips are available to fit and supply appropriate power to many notebook computers and mobile devices. Different tips may use the same connector, but automatically supply different power; it is essential to use the right tip for the apparatus being powered, but no switch needs to be set correctly by the user. The advent of switch-mode power supplies has allowed adapters to work from any AC mains supply from 100 to 240V with an appropriate plug; operation from standard 12V DC vehicle and aircraft supplies can also be supported. With the appropriate adapter, accessories, and tips, a variety of equipment can be powered from almost any source of power.

A "Green Plug" system has been proposed, based on USB technology, by which the consuming device would tell the external power supply what kind of power is needed.

Features:

- 100% Brand New
- Excellent Quality
- Short Circuit, Over Voltage & Over Current Protection.
- Meet CEC Energy Efficiency Level IV.
- Incredibly Low Fault Rates
- No Minimum Load.
- This power supply is a regulated Center Positive power supply and has a 2.1mm x 5.5mm Jack
- Its plug design is for Indian power socket. So, no plug converter is required.
- Compact size & light weight.
- High Reliability.
- Regulated Stable Voltage.
- Good quality SMPS Based Adapter
- Power LED Monitor (LED Glow when in Use)
- Stabilized Output, low ripple & low interference
- Single Output Voltage
- High Efficiency & low energy consumption
- Input - 100-240 VAC 50/60hZ
- Category - Switch Mode Power Adaptor (SMPS)
- Output Type - DC
- Output - 12Volts 1Amp

Applications:

- Powerful 12v 12w 1A max Current Draw. Replaces lower amped adapters 12v 0.5A 1A. 1.5A etc
- Smart Replacement Gadget
- Power Supply for LED, SMD, LED Strip, RGB LED Strip
- Ideal for Routers / Modems / Mobile Phones / Mp3 players / POS Machines etc.
- Best for Routers , Wifi Routers security/spy camera receiver and some advanced cameras CCTV , Gadgets , Portable Players , Set Top Boxes, best for Toys etc., Charging or any gadgets as per the rating of the device , please study and then buy as this a very technical item only works as per its precise current outputs

- This power supply is an ideal replacement for a wireless network router such as the Netgear DG834, DG834GT, DG934 etc.. Plus a range of many other wireless routers. You will need to check the DC socket size & power rating with the supplier of the router you are using as we can not confirm it will work. Please do not ask us as we are unable to verify this kind of information due to the large range of routers currently available on the market. This power unit s also suitable for any other equipment requiring a 12 volt DC supply up to a maximum of 1 Amp.

Disadvantages:

"Power brick" in-line configuration, with detachable AC cord

While useful for many purposes, some external AC adapters have attracted criticism. Problems with this type of power supply include, but are not limited to:

- Size — Power supplies which plug into the mains directly without using a plug on a cable (true wall warts) are bulkier than bare plugs; sometimes they are too large to plug into power sockets with restricted space, or into adjacent sockets on power strips.
- Weight — Some AC adapters can be heavy, exerting excess weight on the power socket (this depends on the socket design of the country in question). Some external power supplies are "power bricks" (also known as "line lumps") having a short AC cord so they can lie on the floor, thus relieving strain, at the expense of clutter. Other wall-hanging types are made long and thin, minimizing the leverage of their weight vector that pulls the plug out, at the expense of exacerbating the size problem. The weight for equipment that must be carried (e.g., for traveling) is not a disadvantage of external supplies, as the alternative is an equally heavy internal supply; in many cases a single universal supply can replace several proprietary ones.
- Inefficiency — some idling power is wasted as the power supply is left running when the equipment power switch is off or the equipment is disconnected from the power supply. In recent years it has become common for equipment with internal supplies to share this problem due to the use of a "soft" power switch.

- Confusion — External power supplies are often generic and not clearly marked to identify the equipment they are designed to power. It is very easy to separate power supply and equipment, and difficult to re-match the many power supplies and devices that users possess.
- Compatibility problems — there is no standardization of connectors; the same connector is often used for different voltages, and for both DC supplies and AC-to-AC transformers. Incompatible voltage or polarity may be present on physically interchangeable connectors. This easily leads to using the wrong power supply, which can destroy equipment.

A survey of consumers showed widespread dissatisfaction with the cost, inconvenience, and wastefulness of the profusion of power adapters used by electronic devices. The science fiction author and satirist Douglas once wrote an essay bemoaning the profusion and confusion of power adapters, and calling for more standardization.

3.4 Wi-Fi

Introduction:

Wi-Fi or WLAN as it is commonly known is fast becoming the preferred mode of connecting to the internet. Many people are not aware of the **descriptions and explanations** related to it. **Wi-Fi** gets its name from a certification called **Wireless Fidelity** given to **networks** operating under 802.11 standards. **Wi-Fi** allows computers, PDAs and other devices to connect to a broadband connection in a **wireless mode**. The 802.11 standard defines the **wireless communication** operating via electromagnetic waves. While reading the **descriptions and explanations** related to Wi-Fi, one should remember there are different **modes for wirelessnetworks** like Infrastructure mode and Ad-Hoc mode that can be used for different criteria.

Wi-Fi allows us to create wireless local area networks at high speed. In practice, the Wi-Fi can connect laptops, desktops, PDAs or other devices (printers, game consoles) to a broadband connection (300 Mbps) over a radius of several meters indoors (usually between 20 and 50 meters). In an open environment, the range can reach over several hundred of meters in optimal conditions.

ISPs are starting to equip areas with high concentrations of internet users (stations, airports, hotels, trains, etc.) with wireless internet access. These access areas are called "hot spots".

Wi-Fi is a mechanism for wirelessly connecting electronic devices. A device enabled with Wi-Fi, such as a personal computer, video game console, smart phone, or digital audio player, can connect to the Internet via a wireless network access point. An access point (or hotspot) has a range of about 20 meters (65 ft) indoors and a greater range outdoors. Multiple overlapping access points can cover large areas.

A Wi-Fi enabled device such as a PC, video game console, mobile phone, MP3 player or PDA can connect to the Internet when within range of a wireless network connected to the Internet. The coverage of one or more interconnected access points — called a hotspot — can comprise an area as small as a single room with wireless-opaque walls or as large as many square miles covered by overlapping access points. "Wi-Fi" is a trademark of the Wi-Fi Alliance and the brand name for products using the IEEE 802.11 family of standards. Wi-Fi is used by over 700 million people. There are over four million hotspots (places with Wi-Fi Internet connectivity) around the world, and about 800 million new Wi-Fi devices are sold every year. Wi-Fi products that complete Wi-Fi Alliance interoperability certification testing successfully may use the "Wi-Fi CERTIFIED" designation and trademark.

3.5.1 History:

The name of a popular wireless networking technology that uses radio waves to provide wireless high-speed Internet and network connections. The Wi-Fi Alliance, the organization that owns the Wi-Fi (registered trademark) term specifically defines Wi-Fi as any "wireless local area network (WLAN) products that are based on the Institute of Electrical and Electronics Engineers' (IEEE) 802.11 standards."

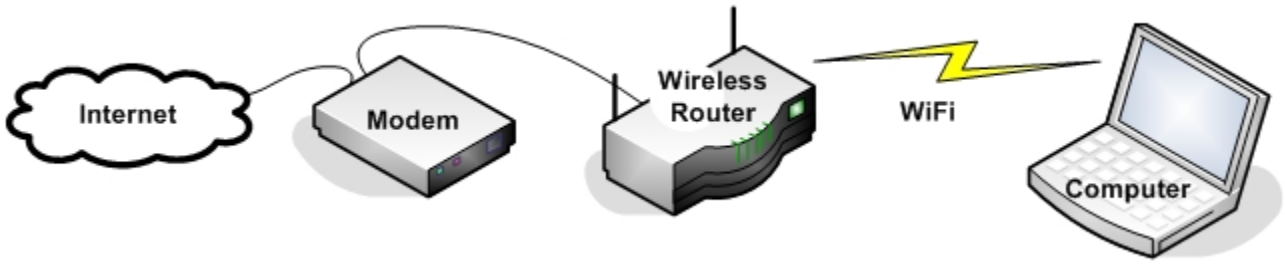
Wi-Fi is a popular term that is used referring to wireless communications between computers and other computer related devices. Regular radio waves are use to broadcast and receive just like a pair of walkie talkies, but on a much higher frequency.
These days you will find many areas around town that offer free Wi-Fi internet access, but the most common use for Wi-Fi is in the home. If you have high speed Internet (DSL, Cable, Satellite, etc) and have a piece of equipment that is called a wireless router, then you have Wi-Fi.

Initially, Wi-Fi was used in place of only the 2.4GHz 802.11b standard, however the Wi-Fi Alliance has expanded the generic use of the Wi-Fi term to include any type of network or WLAN

product based on any of the 802.11 standards, including 802.11b, 802.11a, dual-band, and so on, in an attempt to stop confusion about wireless LAN interoperability.

Wi-Fi works with no physical wired connection between sender and receiver by using radio frequency (RF) technology, a frequency within the electromagnetic spectrum associated with radio wave propagation. When an RF current is supplied to an antenna, an electromagnetic field is created that then is able to propagate through space. The cornerstone of any wireless network is an access point (AP). The primary job of an access point is to broadcast a wireless signal that computers can detect and "tune" into. In order to connect to an access point and join a wireless network, computers and devices must be equipped with wireless network adapters.

Wi-Fi is supported by many applications and devices including video game consoles, home networks, PDAs, mobile phones, major operating systems, and other types of consumer electronics. Any products that are tested and approved as "Wi-Fi Certified" (a registered trademark) by the Wi-Fi Alliance are certified as interoperable with each other, even if they are from different manufacturers. For example, a user with a Wi-Fi Certified product can use any brand of access point with any other brand of client hardware that also is also "Wi-Fi Certified". Products that pass this certification are required to carry an identifying seal on their packaging that states "Wi-Fi Certified" and indicates the radio frequency band used (2.5GHz for 802.11b, 802.11g, or 802.11n, and 5GHz for 802.11a).



In the illustration above you will see what equipment is used in a typical home Wi-Fi network. In some cases an Internet service provider will supply a wireless router/modem combination when you subscribe to their service.

A common misconception is that the term Wi-Fi is short for ***"wireless fidelity,"*** however this is not the case. Wi-Fi is simply a trademarked term meaning IEEE 802.11x.

No matter where you are, you can access the world of web through your handsets and your laptops and your iPads. You might not have noticed what it is but the technology that enables you to plug in internet without any wires whether you are in a cafe, a library, a shopping mall or an airport is Wi-Fi – the wireless network also known as 802.11. The circumference where **wireless technology** is present and available to the users is known as Hotspot. The inexpensive, user-friendly **Wi-Fi networks** are also obtrusive; if you do not need one you would not know there exists any. Wi-Fi could be also installed in home or offices in order to transmit information over the air without the aid of wires. In near future you would find wireless networking available in every nook and corner.

Wi-Fi is derived from the decades old term Hi-Fi that stands for the output's type produced by quality music hardware.Wi-Fi Technology is WIRELESS FIDELITY and stands for all those technologies that fall under the specifications of IEEE 802.11 including 802.11a, 802.11b and 802.11g. The association of the term Wi-Fi with various technologies is merely because of the promotions made by the Wi-Fi Alliance.

For those whose laptops and cell phones do not have a built-in wireless transmitter then you could purchase a wireless adaptor and inject it into USB port. A Wi-Fi hotspot is automatically discovered and connected by the transmitters. The presence of Wi-Fi in public places makes it convenient to stay connected to your official tasks or to the social networking.

3.5.2 What Is Wi-Fi?

A wireless network uses radio waves, just like cell phones, televisions and radios do. In fact, communication across a wireless network is a lot like two-way radio communication. Here's what happens:

1. A computer's wireless adapter translates data into a radio signal and transmits it using an antenna.
2. A wireless router receives the signal and decodes it. The router sends the information to the Internet using a physical, wired Ethernet connection.

The process also works in reverse, with the router receiving information from the Internet, translating it into a radio signal and sending it to the computer's wireless adapter.

The radios used for Wi-Fi communication are very similar to the radios used for walkie-talkies, cell phones and other devices. They can transmit and receive radio waves, and they can convert 1s and

0s into radio waves and convert the radio waves back into 1s and 0s. But Wi-Fi radios have a few notable differences from other radios:

- They transmit at frequencies of 2.4 GHz or 5 GHz. This frequency is considerably higher than the frequencies used for cell phones, walkie-talkies and televisions. The higher frequency allows the signal to carry more data.

They use 802.11 networking standards, which come in several flavors: **802.11a** transmits at 5 GHz and can move up to 54 megabits of data per second. It also uses **orthogonal frequency-division multiplexing** (OFDM), a more efficient coding technique that splits that radio signal into several sub-signals before they reach a receiver. This greatly reduces interference. **802.11b** is the slowest and least expensive standard. For a while, its cost made it popular, but now it's becoming less common as faster standards become less expensive. 802.11b transmits in the 2.4 GHz frequency band of the radio spectrum. It can handle up to 11 megabits of data per second, and it uses **complementary code keying** (CCK) modulation to improve speeds. **802.11g** transmits at 2.4 GHz like 802.11b, but it's a lot faster -- it can handle up to 54 megabits of data per second. 802.11g is faster because it uses the same OFDM coding as 802.11a. **802.11n** is the newest standard that is widely available. This standard significantly improves speed and range. For instance, although 802.11g theoretically moves 54 megabits of data per second, it only achieves real-world speeds of about 24 megabits of data per second because of network congestion. 802.11n, however, reportedly can achieve speeds as high as 140 megabits per second. The standard is currently in draft form -- the **Institute of Electrical and Electronics Engineers (IEEE)** plans to formally ratify 802.11n by the end of 2009.

- Other 802.11 standards focus on specific applications of wireless networks, like wide area networks (WANs) inside vehicles or technology that lets you move from one wireless network to another seamlessly.
- Wi-Fi radios can transmit on any of three frequency bands. Or, they can "frequency hop" rapidly between the different bands. Frequency hopping helps reduce interference and lets multiple devices use the same wireless connection simultaneously.

As long as they all have wireless adapters, several devices can use one router to connect to the Internet. This connection is convenient, virtually invisible and fairly reliable; however, if the router fails or if too many people try to use high-bandwidth applications at the same time, users can experience interference or lose their connections.

3.5.3 Wi-Fi Hotspots

If you want to take advantage of public Wi-Fi hotspots or start a wireless network in your home, the first thing you'll need to do is make sure your computer has the right gear. Most new laptops and many new desktop computers come with built-in wireless transmitters. If your laptop doesn't, you can buy a **wireless adapter** that plugs into the PC card slot or USB port. Desktop computers can use USB adapters, or you can buy an adapter that plugs into the PCI slot inside the computer's case. Many of these adapters can use more than one 802.11 standard.Once you've installed your wireless adapter and the drivers that allow it to operate, your computer should be able to automatically discover existing networks. This means that when you turn your computer on in a Wi-Fi hotspot, the computer will inform you that the network exists and ask whether you want to connect to it. If you have an older computer, you may need to use a software program to detect and connect to a wireless network.Being able to connect to the Internet in public hotspots is extremely convenient. Wireless home networks are convenient as well. They allow you to easily connect multiple computers and to move them from place to place without disconnecting and reconnecting wires. In the next section, we'll look at how to create a wireless network in your home.

3.5.4 Building a Wireless Network

If you already have several computers networked in your home, you can create a wireless network with a **wireless access point**. If you have several computers that are not networked, or if you want to replace your Ethernet network, you'll need a wireless router. This is a single unit that contains:

1. A port to connect to your cable or DSL modem
2. A router
3. An Ethernet hub
4. A firewall
5. A wireless access point

A wireless router allows you to use wireless signals or Ethernet cables to connect your computers to one another, to a printer and to the Internet. Most routers provide coverage for about 100 feet (30.5 meters) in all directions, although walls and doors can block the signal. If your home is very large, you can buy inexpensive range extenders or repeaters to increase your router's range.

As with wireless adapters, many routers can use more than one 802.11 standard. 802.11b routers are slightly less expensive, but because the standard is older, they're slower than 802.11a, 802.11g and 802.11n routers. Most people select the 802.11g option for its speed and reliability.

Once you plug in your router, it should start working at its default settings. Most routers let you use a Web interface to change your settings. You can select:

- **The name of the network, known as its service set identifier (SSID)** -- The default setting is usually the manufacturer's name.
- **The channel that the router uses** -- Most routers use channel 6 by default. If you live in an apartment and your neighbors are also using channel 6, you may experience interference. Switching to a different channel should eliminate the problem.
- **Your router's security options** -- Many routers use a standard, publicly available sign-on, so it's a good idea to set your own username and password.

Security is an important part of a home wireless network, as well as public Wi-Fi hotspots. If you set your router to create an open hotspot, anyone who has a wireless card will be able to use your signal. Most people would rather keep strangers out of their network, though. Doing so requires you to take a few security precautions.

It's also important to make sure your security precautions are current. The Wired Equivalency Privacy (WEP) security measure was once the standard for WAN security. The idea behind WEP was to create a wireless security platform that would make any wireless network as secure as a traditional wired network. But hackers discovered vulnerabilities in the WEP approach, and today it's easy to find applications and programs that can compromise a WAN running WEP security.

To keep your network private, you can use one of the following methods:

- **Wi-Fi Protected Access** (WPA) is a step up from WEP and is now part of the 802.11i wireless network security protocol. It uses temporal key integrity protocol (TKIP) encryption. As with WEP, WPA security involves signing on with a password. Most public hotspots are either open or use WPA or 128-bit WEP technology, though some still use the vulnerable WEP approach.
- **Media Access Control** (MAC) address filtering is a little different from WEP or WPA. It doesn't use a password to authenticate users -- it uses a computer's physical hardware. Each computer has its own unique MAC address. MAC address filtering allows only machines with specific MAC addresses to access the network. You must specify which addresses are allowed when you set up your router. This method is very secure, but if you buy a new computer or if visitors to your home want to use your network, you'll need to add the new machines' MAC addresses to the list of approved addresses. The

system isn't foolproof. A clever hacker can **spoof** a MAC address -- that is, copy a known MAC address to fool the network that the computer he or she is using belongs on the network.

Wireless networks are easy and inexpensive to set up, and most routers' Web interfaces are virtually self-explanatory.

3.5.5 Infrastructure mode:

Infrastructure mode is one of the two methods for connecting to wireless networks with Wi-Fi enabled devices such as laptops, Pda's I-phone etc. These devices are connected to wireless network with the help of Access point (AP). Wireless Access Points are usually routers or switches which are connected to internet by Ethernet port.

Wireless Access points are always required for infrastructure mode of wireless networking. It is necessary to use SSID while configuring AP, this SSID should be known to clients for their computers to connect WLAN. SSID is basically security key which help prevent UN authorized access to WLAN. The Access point is then connected to wire network (Internet) to provide wireless internet connectivity to clients. Multiple access points can be added in the WLAN, this increases the reach of infrastructure for supporting many number of wireless clients.

The most common modes for Wi-Fi connectivity are as follow:

- Infrastructure mode
- Ad hoc Mode

Infrastructure mode:

Infrastructure mode is one of the two methods for connecting to wireless networks with Wi-Fi enabled devices such as laptops, Pda's I-phone etc. These devices are connected to wireless network with the help of Access point (AP). Wireless Access Points are usually routers or switches which are connected to internet by Ethernet port.

Wireless Access points are always required for infrastructure mode of wireless networking. It is necessary to use SSID while configuring AP, this SSID should be known to clients for their computers to connect WLAN. SSID is basically security key which help prevent UN authorized access to WLAN. The Access point is then connected to wire network (Internet) to provide wireless internet connectivity to clients. Multiple access points can be added in the WLAN, this increases the reach of infrastructure for supporting many number of wireless clients.

Setting Up Infrastructure Mode:

Setting up infrastructure mode for wireless connectivity is not that hard, all we are require follow number of steps which will do the job.

We need to make sure that Internet cable is plugged in Ethernet port, second step is to configure AP (Access point). http://192.168.1.1 is usually the access panel address. Write this URL at your Internet browser, username and password window will appear, default settings are usually, admin username and admin password. We name the connection name which will appear to all wireless devices, then we enable infrastructure mode and also assign SSID in the switch/router for it broadcast

Now we scan wireless adaptor to see if any wireless network is available to connect, when configured network is appear, click and connect and provide SSID information, here you should be able to connect to the network for internet browsing and file sharing.

3.5.6Ad hoc Mode

There is another mode of connectivity available for Wi-Fi connectivity. This mode is known to be **ad hoc mode.** By using ad hoc mode, devices are capable for communicating directly with each

other. No Access point (routers / switches) is required for communication between devices and all devices in the range connect in peer to peer communication mode.

Setting up Ad hoc Mode

For setting up ad hoc mode, wireless adaptors of all devices are required to be configured manually at ad hoc mode instead of infrastructure mode. It is also important to use same channel name and configuring all wireless adaptors using same SSID for making connection live.

Limitation of Ad hoc mode

Ad hoc mode is best used for small number of devices which are physically present in close proximity with each other and as the number of devices grows performance of network suffers. Disconnections of random device may occur time to time and managing the network can be difficult task for administrator. There is another limitation associated with ad hoc mode that is, ad hoc mode networks cannot bridge to wire LANs and cannot access internet without installing special gateways.

Advantages of Ad hoc network mode

Ad hoc mode networks works fine in small environment, like building, homes etc. No extra hardware (Access point) is required to use ad hoc mode, therefore it reduces the cost. If devices have wireless network adapters in them already then that will do the job as far as building ad hoc networks is concern. Ad hoc can be useful as back up option for time being if network based on infrastructure mode and access points are malfunctioning.

3.5.7 Advantages of Infrastructure mode (comparison between Ad hoc and infrastructure)

If we are to compare ad hoc with infrastructure mode then infrastructure mode provide much more stability, scalability, ease of management and improved security. Ad hoc on the other hand does not provide security to that level and managing can be difficult incase of network growth. Performance suffers as we increase devices as well.

The only disadvantage associated with infrastructure mode is extra cost to for Access points (routers and switches).Compared to the alternative, ad-hoc wireless networks, infrastructure mode networks offer the advantage of scalability, centralized security management and improved reach. The

disadvantage of infrastructure wireless networks is simply the additional cost to purchase AP hardware.

3.5.8 Step by step instructions for installing Wi-Fi

1. First of all you need to list the equipment and hardware thetas is required for installing Wi-Fi uh as router or compatible gateway different models are available one of the common is DI version 624 and Extreme G 802, 802.11 routers which are highly compatible with the Wi-Fi at present. Second is the wireless adapter that is required or connecting the network. wireless adapters also comes in many specification such as DWL G650 is card adapter that is compatible with the windows operating system and note books and Personal computers ,it is also equally reliable for the windows Xp home based systems.
2. Next you need a connection like broad band to access the internet which comes typically with the dialing modem or DSL modems.
3. In case of static IP address Ethernet cable is also required.
4. if your connection is DSL then you must require provide username and password by the service providers more specifically ISP
5. Another requirement is the usable Mac address that will support all the connected wireless network adapters.
6. It is recommended to keep a notebook and pen to write down the MAC and IP setting s and addressee because you may need it further during Wi-Fi configuration.

3.5.9 Installation of Wireless Adapters

Now install the wireless adapters by following the provide instructions.

1. Install wireless adapter in the system that are required to be connected to the attached router.
2. if you are using the Card adapter you may require to install the software for support before starting hardware installation. This require complete shutdown of the system, install adapter and reboot the complete computer to make it ready for installing wireless. To avoid the computer reboot it is recommended to use wireless adapters which are independent of additional software installations.
3. When system is restarted automatically new hardware wizard will run which will atomically activate the adapter and it become ready to use.

3.5.10Configuring Wi-Fi

If the system doesn't automatically detect the Wi-Fi network you may need to perform few more steps.

1. For configuring click networking icon that is available in the system tab
2. From appeared dialog box of **Wireless Network Connection** click "**advanced**" and choose the **Wireless network** tab.
3. Select my wireless network setting ok system will be restarted.
4. now set security parameters to activate the secure connection by accessing IP address to the browser
5. Configure the network for changing the name SSID by simply clicking wireless button present in the network setting in control panel.
6. Enable the Wi-Fi network encryption to stay protected from hackers and data stealers, wireless adapter offers WPA protected access encryption technique for the Wi-Fi security through the wired Equivalent Privacy encryption algorithms.
7. next step is the Mac address filtration .for extra safety apply the limit the Mac address by fix the router routines .enter Mac address in Wi-Fi adapter and save it
8. For improving the performance choose 802.11 for all the available wireless system in the network. This mode will help you speed up the communication.
9. Next step is saving and exit the step the wireless network is activated.

3.5.11 Features of "Wi-Fi" Technology - A new aspect in networking

Wi-Fi has brought a new aspect in the ground of networking. The broadcast of data is completed via radio waves and the cost of cables for network lying down.Wi-Fi enable a user to get access to internet anywhere in the given location. Now you can make a network in Hotels, Libraries, colleges, universities, campus, private institutes, and coffee shops and even on a public place to make your business more profitable and connect with their client any time. Wi-Fi makes waves for business with their highly effective cable less media.

WI-FI technology supports two types, one is called "infrastructure" other one is "Ad hoc" In ad hoc Wi-Fi network can be connected without central device known as router or access point Ad hoc mode is always preferred over infrastructure mode, however ad hoc networks have following issues Wi-Fi devices configure on Ad hoc mode offers nominal security against network intruders. Ad hock Wi-Fi configured devices cannot disable SSID broadcast in contrast to infrastructure mode. Network attackers will not required much of effort to prevail in Ad hoc Network.

Wi-Fi Limitations - Wireless network limitations:

WI-FI technology supports two types, one is called “infrastructure” other one is “Ad hoc” In ad hoc Wi-Fi network can be connected without central device known as router or access point Ad hoc mode is always preferred over infrastructure mode, however ad hoc networks have following issues Wi-Fi devices configure on Ad hoc mode offers nominal security against network intruders. Ad hock Wi-Fi configured devices cannot disable SSID broadcast in contrast to infrastructure mode. Network attackers will not required much of effort to prevail in Ad hoc Network.

Using Ad hoc mode signals issues can experienced where as using alternative infrastructure mode will provide full strength singles. Wi-Fi networking standard including 802.11g requires ad hoc mode of communication supports which 11Mbps bandwidth. Wi-Fi devices when configured to infrastructure mode can transfer data up to 54 Mbps, where as using ad hoc mode only 11 mbps can be achieved. Ad hoc mode is considered slower in comparison to infrastructure for this reason.

Security concerns

It is simple to set Wi-Fi network but keeping it secure takes much more effort, Access points of Wi-Fi do not deploy encryption methods. It is required to be done as network is enabled. Secure Wi-Fi network can be easily attacked by hackers to steal private information. Guests who are not potentially harmful can still utilize the network resources and minimize the performance.

Interference from other devices

Wi-Fi transmits data at 2.4 GHz making susceptible to interfere Bluetooth enabled devices, mobile phones, cordless, Microwaves and other communication devices, closer the interfering devices are the poor communication will be and vice versa.

Lacking high-quality media streaming

Today’s fastest Wi-Fi standards are pushed beyond their limit when trying to view high end media. High definition video and audios cannot be viewed flawlessly because of lower transfer rate; things can be much more worst if other clients are accessing the same access points.

Even the fastest current Wi-Fi standards are pushed beyond their limit when trying to handle some of today's high-end media. High-definition audio and video files are timely-delivery-intensive,

and typical wireless networks have neither the transfer speeds nor the consistency to transfer them flawlessly. This problem is further compounded if there are multiple devices connected to the same because the bandwidth must be divided between all of the equipment.

3.5.12 Specifications:

IEEE 802.11 is standard use world widely for Wireless Local Area network, WLAN, it stands for **Institute of Electrical and Electronics Engineers.** In simple words it is standard of communication between computers and wireless devices. The standards were set by IEEE LAN/MAN Standard committee in 5GHZ and 2.4 GHz public spectrum bands.

The IEEE 802.11 is consisting of numerous mechanism and services that interrelate to present station mobility translucent to the higher layers of the network mass. IEEE 802.11 is a combination of wireless LAN. We can use IEEE 802.11 and Wi-Fi often as interchangeably depends on market demand. It is a basic protocol of any high speed wireless network from 1997 to 2008. There are lots of change occurred in it but the basics “high performance" never change. The frequency of it is from 2.4 GHz to 5 GHz and range is from 100 meter to 5000 meter. It specifies a single medium access control as MAC and three physical layers and offering different service such as Authentication of system, Deauthentication of system, data privacy, MSDU delivery, Association with files and system, Disassociation, Distribution of data, Integration of information, and Reassociation. You can configure a station with IEEE 802.11 in different ways as independent configuration, and infra-structure configuration. The IEEE 802.11 starts from the need to connect more than one computer with each other wirelessly especially where the wire cannot go.

IEEE 802.11 required accepting device to activate in a peer-to-peer fashion within the possibly of overlie same as access control level and data transformation services to allow top layers .There are several physical layer represent signaling methods and interfaces. Solitude and safety of user data is being moved over the wireless media via IEEE 802.11.According to OSI model an IEEE 802.11 protocol having direct frequency, hopping sequence, spread spectrum PHY and infrared PHY. IEEE 802.11 specifies a single medium access sub layers called MAC and physical layers with different specifications. The first is physical layers having two radio and infrared. The physical layer contain Frequency Hopping Spread Spectrum Radio PHY offering 1 Mbit/s with two level and 2 Mbit/s with 4 level, As it is Direct Sequence Spread Spectrum Radio PHY provides DBPSK and DQPSKwith 1 and 2 Mbit/s operation**,** Infrared PHYoffer 1 Mbit/s with16-PPM and 2 Mbit/s with –PPM.

These days 802.11a, 802.11b, and 802.11g are mostly in use to create a network home, offices and commercial locations. The architecture of IEEE 802.11 consists of 802.1X for verification, RSN for maintenance track of links, and AES-based CCMP to provide privacy, veracity and basis. IEEE 802.11with the propagation of modems and DSL services makes it reliable. People always desire to set up small networks in their homes to distribute their elevated rate Internet connection. There are many free networks commonly permit anyone within specific range, counting passersby outer, to connect to the Internet. There is also hard work by unpaid helper groups to launch wireless community networks to provide free wireless connectivity to the public. Networking without wire has a talented prospect with 802.11.The dynamic nature of IEEE 802.11 increasing radically in the near future.

3.5.13 Description and working of Wi- Fi module:

There are three most important items which makes Wi-Fi working in your laptop or desktop. These are:

- Radio Signals
- Wi-Fi Card which fits in your laptop or computer.
- Hotspots which create Wi-Fi Network.

Radio Signals:

Radio Signals are the keys which make Wi-Fi networking possible. These radio signals transmitted from Wi-Fi antennas are picked up by Wi-Fi receivers such as computers and cell phones that are equipped with Wi-Fi cards. Whenever a computer receives any of the signals within the range of a Wi-Fi network which is usually 300 - 500 feet for antennas, the Wi-Fi card will read the signals and thus create an internet connection between the user and the network without the use of a cord.

Access points which consist of antennas and routers are the main source which transmit and receive radio waves. Antennas work stronger and have a longer radio transmission with a radius of 300-500 feet which are used in public areas while the weaker yet effective router is more suitable for homes with a radio transmission of 100-150 feet.

Wi-Fi Cards:

You can think Wi-Fi card as being an invisible cord that connects your computer to the antenna for a direct connection to the internet.

Wi-Fi cards can be external or internal, meaning that if a Wi-Fi card is not installed in your computer, you may purchase a USB antenna attachment and have it externally connect to your USB port, or have an antenna-equipped expansion card installed directly to the computer. For laptops, this card will be a PCMCIA card in which you insert to the PCMCIA slot on the laptop.

Wi-Fi Hotspots:

A Wi-Fi hotspot is created by installing an access point to an internet connection. The access point transmits a wireless signal over a short distance typically covering around 300 feet. When a Wi-Fi .enabled device, such as a Pocket PC, encounters a hotspot, the device can then connect to that network wirelessly.

Most hotspots are located in places that are readily accessible to the public, like airports, coffee shops, hotels, book stores and campus environments. 802.11b is the most common specification for hotspots worldwide. The 802.11g standard is backwards compatible with .11b but .11a uses a different frequency range and requires separate hardware such as an a, a/g, or a/b/g adapter. The largest public Wi-Fi networks are provided by private internet service providers (ISPs) that charge a fee for users to connect to the internet.

Hotspots are increasingly developing around the world. In fact, T-mobile USA controls more than 4,100 hotspots located in public locations such as Starbucks, Borders, Kinko.s, and the airline clubs of Delta, United, and US Airways. Even select McDonald.s restaurants now feature Wi-Fi hotspot access.

Any notebook computer with integrated wireless, a wireless adapter attached to the motherboard by the manufacturer, or a wireless adapter such as a PCMCIA card can access a wireless network. Furthermore, all Pocket PCs or Palm units with Compact Flash, SD I/O support, or built-in Wi-Fi, can access hotspots.

Some Hotspots require WEP key to connect that is the connection is considered to be private or secure. As for open connections, anyone with a Wi-Fi card can gain access to that hotspot. So in order for a user to gain access to the internet under WEP, the user must input the WEP key code.

Working of Wi-Fi:

Wi-Fi works with no physical wired connection between sender and receiver by using radio frequency (RF) technology, a frequency within the electromagnetic spectrum associated with radio wave propagation. When an RF current is supplied to an antenna, an electromagnetic field is created that then is able to propagate through space. The cornerstone of any wireless network is an access point (AP). The primary job of an access point is to broadcast a wireless signal that computers can detect and "tune" into. In order to connect to an access point and join a wireless network, computers and devices must be equipped with wireless network adapters

Wi-Fi Support:

Wi-Fi is supported by many applications and devices including video game consoles, home networks, PDAs, mobile phones, major operating systems, and other types of consumer electronics. Any products that are tested and approved as "Wi-Fi Certified" (a registered trademark) by the Wi-Fi Alliance are certified as interoperable with each other, even if they are from different manufacturers. For example, a user with a Wi-Fi Certified product can use any brand of access point with any other brand of client hardware that also is also "Wi-Fi Certified". Products that pass this certification are required to carry an identifying seal on their packaging that states "Wi-Fi Certified" and indicates the radio frequency band used (2.5GHz for 802.11b, 802.11g, or 802.11n, and 5GHz for 802.11a).

3.5.14 Differences between Bluetooth and Wi-Fi:

Wi-Fi and Bluetooth are the most famous wireless technology that creates and managed wireless network with the of radio frequency waves. both them have the same mechanisms through which they develop the wireless networks for the organizations adopting but still there are some differences which make them different in use.

Coverage difference:

The major purpose of the Bluetooth technology is to create the connection between the user device and network over the short distances. Their ranges are limited to the one meter or 2 meter hardly. The device outside this renege cannot be connected with the help of Bluetooth where as Wi-Fi has no such small distance limitation as compared to Bluetooth devices. Wi-Fi has the ability to create and provide its users and wireless internet facility over the long distances, they can link multiple computer at the time. Although Bluetooth can

also connect multiple device at a time but coverage area difference is huge which makes Wi-Fi a better approach than Bluetooth?

Structure:

The 802.11 standard defines the lower layers of the OSI model for wireless communication using electromagnetic waves, that is to say:

The physical layer offering three types of information coding;

- The data link layer, composed of two sub-layers:
- The Logical Link Control or LLC.
- The Media Access Control or MAC.

The physical layer defines the modulation of radio waves and the characteristics of the signal for transmission of data, while the data link layer defines the interface between the bus of the machine and the physical layer, including a method access similar to the one used for a standard Ethernet network and the communication rules between the different stations. 802.11 actually propose three physical layers, defining alternative modes of transmission:

It is possible to use any protocol over a wireless Wi-Fi network as well as an Ethernet network.

The modes:

Infrastructure mode:

Infrastructure mode allows you to connect computers equipped with a wireless network adapter with each other via one or more access points (AP), which will assume the role of a hub.

This mode is mainly used in the business environment. The establishment of such a network requires the setting up of terminals (AP) at regular intervals over the area to be covered by the network. The terminals and the machines must be configured with the same SSID (network name) in order to communicate.

The advantage of this mode is that it allows you to check out who enters the network (Connection trough AP is compulsory). The cons: the network can only be expanded by installing new terminals.

Ad-Hoc

The Ad-Hoc mode is used to connect (directly) computers equipped with a wireless network card (without using a third-party hardware such as an access point (AP)) to the internet. This mode is ideal for connecting machines together quickly without additional hardware (eg: file sharing between phones, sharing Internet access).

The establishment of such a network needs you to configure the machines in Ad-Hoc mode (instead of infrastructure mode), selecting channel (frequency) and SSID (network name) common to all and the job is done. The advantage of this mode is to avoid costly third-party hardware and it is easier to implement. With the addition of a simple dynamic routing software (Ex: OLSR, AODV...) the network grows naturally (enabling the connection of new machines).

Creation of network:

Bluetooth Technology

Both Wi-Fi and Bluetooth are the network creation technologies PAN Personal Area Networks are created by the Bluetooth by utilizing radio wave so flow frequency which lows the Bluetooth device to instantly connect the c0onntion device in the specific are to develop a pair. For instant cell phone with blue tooth technology, ipods and PDAs.

Most frequency Bluetooth is used for immediately sending files for m laptop to laptop desktop, hand set to hand set, and hand set to Desktop from the source like printer .when the device that you are using for the communication does not contain the Bluetooth feature then it contains the built in adapters to support the transfer feature. Most common of thee adapters are the USB's Universal Serial Buses that are extensive in use for carrying data.

Wi-Fi Technology:

Wi-Fi use the same radio frequency technology. But it has one main advantage over Bluetooth which is the dynamic protocol which can connect various computers over long distance through wireless. Wi-Fi permit the computer within the wireless network to communicate and send message without any plugged cable. It just matches the radio frequency omitted by the service provider boosters which ultimately save money and time effort. Because huge installation are not required Computer that uses Local area Networks Wi-Fi can share, send files, codes and document over the Wi-Fi network and can also share the resource like printers, fax machines and scanning devices .internet facility and also be shared and no specific type of LAN is required for this purpose too.

File swapping

File swapping is enjoyable task when using Wi-Fi Bluetooth or button devices both of them can easily transfer the related files to your nearby computer system any time by activating wireless connection. No

complex installation and configuration is required for this purpose**.** Both of them only require some allocation of resource for sharing the files.

Protocol specification

There is no extensive protocol specification for the Wi-Fi and Bluetooth, both of them follow the typical standards designed by the manufacturer to become compatible with the hardware and to communicate with interoperable products the current protocol standard for the Bluetooth is 2.0 and for Wi-Fi standard for networking protocol is 802.extension to 802, 802.11 is also in great fame these days.

Support difference

Another important difference between Wi-Fi and Bluetooth is the support. Wi-Fi is supported by the modem because it has to cover the wider area n sense of range and Bluetooth is supported by the router that is available in the small carrying communications.

Network security

The main issue with wireless network security is its simplified access to the network compared to traditional wired networks such as Ethernet, with wired networking one must either gain access to a building (physically connecting into the internal network) or break through an external firewall. To enable Wi-Fi, one merely needs to be within the wireless range of the Wi-Fi network. Most business networks protect sensitive data and systems by attempting to disallow external access. Enabling wireless connectivity reduces security if the network uses inadequate or no encryption.

An attacker who has gained access to a Wi-Fi network router can initiate a DNS spoofing attack against any other user of the network by forging a response before the queried DNS server has a chance to reply

Securing methods

A common measure to deter unauthorized users involves hiding the access point's name by disabling the SSID broadcast. While effective against the casual user, it is ineffective as a security method because the SSID is broadcast in the clear in response to a client SSID query. Another method is to only allow computers with known MAC addresses to join the network, but determined eavesdroppers may be able to join the network by spoofing an authorized address.

Wired Equivalent Privacy (WEP) encryption was designed to protect against casual snooping but it is no longer considered secure. Tools such as AirSnort or Aircrack-ng can quickly recover WEP encryption keys. Because of WEP's weakness the Wi-Fi Alliance approved Wi-Fi Protected Access (WPA) which uses TKIP. WPA was specifically designed to work with older equipment usually through a firmware upgrade. Though more secure than WEP, WPA has known vulnerabilities.

The more secure WPA2 using Advanced Encryption Standard was introduced in 2004 and is supported by most new Wi-Fi devices. WPA2 is fully compatible with WPA.

A flaw in a feature added to Wi-Fi in 2007, called Wi-Fi Protected Setup, allows WPA and WPA2 security to be bypassed and effectively broken in many situations. The only remedy as of late 2011 is to turn off Wi-Fi Protected Setup, which is not always possible.

Piggybacking

Piggybacking refers to access to a wireless Internet connection by bringing one's own computer within the range of another's wireless connection, and using that service without the subscriber's explicit permission or knowledge.

During the early popular adoption of 802.11, providing open access points for anyone within range to use was encouraged to cultivate wireless community networks, particularly since people on average use only a fraction of their downstream bandwidth at any given time.

Recreational logging and mapping of other people's access points has become known as war driving. Indeed, many access points are intentionally installed without security turned on so that they can be used as a free service. Providing access to one's Internet connection in this fashion may breach the Terms of Service or contract with the ISP. These activities do not result in sanctions in most jurisdictions; however, legislation and case law differ considerably across the world. A proposal to leave graffiti describing available services was called warchalking. A Florida court case determined that owner laziness was not to be a valid excuse.

Piggybacking often occurs unintentionally, since most access points are configured without encryption by default and operating systems can be configured to connect automatically to any available wireless network. A user who happens to start up a laptop in the vicinity of an access point may find the computer has joined the network without any visible indication. Moreover, a user

intending to join one network may instead end up on another one if the latter has a stronger signal. In combination with automatic discovery of other network resources (see DHCP and Zeroconf) this could possibly lead wireless users to send sensitive data to the wrong middle-man when seeking a destination. For example, a user could inadvertently use an unsecure network to log in to a website, thereby making the login credentials available to anyone listening, if the website uses an unsecure protocol such as HTTP.

Safety

The World Health Organization (WHO) says "there is no risk from low level, long-term exposure to wi-fi networks" and the United Kingdom's Health Protection Agency reports that exposure to Wi-Fi for a year results in the "same amount of radiation from a 20-minute mobile phone call."

A small percentage of Wi-Fi users have reported adverse health issues after repeat exposure and use of Wi-Fi, though there has been no publication of any effects being observable in double-blind studies. A review of studies involving 725 people that claimed electromagnetic hypersensitivity found no evidence for their claims.

One study claims, in preliminary results, that "laptops (Wi-Fi mode) on the lap near the testicles may result in decreased male fertility". Another small sample study (15 males and 15 females) found a decrease in the P300 event-related potential (ERP) for male but not female subjects while engaging in a working memory oriented linguistic task during Wi-Fi exposure.

Features of Wi-Fi Technology:

- **Ease of Use in Wi-Fi Technology:**

Wi-Fi is a new characteristic in networking and has fetched a new feature in the field of networking. The data broadcasting is finished by means of radio waves. Through **Wi-Fi Technology** a user can easily get access to internet for objective to sharing all around the world. Now establishing a computer network in any type of business is not difficult such as companies, coffee shops, Libraries, campus, Hotels, colleges, universities, , private institutes etc. Wi-Fi Technology enabled you to get more profit from your business and facilitate you to connect your client anywhere or any time.

- **No need for cabling in Wi-Fi Technology:**

Wi-Fi is creating new strength to connect any thing or any task without giving up task. There are lots of utilities initiated by Wi-Fi. Music streamers are a great utility by Wi-Fi Technology that put your music to speakers without any cable. Wi-Fi Technology also enables you to play radio online and enjoy music via remote computer. Downloading of songs, sharing of files, above the ground speed makes Wi-Fi Technology matchless mobility. Wi-Fi has no cable so you can easily move your pc from one location to another because it is very helpful, easy and convenient.

- **Wi-Fi Fortress Technology Features:**

Wi-Fi is offering **Fortress Technology** to secure wireless resolution hold up the intensification. The advantages of Wi-Fi are more than a few. It supports a group to create a network between same types of networks for transformation of data. Wi-Fi make easy to attach different devices to your pc such as PDA's, games, MP3 player, Radio etc. It is an expedient technology therefore you can use internet during traveling and set up your business at any location and if you are existing near Wi-Fi hotspot then you will automatically connect to internet.

- **Wi-Fi Technology Secure Internet Connection Features**

Nowadays Wi-Fi continue living all over the place with all its wonder. Through Wi-Fi user can get quick access to internet, build up a network in a small place doesn't matter now because it don't need any professional installation. Just make sure the wireless connection and ready to go of fast speed internet journey depending on the speed of your internet provider. The security system of Wi-Fi Technology make it more reliable and its tool protect your data .The configuration of devices are very easy because it has network security, embedded system and standard devices.

- **Wi-Fi Feature of No Limitation:**

Wi-Fi Technology connects you worldwide and it has no limitation. Wi-Fi Technology has great power for consumption as compared to other network. You can get high speed internet. Its site surveys help you to know about it in short we can say that feature of Wi-Fi such as high speed internet access, Site Surveys is a great deal of Wi-Fi Technology to make sure visual presentation, safety and fulfillment. Business enterprising tools make it a pioneering inventor of technology solutions.

- **Wi-Fi Features of Scalable System and Robust Performance:**

The scalable system of Wi-Fi Technology shows performance in a while rather than days and makes exact result. Flexible authentication methods facilitate the transportation and procedure of keying substance. You never need any experience to install devices because Wi-Fi Technology itself smart enough to provide security to your personal computer but also offer an easy way of network installation.

3.5.15 Advantages:

One of the most prevalent advantages of Wi-Fi wireless LAN technologies is that it is completely wire-free. Now, if you want to sit in one of your comfort zones in you house, such as a couch or in your yard, you can carry your laptop with you and still be able to access the internet. These days, if you buy a new laptop, a built-in Wi-Fi card will allow you automatic allowance of use. (Brain) A second advantage is becoming increasingly popular: hotspots. Instead of having to be constricted to your home wired/wireless Local Area Network (LAN), you can now enjoy the advantages of the Internet in public places such as lobbies, cafes, universities, hotels, airports, and many other common areas.

Wi-Fi also supports roaming: you can walk around a building from one access point to another. A third advantage of Wi-Fi is the ability for 802.11b and 802.11g to frequency hop. This process allows the 802.11b and 802.11g cards to transmit themselves on any of three bands, or splitting the radio bandwidth into channels and 'hop' between them. (Brain) This enables the Wi-Fi cards to talk at the same time without interference. The fourth advantage I know of is the development of the Wi-Fi Alliance (formerly WECA). This association is formed with more than 200 member companies whose business it is to test and certify the interoperability of WLAN products, making sure they are based solely on the IEEE 802.11 specifications. (Wi-Fi Alliance, What) You also must remember that Wi-Fi products are widely distributed, and an easy technology to use.

Advantages of Wi-Fi:

*Many Wi-Fi networks support roaming, in which a mobile client station such as a laptop computer can move from one access point to another as the user moves around a building or area.

* Many access points and network interfaces support various degrees of encryption to protect traffic from interception. Wi-Fi is a global set of standards. Unlike cellular carriers, the same Wi-Fi client works in different countries around the world.

* Allows LANs to be deployed without cabling, potentially reducing the costs of network deployment and expansion. Spaces where cables cannot be run, such as outdoor areas and historical buildings, can host wireless LANs.

* Wi-Fi silicon pricing continues to come down, making Wi-Fi a very economical networking option and driving inclusion of Wi-Fi in an ever-widening array of devices.Wi-Fi products are widely available in the market. Different brands of access points and client network interfaces are interoperable at a basic level of service. Products designated as Wi-Fi CERTIFIED by the Wi-Fi Alliance are interoperable and include WPA2 security.

* Widely available in more than 100 thousand public hot spots and millions of homes and corporate campuses worldwide.

* Wi-Fi networks support roaming, in which a mobile client station such as a laptop computer can move from one access point to another as the user moves around a building or area.Wi-Fi is a global set of standards. Unlike cellular carriers, the same Wi-Fi client works in different countries around the world.

- Wi-Fi uses unlicensed radio spectrum and does not require regulatory approval for individual deployers.
- It allows local area networks (LANs) to be setup with cabling. The can reduce associated costs of network connection and expansions. Places where cables cannot be run, such as outdoor areas and historical buildings can use wireless LANs.
- Wi-Fi products are extensively available in the market. There are different brands of access points and user's network interfaces are able to inter-operate at a very basic service level.
- Prices are considerably lower as competition amongst vendors' increases.
- Wi-Fi networks can support roaming. This allows mobile users with laptop computer to be able to move from one access point to another.
- Numerous access points and network interfaces support various degrees of encryption to protect traffic from interception.
- Wi-Fi has a set of global standards. Not like the cellular carriers, the same Wi-Fi users can work in different countries around the world at all time.

3.5.16 Disadvantages:

Even though Wi-Fi products are useful, there are still some downfalls that must be stated in order to get a firm grasp on exactly what Wi-Fi wireless LAN technologies are composed of. As Morley points out, "Wi-Fi is designed for medium-range data transfers, and most versions of 802.11 works up to about 250-300 feet away from the access point indoors, and about 1,000 feet away outdoors."(341) yes, and with more distance between your computer/laptop and the access point, the speed and the quality falters tremendously.

It also doesn't help when there is interference from microwaves or cordless phones which use the same frequency that 802.11g and 802.11b use: 2.4 GHz. Another disadvantage for Wi-Fi products is their security system. The Wired Equivalent Privacy (WEP) is the common wireless encryption standard which is easily broken even when configured accurately. (Wikipedia, Wi-Fi) To counteract this problem, however, Wi-Fi Protected Access (WPA) has been established.

Due the fact that Wi-Fi are still relatively new, there are considerably more disadvantages to users. Let's have a look at them:

- The use of Wi-Fi band that is 2.4 GHz does not require a license in most countries provided that is stays below limit of 100mW and one accepts interference from other sources; including interference which causes the users devices to no longer function.
- The spectrum assignments and operational limitations are not consistent worldwide.
- Power consumption is fairly high compared to some other standards, making the battery life and heat a concern to some users.
- Wi-Fi uses the unlicensed 2.4GHz spectrum, which often crowded with other devices such as Bluetooth, microwave ovens, cordless phones, or video sender devices, and among many others. This may cause degradation in performance.
- Wi-Fi networks have limited range. A typical Wi-Fi home router might have a range of 45m (150ft) indoors and 90m (300ft) outdoors. Ranges may also vary as Wi-Fi is no exception to the physics of radio wave propagation with frequency band.
- The most common wireless encryption standard, wired equivalent privacy or WEP has been shown to be breakable even when it has been correctly configured.
- Access points could be used to steal personal and confidential information transmitted from Wi-Fi consumers.
- Intervention of a closed or encrypted access point with other open access points on the same or a nearby channel can prevent access to the open access points by others in the area. It poses a high

problem in high-density areas such as large apartment blocks where many residents are operating Wi-Fi access points.

- Inter-operability issues between brands or deviations can cause limited connection or lower output speeds.
- Free access points can be used by the malicious to anonymous to initiate an attack that would be extremely difficult to track beyond the owner of the access point.

3.5.17 Limitations:

Spectrum assignments and operational limitations are not consistent worldwide: most of Europe allows for an additional two channels beyond those permitted in the US for the 2.4 GHz band (1–13 vs. 1–11), while Japan has one more on top of that (1–14). As of 2007, Europe is essentially homogeneous in this respect.

A Wi-Fi signal occupies five channels in the 2.4 GHz band. Any two channel numbers that differ by five or more, such as 2 and 7, do not overlap. The oft-repeated adage that channels 1, 6, and 11 are the *only* non-overlapping channels is, therefore, not accurate. Channels 1, 6, and 11 are the only group of three non-overlapping channels in the U.S. In Europe and Japan using Channels 1, 5, 9, and 13 for 802.11g and n is recommended.

Equivalent isotropic ally radiated power (EIRP) in the EU is limited to 20 dBm (100 mW).

The current 'fastest' norm, 802.11n, uses double the radio spectrum/bandwidth (40 MHz) compared to 802.11a or 802.11g (20 MHz). This means there can be only one 802.11n network on the 2.4 GHz band at a given location, without interference to/from other WLAN traffic. 802.11n can also be set to use 20 MHz bandwidth only to prevent interference in dense community.

Data security risks:

The most common wireless encryption-standard, Wired Equivalent Privacy (WEP), has been shown to be easily breakable even when correctly configured. Wi-Fi Protected Access (WPA and WPA2) encryption, which became available in devices in 2003, aimed to solve this problem. Wi-Fi access points typically default to an encryption-free (open) mode. Novice users benefit from a zero-configuration device that works out-of-the-box, but this default does not enable any wireless security, providing open wireless access to a LAN. To turn security on requires the user to configure the

device, usually via a software graphical user interface (GUI). On unencrypted Wi-Fi networks connecting devices can monitor and record data (including personal information). Such networks can only be secured by using other means of protection, such as a VPN or secure Hypertext Transfer Protocol (HTTPS) over Transport Layer Security.

Interference:

Wi-Fi connections can be disrupted or the internet speed lowered by having other devices in the same area. Many 2.4 GHz 802.11b and 802.11g access-points default to the same channel on initial startup, contributing to congestion on certain channels. Wi-Fi pollution, or an excessive number of access points in the area, especially on the neighboring channel, can prevent access and interfere with other devices' use of other access points, caused by overlapping channels in the 802.11g/b spectrum, as well as with decreased signal-to-noise ratio (SNR) between access points. This can become a problem in high-density areas, such as large apartment complexes or office buildings with many Wi-Fi access points.

Additionally, other devices use the 2.4 GHz band: microwave ovens, ISM band devices, security cameras, ZigBee devices, Bluetooth devices, video senders, cordless phones, baby monitors, and (in some countries) Amateur radio all of which can cause significant additional interference. It is also an issue when municipalities or other large entities (such as universities) seek to provide large area coverage.

Applications:

There are many **applications of Wi-Fi** connectivity; you can take advantage of it in many different ways. Home-based Wi-Fi enabled device helps you to connect your PC, game console or laptop. If 2 or 3 members of your family have their own laptops they would be able to share their data, files and photos. They can take prints by connecting their laptops to the PC or through print server. Moreover there is no mess of wires and your surrounding remains clean.

There are no boundaries if you are using Wi-Fi, you move from one room to another or even away from home you have the liberty to access internet within the range of radial distance. Just go for security measurements.

In an office set up, all the computers can be interconnected and enable check and balance whether employs are working or not. A project can be presented to all the members whether in the form of excel sheet, ppt or word document.

Also you can share windows if you are delivering lecture, it would help you make your lecture effective as they not only hear you but also receiving the visual information. Many public places are also supplied with Wi-Fi connectivity to access internet like café, restaurants, and clubs to attract the clients. A prospect customer surely appreciates your efficiency and awareness of the modern fast-paced world.

Wi-Fi hotspots concept is getting popularity among business communities and mobile workers. For this purpose ISPs are incorporating Wi-Fi routers to many places for the coverage of wide range. The accessibility of internet is somewhere free and somewhere you have to pay for internet connectivity per hour or of the package for which you go. These routers are the access point for internet while attached to the DSL modem or cable modem.

Now-a-days, computers, laptops and mobiles are Wi-Fi oriented so two computers or two laptops are easily connected without access point. This is the most modern Wi-Fi technology. A prospect businessman surely utilizes this Wi-Fi technology to increase the workers' output. You can produce efficient workforce to enhance your business.

A whole city can provide Wi-Fi connectivity by deploying routers at specific rang to access the internet. It seems unbelievable and even inapplicable in the beginning but now in European countries some of the cities are wholly Wi-Fi.

It is now also a very common **applicationof Wi-Fi** that a whole campus is having a single access point at a certain height and all of the student can take benefits from it. For this they have to pay a monthly fee through their monthly fee-bill.

Wi-Fi is not limited to private users or small-business communities. Its future is predictable as Wi-Fi is applicable in a versatile modes and profitable ways. Everyone whether starting a home-based business or a world-wide mobile business is heading for the Wi-Fi technology to prosper their business by leaps and bounds.

Benefits

The benefits of Wi-Fi are to enable wireless access to deliver voice, data, and video streams with no cables attached. The benefits of Wi-Fi include:

- Enable staff to carry portable Wi-Fi handsets giving them the ability to take calls anywhere, anytime
- Using Wi-Fi enables PDA's and Laptops to deliver quick and easy access to information
- Eliminate cellular usage charges
- Eliminate cabling and wiring for PC's
- Eliminate switches, adapters, plugs, pins, and connectors

When looking to implement a Wi-Fi (WAN) system, it is important to ensure that the system has a few important features which include:

- Secure networking using the latest secure networking protocols
- Simplified management software
- Simultaneous Dual Mode radios allowing fully segmented voice/ data/ video traffic across the entire network
- Radio Frequency sweeping technology that disables rogue and ad hoc wireless devices not relevant to the network
- Fast roaming and seamless data packet transfer
- Quality of Service or QOS

A wireless LAN (WAN) system needs to be capable of delivering high-quality voice, multi-media and business applications while meeting stringent security requirements, delivering seamless mobility functionality, easily, and cost effectively. It must also be quick to install and simple to operate.

Also Communication Products Ltd. offers industry leading wireless solutions, delivering secure mobility, the broadest range of services, and the lowest total cost of ownership of any WLAN systems.

One thing is certain; the promise of Wi-Fi is that it will continue to deliver a solution with an increasingly compelling business proposition. As Wi-Fi evolves, it will become difficult to justify making an investment in wired technology when wireless will get you everything you need for less money, increased flexibility, with as good or even better security than legacy wired systems.

Uses:

Three quarters of us now have a home Wi-Fi network, which means we can access the internet from any room in the house. Convenient though this might be, it's barely scratching the surface of what's possible from wireless internet. Our aim in this feature is to help you unleash that potential. Our guide will show you how to better use what you already have. So, for example, without spending a penny you can share files between computers and print wirelessly from your laptop.

If you are prepared to invest in some extra kit, though, things just get better. A NAS drive is a worthwhile addition to your wireless network, as we'll demonstrate. While you probably think of a Wi-Fi network as a computing resource, its potential goes far beyond. We'll also look at how you can use your network for home-entertainment and -automation. In particular, we'll explain how to stream audio around the house, control your home-entertainment system from a phone or tablet, and even keep an eye on what's occurring via IP cameras.

If you start to use your wireless network for more than just web browsing, though, you might find it starts to run out of steam. If this is the case, see our feature on how to get better Wi-Fi speed and range to cope with the increased demand.

1. Share your internet

If you're new to networking, your PC may be connected with an ethernet cable to a Wi-Fi router and you have yet to take advantage of wireless connectivity in the home. As a first step, then, we'll explain how to wirelessly access the internet on a laptop.

If you're going to switch on wireless internet in your home, you should be aware that the signal may also be accessible to your neighbours. Allowing others to piggyback on your internet connection can slow it down, and it can open access to your personal files.

Modern laptops have Wi-Fi connectivity built in. If yours doesn't, USB adaptors that add this functionality are very cheap and easy to find. Most such adaptors support the current 802.11n wireless standard, but will be backwards-compatible with older 802.11g. If your router is an 802.11g model then consider upgrading for faster wireless performance. See our guide to Wi-Fi standards for an explanation of these terms.

Wirelessly connect your laptop to the router, click the Wireless Networks icon in the Windows 7 taskbar. In XP, this icon depicts a PC and a red cross.

Find and select your router in the list that appears. If you don't know its name, look on the rear panel or original packaging for an SSID. The network name might include your router manufacturer's name and a string of numbers or letters. Typically, your router will be at the top of the list, which is ordered by signal strength.

Enable Connect automatically, which will allow your laptop to automatically connect to the internet in future, and then click Connect. You'll be prompted for a password (again, if you don't know this, look on the router's rear panel or original packaging, or search the manual for a default code).

Find and select your router in the list that appears. If you don't know its name, look on the rear panel or original packaging for an SSID. The network name might include your router manufacturer's name and a string of numbers or letters. Typically, your router will be at the top of the list, which is ordered by signal strength.

Enable Connect automatically, which will allow your laptop to automatically connect to the internet in future, and then click Connect. You'll be prompted for a password (again, if you don't know this, look on the router's rear panel or original packaging, or search the manual for a default code).

You are now connected to the internet. Launch your web browser to begin wirelessly surfing the web.

2. Share resources between PCs

The most common use for a wireless network is to share resources such as files and printers between computers. There are several methods of achieving this, but the easiest solution is to use Homegroup. Setting up a Homegroup in Windows 7 or 8 is straightforward, and doesn't require additional hardware. See How to create a Homegroup network in Windows 7 and Windows 8.

If one or more of your computers is running an earlier version of Windows that doesn't support Homegroup, you can still get them to talk to each other. See How to network Vista and Windows 7 PCs to find out how.

Having set up a Homegroup, it's easy to share or un-share folders, documents and printers between your PCs. To share a folder or document, simply right-click it in Windows Explorer and select Share with from the drop-down menu. You should then choose to share it with Homegroup, opting to provide either read-only or read-and-write access.

To share a printer, select 'Devices and Printers' from the Start menu, double-click the printer you want to share, then click 'Display Printer Properties'. Select the Sharing tab in the Properties dialog box for your printer and choose 'Share this printer'.

You should now be able to see in Windows Explorer any files and printer(s) you've shared on other computers in the Homegroup (or on the network if you're not using Home group).

3. Access your files via a NAS drive

NAS stands for network-attached storage and is, in essence, a hard disk that is connected to a network rather than a PC. You can share files and folders on any PC connected to a wireless network, but using a NAS provides a better solution in several respects.

First, if you have computers and mobile devices running various operating systems, a NAS could share files with them all – including over the internet. Without a NAS, sharing files among these devices would be not only time-consuming, but potentially impossible (with certain combinations of OS).

Second, using a NAS means you don't have to leave switched on 24/7 any PC that contains shared files. Although the NAS itself will be constantly powered on, it is designed to operate in this

manner. A NAS drive will consume significantly less power than your desktop PC, which is good for the environment and your wallet.

Most NAS drives can also download files from the internet, and have other functions including a print server, which will be accessible to any PC at any time.

Third, storing all your files – documents, videos, photos, music and more – on a NAS drive provides one central repository, rather than you having to remember on which machine a particular file is stored.

NAS devices cost from as little as £50, although you tend to get what you pay for in terms of capacity, performance and features.

A decent NAS device that includes a terabyte of storage will cost £80-plus. In addition to pre-built NAS drives with a certain storage capacity, it's possible to buy the enclosure alone, and then add your own hard disks. Synology's Disk Station range is particularly good in this respect.

Some NAS devices operate wirelessly, but we recommend hardwiring one to your wireless router using an ethernet (or network) cable. This will provide a faster connection than a wireless link, and it won't unnecessarily clog up your Wi-Fi network.

Even though the NAS itself is connected to the network via a cable, all your PCs will be able to wirelessly access its files through the router.

3.5 The Raspberry Pi Camera Module

Introduction:

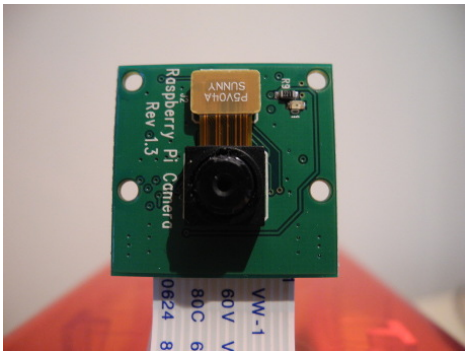
In this post we'll look at the Raspberry Pi camera module, a low cost hardware module that supports still picture and video recording and is the first official hardware add-on for the Raspberry Pi. We'll also look at the camera's features and specifications, how to set it up and examine the software that supports it.

This post is based on a lightning talk I gave at Qt Developer Days 2013, updated with some additional information since then.

Camera Details

Since 2012, the Raspberry Pi Foundation had been reporting that an official camera module was in development. In May 2013, an announcement was made by RS Components and Premier Farnell/Element 14, distribution partners of Raspberry Pi, that the camera module was available (it is also available from other sources) and sells for retail €30 or US$25.

The camera consists of a small (25mm by 20mm by 9mm) circuit board, which connects to the Raspberry Pi's Camera Serial Interface (CSI) bus connector via a flexible ribbon cable. The camera's image sensor has a native resolution of five megapixels and has a fixed focus lens. The software for the camera supports full resolution still images up to 2592x1944 and video resolutions of 1080p30, 720p60 and 640x480p60/90. The camera module is shown below:



Installation involves connecting the ribbon cable to the CSI connector on the Raspberry Pi board. This can be a little tricky, but if you watch the videos that demonstrate how it is done, you shouldn't have any trouble.

The OV5647 is a low voltage, high performance, 5 megapixel CMOS image sensor that provides 2592x1944 video output using OmniBSI™ technology. It provides multiple resolution raw images via the control of the serial camera control bus or MIPI interface.

The OV5647 has an image array capable of operating up to 15 fps in 2592x1944 resolution with user control of image quality, data transfer, camera functions through the SCCB interface. The OV5647 uses innovative OmniBSI technology to improve the sensor performance without the physical and optical trade-off.

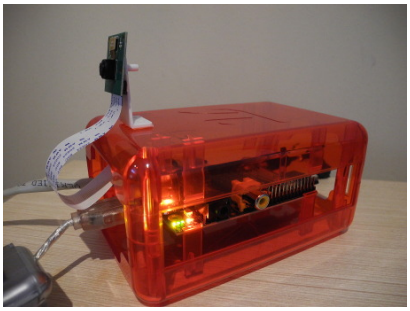
For customized application, the OV5647 includes a one-time programmable (OTP) memory.

Features:

- 1.4 µm x 1.4 µm pixel with Omni BSI technology for high performance (high sensitivity, low crosstalk, low noise)
- optical size of 1/4"
- automatic image control functions: automatic exposure control (AEC), automatic white balance (AWB), automatic band filter (ABF), automatic 50/60 Hz luminance detection, and automatic black level calibration (ABLC)
- programmable controls for frame rate, AEC/AGC 16-zone size/position/weight control, mirror and flip, cropping, windowing, and panning
- image quality controls: lens correction, defective pixel canceling
- support for output formats: 8-/10-bit raw RGB data support for video or snapshot operations support for LED and flash strobe mode
- support for internal and external frame synchronization for frame exposure mode
- support for horizontal and vertical sub-sampling standard serial SCCB interface
- digital video port (DVP) parallel output interface MIPI interface (two lanes)

- 32 bytes of embedded one -time programmable (OTP) memory
- on- chip phase lock loop (PLL)
- embedded 1.5V regulator for core power
- programmable I/O drive capability, I/O tri-state configurability
- support for black sun cancellation

When you purchase the camera, you will receive a small camera board and cable. You'll want to devise some method of supporting the camera in order to use it. Some camera stands and Raspberry Pi cases are now available. You can also rig up something simple yourself if you wish. I attached mine to a case using a small piece of plastic and double-sided tape, as shown below:



Once the hardware is set up, you can move on to configuring the software.

2.4 I/O control

2.4.1system clock control

The PLL is inside the chip which generates a default 96 MHz clock from 6~27 MHz input clock. An inside programmable clock divider is used to generate different frame rate timing.

2.5 power up sequence

Based on the system power configuration (1.8V or 2.8V for I/O power), using external DVDD or internal DVDD, the power up sequence will differ. If 1.8V is used for I/O power, using the internal DVDD is preferred. If 2.8V is used for I/O power, due to a high voltage drop at the internal DVDD

regulator, there is a potential heat issue. Hence, for a 2.8V power system, OmniVision recommends using an external DVDD source. Due to the higher power down current when using an external DVDD source, OmniVision strongly recommends cutting off all power supplies, including the external DVDD, when the sensor is not in use in the case of 2.8V I/O and external DVDD.

2.5.1 power up with internal DVDD

For powering up with the internal DVDD and SCCB access during the power ON period, the following conditions must occur:

1. if $V_{DD\text{-}IO}$ and $V_{DD\text{-}A}$ are turned ON at the same time, make sure $V_{DD\text{-}IO}$ becomes stable before $V_{DD\text{-}A}$ becomes stable
2. PWDN is active high with an asynchronized design (does not need clock)
3. PWDN must go high during the power up period
4. for PWDN to go low, power must first become stable (AVDD to PWDN $\geq$ 5 ms)
5. RESETB is active low with an asynchronized design
6. state of RESETB does not matter during power up period once DOVDD is up
7. master clock XCLK should provide at least 1 ms before host accesses sensor's SCCB
8. host can access SCCB bus (if shared) during entire period. 20 ms after PWDN goes low or 20 ms after RESETB goes high if reset is inserted after PWDN goes low, host can access sensor's SCCB to initialize sensor

2.6 reset

Two reset modes are available for the OV5647:

- hardware reset
- SCCB software reset

The OV5647 sensor includes a **RESETB** pad that forces a complete hardware reset when it is

pulled low (GND). The OV5647 clears all registers and resets them to their default values when a hardware reset occurs. A reset can also be initiated through the SCCB interface by setting register 0x0103[0] to high.

The whole chip will be reset during power up. Manually applying a hard reset upon power up is recommended even though the on-chip power up reset is included. The hard reset is active low with an asynchronized design. The reset pulse width should be greater than or equal to 1 ms.

2.7 standby and sleep

Two suspend modes are available for the OV5647:

- hardware standby
- SCCB software sleep

To initiate hardware standby mode, the **PWDN** pad must be tied to high. When this occurs, the OV5647 internal device clock is halted and all internal counters are reset and registers are maintained. Executing a software sleep (0x0100[0]) through the SCCB interface suspends internal circuit activity but does not halt the device clock. All register content is maintained in both modes.

Block Level Description

3.1 pixel array structure

The OV5647 sensor has an image array of 2624 columns by 1956 rows (5,132,544 pixels). **figure 3-1** shows a cross-section of the image sensor array.

The color filters are arranged in a Bayer pattern. The primary color BG/GR array is arranged in line-alternating fashion. Of the 5,132,544 pixels, 5,038,848 (2592x1944) are active pixels and can be output. The other pixels are used for black level calibration and interpolation. The center

2592x1944 is suggested to be output from the whole active pixel array. The backend processor can use the boundary pixels for additional processing.

The sensor array design is based on a field integration read-out system with line-by-line transfer and an electronic shutter with a synchronous pixel read-out scheme.

3.3 analog amplifier

When the column sample/hold circuit has sampled one row of pixels, the pixel data will shift out one-by-one into an analog amplifier.

3.4 10-bit A/D converters

The balanced signal is then digitized by the on-chip 10-bit ADC. It can operate at up to 27 MHz and is fully synchronous to the pixel clock. The actual conversion rate is determined by the frame rate.

4 image sensor core digital functions

4.1 mirror and flip

The OV5647 provides mirror and flip read-out modes, which respectively reverse the sensor data read-out order horizontally and vertically (see **figure 4-1**). In flip mode, the OV5647 does not need additional settings because the ISP block will auto-detect whether the pixel is in the red line or blue line and make the necessary adjustments.

Specifications:

- active array size: 2592 x 1944 power supply:
- core: 1.5V ± 5% (with embedded 1.5V regulator) analog: 2.6 ~ 3.0V (2.8V typical)
- I/O: 1.7V ~ 3.0V
- power requirements: active: TBD standby: TBD
- temperature range:
- operating: -30°C to 70°C stable image: 0°C to 50°C
- output formats: 8-/10-bit RGB RAW output
- lens size: 1/4"
- lens chief ray angle: 24 input clock frequency: 6~27 MHz
- S/N ratio: TBD
- dynamic range: TBD
- maximum image transfer rate:
- QSXGA (2592 x 1944): 15 fps 1080p: 30 fps
- 960p: 45 fps 720p: 60 fps
- VGA (640 x 480): 90 fps QVGA (320 x 240): 120 fps
- sensitivity: TBD
- shutter: rolling shutter / global shutter
- maximum exposure interval: 1968 x t_{ROW}
- pixel size: 1.4 µm x 1.4 µm
- well capacity: TBD dark current: TBD
- fixed pattern noise (FPN): TBD
- image area: 3673.6 µm x 2738.4 µm
- die dimensions: 5520 µm x 4700 µm

Software:

Since its inception, the camera is supported in the latest version of Raspbian, the preferred operating system for Raspberry Pi. The instructions in this blog post assume you are running Raspbian. The first step is to get the latest Raspberry Pi firmware, which supports the camera. You can do that from a console by running:

```
sudo apt-get update

sudo apt-get upgrade
```

You then need to enable the camera from the Raspberry Pi configuration program by running:

```
sudo raspi-config
```

Choose "camera" from the program and then select "Enable support for Raspberry Pi camera". You should then reboot when prompted by the raspi-config program. The camera will be enabled on subsequent boots of the Raspberry Pi.

Several applications should now be available for the camera: the rapistill program captures images, raspivid captures videos, and raspiyuv takes uncompressed YUV format images. These are command line program. They accept a number of options, which are documented if you run the commands without options. Several examples are given here[1]. That reference also describes some more sophisticated things you can do, like streaming the video over the network and viewing it on another computer.

The following shell command runs the video capture program with a preview showing all of the built-in camera effects and makes for an interesting demonstration:

```
 for effect in none negative solarise sketch denoise emboss oilpaint hatch gpen pastel watercolour film blur saturation colourswap washedout posterise colourpoint colourbalance cartoon

do

    echo $effect

    raspivid -d -ifx$effect

done
```

If you want to examine the source code for the programs, report bugs or compile them yourself, they are maintained at the project on github.com[2]. You can either cross-compile or build the tools natively on the Raspberry Pi.

User Space V4L2 Driver

The camera drivers are proprietary in the sense that they do not follow any standard APIs. That means that applications have to be written specifically for the Raspberry Pi camera. Under Linux, the standard API for cameras (including web cams) is V4L (Video for Linux), and a number of applications have been written that support any camera with a V4L driver. An independent developer has now written a user space V4L driver for the Raspberry Pi camera, which is available from here[3]. With that driver, you can use generic Linux applications written for cameras. The driver has a few limitations: it is closed sourced, and can be a little slow because it runs as a user program rather than a kernel driver. The program worked reasonably well when I tested it and it is expected to continue to improve.

Official V4L2 Driver

Recognizing that a V4L driver is needed, the Raspberry Pi Foundation reported that they were working with Broadcom to develop an official kernel V4L driver. As a kernel driver, it should be faster than the user space driver. The official driver became available in December 2013. The driver is still quite new and not many people appear to have tried it yet. The latest Raspbian distribution and latest Raspberry Pi boot firmware is required for use. You will also need to build some code yourself. If you want to try it, some brief instructions showing the commands to build it are listed below (these commands should be run from a shell).

```
# Get the latest Raspbian packages

sudo apt-get update

sudo apt-get upgrade

# Get the latest firmware

sudo rpi-update

# Get the source code for the V4L utilities

git clone git://git.linuxtv.org/v4l-utils.git

cd v4l-utils

# Install some packages needed to build it

sudo apt-get install autoconf gettext libtool libjpeg62-dev
```

```
# Configure and build it.
autoreconf -vfi
./configure
make
sudo make install
```

Building the software should take about fifteen minutes. You need to have the camera enabled and sufficient Graphics Processing Unit (GPU) memory configured. Here are some example commands to get started:

```
# Load the module
sudo modprobe bcm2835-v4l2
# Control the viewfinder
v4l2-ctl --overlay=1 # enable viewfinder
v4l2-ctl --overlay=0 # disable viewfinder
# Record a video
v4l2-ctl --set-fmt-video=width=1920,height=1088,pixelformat=4
v4l2-ctl --stream-mmap=3 --stream-count=100 --stream-to=somefile.264
# Capture a JPEG image
v4l2-ctl --set-fmt-video=width=2592,height=1944,pixelformat=3
v4l2-ctl --stream-mmap=3 --stream-count=1 --stream-to=somefile.jpg
# Set the video bitrate
v4l2-ctl --set-ctrl video_bitrate=10000000
# List the supported formats
```

```
v4l2-ctl --list-formats
```

In theory, any camera application written to use the V4L APIs should work with the driver. I encourage you to try it out and report all positive or negative outcomes back to the developers.

Qt Support

TThe Qt Multimedia module under Linux supports cameras that have V4L drivers. Using the user space driver with Qt 5.1, a couple of simple QML camera example applications[4] that I had written earlier were able to run and correctly show the video preview. At higher resolutions, I did see the applications hang. These appeared to be driver related because reinitializing the driver using the v4l-ctl command got it working again. At lower resolutions (e.g. 256x256), it seemed to work reliably.
I have not yet had a chance to try the Qt Multimedia module with the new kernel mode V4L driver.

Other Hardware

As well as the official Raspberry Pi camera module, other cameras supported by Linux, most notably many USB webcams, should work on the Raspberry Pi.

3.6 Servomotor:

Hobby servomotors are a very elegant solution to the problem of adding a motor to your robot. They are mainly used in hobby RC airplanes, so they are very compact, powerful, light and power conservative. Since they have been in production for a long time they are also very cheap. You can buy a standard hobby servomotor for under $13. In a very small package you get a DC motor, gearbox, and feedback control system.

A servomotor is a rotary actuator that allows for precise control of angular position, velocity and acceleration. It consists of a suitable motor coupled to a sensor for position feedback. It also requires a relatively sophisticated controller, often a dedicated module designed specifically for use with servomotors.

Servomotors are not a specific class of motor although the term servomotor is often used to refer to a motor suitable for use in a closed-loop control system.

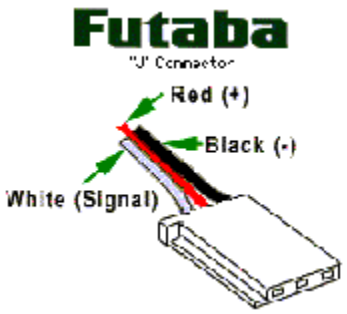
Servomotors are used in applications such as robotics, CNC machinery or automated manufacturing.

The Futaba S-148 standard servo.

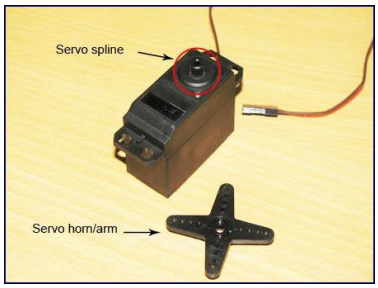
Available from towerhobbies.com or any RC hobby store (Colpar)

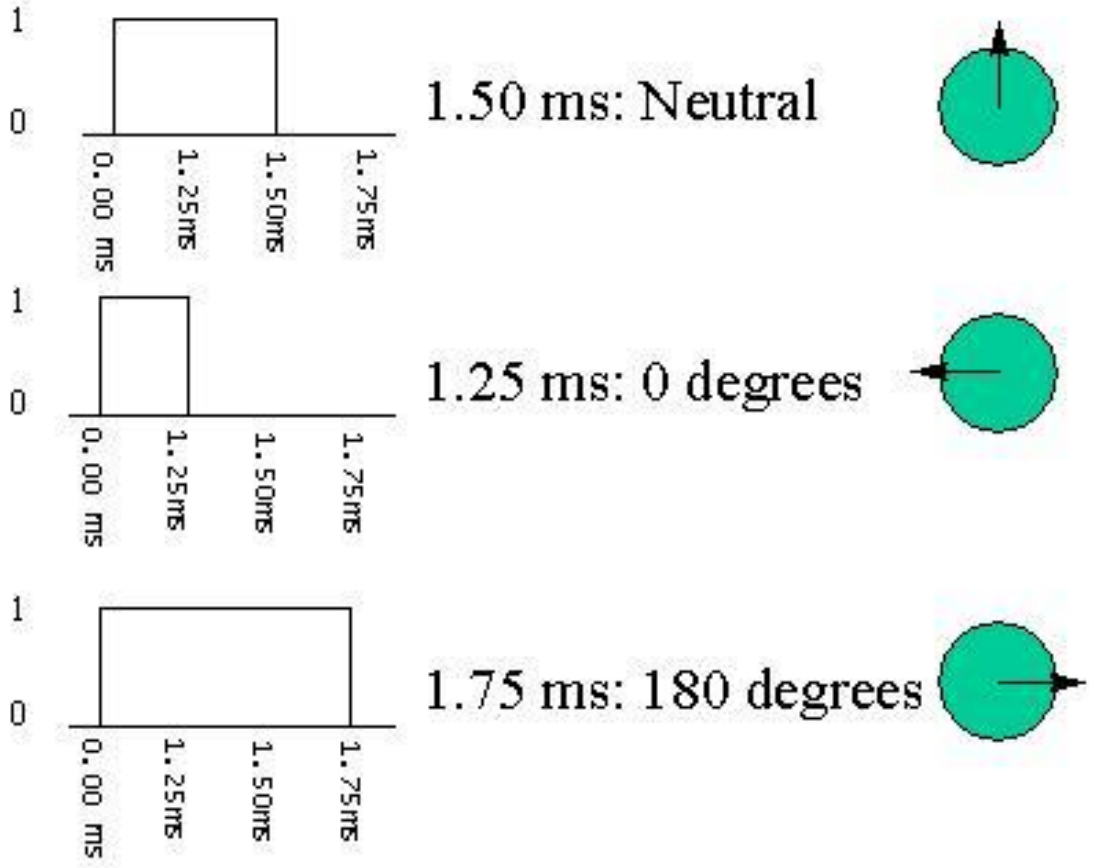
Servomotors are designed to operate control surfaces on hobby RC planes. So they do not rotate continuously. Rather they are designed to rotate through 180 degrees with precise position control. If you want to use them as the main drive motor for a mobile robot you need to modify them so that they will rotate continuously. This is not a difficult thing to do. I will not cover it here but if you want to do it there are many sites on the web that cover this. An excellent one is

They do not simply run on a DC voltage like a standard DC motor. They have 3 wires. Red is power (generally 3V – 12V max), black is ground and then there is another wire, usually white or yellow that is the "input signal wire".



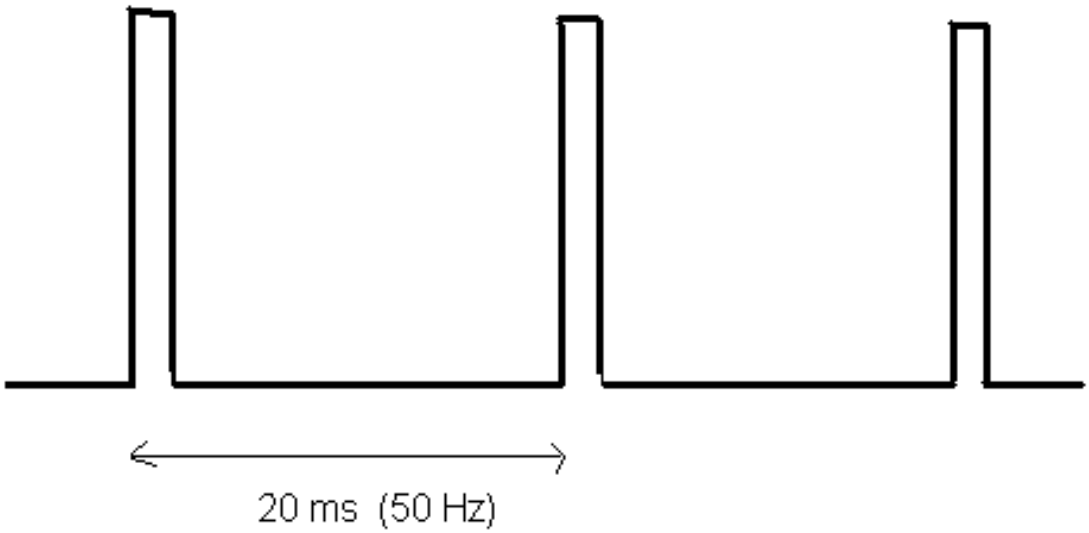
A servomotor is controlled by sending a pulse signal that is HIGH for a brief time, generally 1 – 2 ms. If you just connect a battery to power and ground, nothing will happen. You must have a timer circuit that generates this pulsed signal and by varying the pulse ON time (or the pulse width) the motor will move to a certain position over its range of motion and then stop as long as the input pulse width is the same. Depending on the pulse width, you'll get a different position.





This diagram shows some control signal pulses for a typical servo and the position to which it will rotate in response to the pulse width.

There is another element to the signal that also requires timing accuracy. The frequency of the signal or its rate of refresh. Not only do you have to send the pulse, you have to keep sending them as long as you want the motor to be in that position (or to keep rotating for modified servos). Generally a frequency of 50 Hz is good. This means that you send the Hi pulse 50 times every second.



I mentioned earlier that a servo will only rotate through 180 degrees unless you modify it for continuous rotation (you can also buy them already modified through Acroname and other companies). One interesting thing that comes out this modification is that you get a speed control function out of it, though somewhat coarse.

When you make the modification you replace the circuitry in the motor that tells the motor what position it is in. The mods you make tell the motor that it is always in the center position. So if you feed a 1.75 ms pulse, it rotates to the 180 degree position, checks the feedback which tells it that "hey, you haven't moved yet. You're still in the center position, keep going" so it does, checks and sees that it hasn't moved yet and keeps doing it. Since it thinks that it is in center position and it has to move to its right most position it will move at its fastest rate.

Now suppose you send it a signal that says to rotate to 95 degrees, 5 degrees right of center. The internal control system knows that it is now to move a very short distance. It also knows that if it rotates at its fastest speed that it may overshoot this and have to come back, and overshoot again in the other direction and try again, and so forth. This is called oscillation and is not a good thing. The advantage that you get out of this is that the motor will move slower when you feed a signal that is close to the center position. So you feed it a "go to 95 degree" signal and it will rotate CW at a slow rate. Give it "go to 180 degrees" and it will rotate CW at its fastest rate. And the same for CCW.

Mechanism

As the name suggests, a servomotor is a servomechanism. More specifically, it is a closed-loop servomechanism that uses position feedback to control its motion and final position. The input to its control is some signal, either analogue or digital, representing the position commanded for the output shaft.

The motor is paired with some type of encoder to provide position and speed feedback. In the simplest case, only the position is measured. The measured position of the output is compared to the command position, the external input to the controller. If the output position differs from that required, an error signal is generated which then causes the motor to rotate in either direction, as needed to bring the output shaft to the appropriate position. As the positions approach, the error signal reduces to zero and the motor stops.

The very simplest servomotors use position-only sensing via a potentiometer and bang-bang control of their motor; the motor always rotates at full speed (or is stopped). This type of servomotor is not widely used in industrial motion control, but it forms the basis of the simple and cheap servos used for radio-controlled models.

More sophisticated servomotors measure both the position and also the speed of the output shaft. They may also control the speed of their motor, rather than always running at full speed. Both of these enhancements, usually in combination with aPID control algorithm, allow the servomotor to be brought to its commanded position more quickly and more precisely, with less overshooting.

Servo motor Vs Stepper motor

A servomotor consumes power as it rotates to the commanded position but then the servomotor rests. Stepper motors run warm to the touch because they continue to consume power to lock in and hold the commanded position.

Servomotors are generally used as a high performance alternative to the stepper motor. Stepper motors have some inherent ability to control position, as they have built-in output steps. This often allows them to be used as an open-loop position control, without any feedback encoder, as their drive signal specifies the number of steps of movement to rotate, but for this the controller needs to 'know' the position of the stepper motor on power up. Therefore, on first power up, the controller will

have to activate the stepper motor and turn it to a known position, e.g. until it activates an end limit switch. This can be observed when switching on an inkjet printer; the controller will move the ink jet carrier to the extreme left and right to establish the end positions. A servomotor will immediately turn to whatever angle the controller instructs it to, regardless of the initial position at power up.

The lack of feedback of a stepper motor limits its performance, as the stepper motor can only drive a load that is well within its capacity, otherwise missed steps under load may lead to positioning errors and the system may have to be restarted or recalibrated. The encoder and controller of a servomotor are an additional cost, but they optimise the performance of the overall system (for all of speed, power and accuracy) relative to the capacity of the basic motor. With larger systems, where a powerful motor represents an increasing proportion of the system cost, servomotors have the advantage.

There has been increasing popularity in closed loop stepper motors in recent years. They act like servomotors but have some differences in their software control to get smooth motion. The top 3 manufacturers of closed loop stepper motor systems employ magnetic encoders as their feedback device of choice due to low cost, and resistance to vibration. The main benefit of a closed loop stepper motor is the cost to performance ratio. There is also no need to tune the PID controller on a closed loop stepper system saving time.

Many applications, such as laser cutting machines, may be offered in two ranges, the low-priced range using stepper motors and the high-performance range using servomotors.

Encoders

The first servomotors were developed with synchros as their encoders. Much work was done with these systems in the development of radar and anti-aircraft artillery during World War II.

Simple servomotors may use resistive potentiometers as their position encoder. These are only used at the very simplest and cheapest level, and are in close competition with stepper motors. They suffer from wear and electrical noise in the potentiometer track. Although it would be possible to electrically differentiate their position signal to obtain a speed signal, PID controllers that can make use of such a speed signal generally warrant a more precise encoder.

Modern servomotors use rotary encoders, either absolute or incremental. Absolute encoders can determine their position at power-on, but are more complicated and expensive. Incremental

encoders are simpler, cheaper and work at faster speeds. Incremental systems, like stepper motors, often combine their inherent ability to measure intervals of rotation with a simple zero-position sensor to set their position at start-up.

Instead of servomotors, sometimes a motor with a separate, external linear encoder is used. These motor + linear encoder systems avoid inaccuracies in the drive train between the motor and linear carriage, but their design is made more complicated as they are no longer a pre-packaged factory-made system.

Control

Most modern servomotors are designed and supplied around a dedicated controller module from the same manufacturer. Controllers may also be developed around microcontrollers in order to reduce cost for large volume applications.

Controlling a Servo Motor

A servo motor is controlled by controlling its position using Pulse Width Modulation Technique. The width of the pulse applied to the motor is varied and send for a fixed amount of time.

The pulse width determines the angular position of the servo motor. For example a pulse width of 1 ms causes a angular position of 0 degrees, whereas a pulse width of 2 ms causes a angular width of 180 degrees.

Advantages:

- If a heavy load is placed on the motor, the driver will increase the current to the motor coil as it attempts to rotate the motor. Basically, there is no out-of-step condition.
- High-speed operation is possible.
- *High performance*
- *Accurate positioning (because of feedback)*
- Comparatively high power output than the motor weight and size.

- Encoder sets resolution and accuracy.
- Highly efficient. It can reach up to 90% of light loads.
- Exuberantly high torque to inertia ratio which leads to quick acceleration of loads.
- Reserves power for emergencies and can provide 2-3 times constant power for short time.

Disadvantages:

- Since the servomotor tries to rotate according to the command pulses, but lags behind, it is not suitable for precision control of rotation.
- Higher cost.
- When stopped, the motor's rotor continues to move back and forth one pulse, so that it is not suitable if you need to prevent vibration
- 7 Applications of Servo Motors
- Servomotors are used in applications requiring rapid variations in speed without the motor getting overheated.
- In Industries they are used in machine tools, packaging, factory automation, material handling, printing converting, assembly lines, and many other demanding applications robotics, CNC machinery or automated manufacturing.
- They are also used in radio controlled airplanes to control the positioning and movement of elevators.
- They are used in robots because of their smooth switching on and off and accurate positioning.
- They are also used by aerospace industry to maintain hydraulic fluid in their hydraulic systems.
- They are used in many radio controlled toys.
- They are used in electronic devices such as DVDs or Blue ray Disc players to extend or replay the disc trays.
- They are also being used in automobiles to maintain the speed of vehicles.

Applications of Servo Motor

Each motor has three inputs: VCC, ground and a periodic square wave signal. The pulse width of the square wave determines the speed and direction of the servo motors. In our case, we just need to

change the direction to allow the device to move forward, backward and turn left and right. If the pulse width is under a certain time frame, the motor will drive in a clockwise direction. If the pulse width exceeds that time frame, the motor will drive in a counterclockwise direction. The middle time frame can be adjusted through a built-in potentiometer inside the motor.

3.7 Bluetooth module:

INTRODUCTION

HC-05 module is an easy-to-use Bluetooth SPP (Serial Port Protocol) module, designed for transparent wireless serial connection setup. The HC-05 Bluetooth Module can be used in a Master or Slave configuration, making it a great solution for wireless communication. This serial port Bluetooth module is fully qualified Bluetooth V2.0+EDR (Enhanced Data Rate) 3Mbps Modulation with complete 2.4GHz radio transceiver and baseband. It uses CSR Bluecore 04-External single chip Bluetooth system with CMOS technology and with AFH (Adaptive Frequency Hopping Feature).

Bluetooth module HC-05

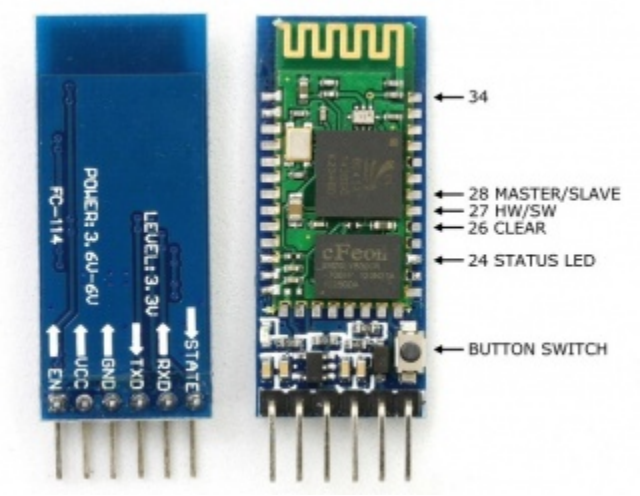
The Bluetooth module HC-05 is a MASTER/SLAVE module. By default the factory setting is SLAVE. The Role of the module (Master or Slave) can be configured only by AT COMMANDS. The slave modules cannot initiate a connection to another Bluetooth device, but can accept connections. Master module can initiate a connection to other devices. The user can use it simply for a serial port replacement to establish connection between MCU and GPS, PC to your embedded project, etc. Just go through the datasheet for more details.

Hardware Features

- Typical -80dBm sensitivity.
- Up to +4dBm RF transmit power.
- 3.3 to 5 V I/O.
- PIO(Programmable Input/Output) control.
- UART interface with programmable baud rate.
- With integrated antenna.
- With edge connector.

Software Features

- Slave default Baud rate: 9600, Data bits:8, Stop bit:1,Parity:No parity.
- Auto-connect to the last device on power as default.
- Permit pairing device to connect as default.
- Auto-pairing PINCODE:"1234" as default.



PIN DESCRIPTION:

The HC-05 Bluetooth Module has 6pins. They are as follows:

ENABLE:

When enable is pulled **LOW**, the module is disabled which means the module will **not turn on** and it **fails to communicate**.When enable is **left open or connected to 3.3V**, the module is enabled i.e the module **remains on** and **communication also takes place**.

Vcc:

Supply Voltage 3.3V to 5V

GND:

Ground pin

TXD & RXD:

These two pins acts as an UART interface for communication

STATE:

It acts as a status indicator. When the module is **not connected to / paired** with any other Bluetooth device, signal goes **Low**. At this **low state**, the **led flashes continuously** which denotes that the module is **not paired** with other device. When this module is **connected to/paired** with any other Bluetooth device, the signal goes **High**. At this **high state**, the **led blinks with a constant delay** say for example 2s delay which indicates that the module is **paired**.

BUTTON SWITCH:

This is used to switch the module into AT command mode. To enable AT command mode, press the button switch for a second. With the help of AT commands, the user can change the parameters of this module but only when the module is not paired with any other BT device. If the module is connected to any other Bluetooth device, it starts to communicate with that device and fails to work in AT command mode.

3.8 DC Motors with driver

D.C. Motor:

A dc motor uses electrical energy to produce mechanical energy, very typically through the interaction of magnetic fields and current-carrying conductors. The reverse process, producing electrical energy from mechanical energy, is accomplished by an alternator, generator or dynamo. Many types of electric motors can be run as generators, and vice versa.The input of a DC motor is current/voltage and its output is torque (speed).

Fig 3.19: DC Motor

The DC motor has two basic parts: the rotating part that is called the armature and the stationary part that includes coils of wire called the field coils. The stationary part is also called the stator. Figure shows a picture of a typical DC motor, Figure shows a picture of a DC armature, and Fig shows a picture of a typical stator. From the picture you can see the armature is made of coils of wire wrapped around the core, and the core has an extended shaft that rotates on bearings. You should also notice that the ends of each coil of wire on the armature are terminated at one end of the armature. The termination points are called the commutator, and this is where the brushes make electrical contact to bring electrical current from the stationary part to the rotating part of the machine.

Operation:

The DC motor you will find in modem industrial applications operates very similarly to the simple DC motor described earlier in this chapter. Figure 12-9 shows an electrical diagram of a simple DC motor. Notice that the DC voltage is applied directly to the field winding and the brushes. The armature and the field are both shown as a coil of wire. In later diagrams, a field resistor will be added in series with the field to control the motor speed.When voltage is applied to the motor, current begins to flow through the field coil from the negative terminal to the positive terminal. This sets up a strong magnetic field in the field winding. Current also begins to flow through the brushes into a commutator segment and then through an armature coil. The current continues to flow through the coil back to the brush that is attached to other end of the coil and returns to the DC power source. The current flowing in the armature coil sets up a strong magnetic field in the armature.

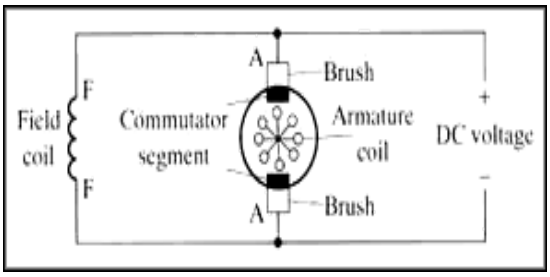


Fig 3.20: Simple electrical diagram of DC motor

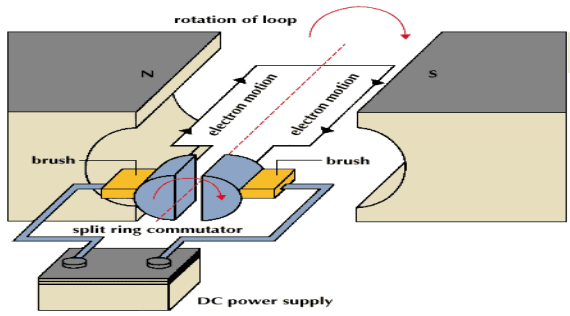


Fig 3.21: Operation of a DC Motor

The magnetic field in the armature and field coil causes the armature to begin to rotate. This occurs by the unlike magnetic poles attracting each other and the like magnetic poles repelling each other. As the armature begins to rotate, the commutator segments will also begin to move under the brushes. As an individual commutator segment moves under the brush connected to positive voltage, it will become positive, and when it moves under a brush connected to negative voltage it will become negative. In this way, the commutator segments continually change polarity from positive to negative. Since the commutator segments are connected to the ends of the wires that make up the field winding in the armature, it causes the magnetic field in the armature to change polarity continually from north pole to south pole. The commutator segments and brushes are aligned in such a way that the switch in polarity of the armature coincides with the location of the armature's magnetic field and the field winding's magnetic field. The switching action is timed so that the armature will not lock up magnetically with the field. Instead the magnetic fields tend to build on each other and provide additional torque to keep the motor shaft rotating.

When the voltage is de-energized to the motor, the magnetic fields in the armature and the field winding will quickly diminish and the armature shaft's speed will begin to drop to zero. If voltage is applied to the motor again, the magnetic fields will strengthen and the armature will begin to rotate again.

Types of DC motors:

1. DC Shunt Motor,

2. DC Series Motor,

3. DC Long Shunt Motor (Compound)

4. DC Short Shunt Motor (Compound)

The rotational energy that you get from any motor is usually the battle between two magnetic fields chasing each other. The DC motor has magnetic poles and an armature, to which DC electricity is fed, The Magnetic Poles are electromagnets, and when they are energized, they produce a strong magnetic field around them, and the armature which is given power with a commutator, constantly repels the poles, and therefore rotates.

1. The DC Shunt Motor:

In a 2 pole DC Motor, the armature will have two separate sets of windings, connected to a commutator at the end of the shaft that are in constant touch with carbon brushes. The brushes are static, and the commutator rotate and as the portions of the commutator touching the respective positive or negative polarity brush will energize the respective part of the armature with the respective polarity. It is usually arranged in such a way that the armature and the poles are always repelling.

The general idea of a DC Motor is, the stronger the Field Current, the stronger the magnetic field, and faster the rotation of the armature. When the armature revolves between the poles, the magnetic field of the poles induce power in the armature conductors, and some electricity is generated in the armature, which is called back emf, and it acts as a resistance for the armature. Generally an armature has resistance of less than 1 Ohm, and powering it with heavy voltages of Direct Current could result in immediate short circuits. This back emf helps us there.When an armature is loaded on a DC Shunt Motor, the speed naturally reduces, and therefore the back emf reduces, which allows more armatures current to flow. This results in more armature field, and therefore it results in torque.

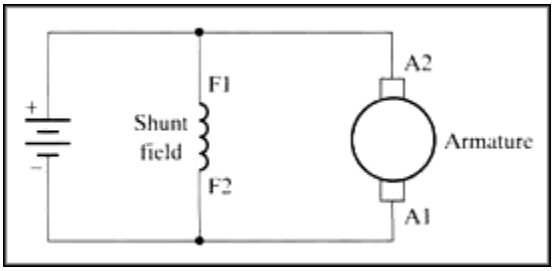


Fig: Diagram of DC shunt motor

When a DC Shunt Motor is overloaded, if the armature becomes too slow, the reduction of the back emf could cause the motor to burn due to heavy current flow thru the armature.

The poles and armature are excited separately, and parallel, therefore it is called a Shunt Motor.

2. The DC Series Motor:

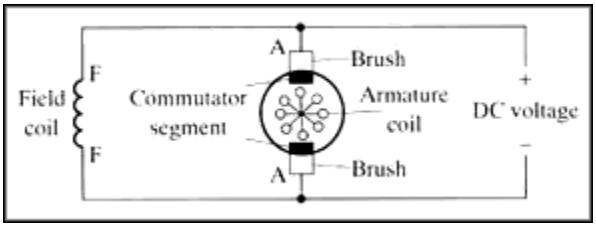


Fig: Diagram of DC series motor

A DC Series Motor has its field coil in series with the armature. Therefore any amount of power drawn by the armature will be passed thru the field. As a result you cannot start a Series DC Motor without any load attached to it. It will either run uncontrollably in full speed, or it will stop.

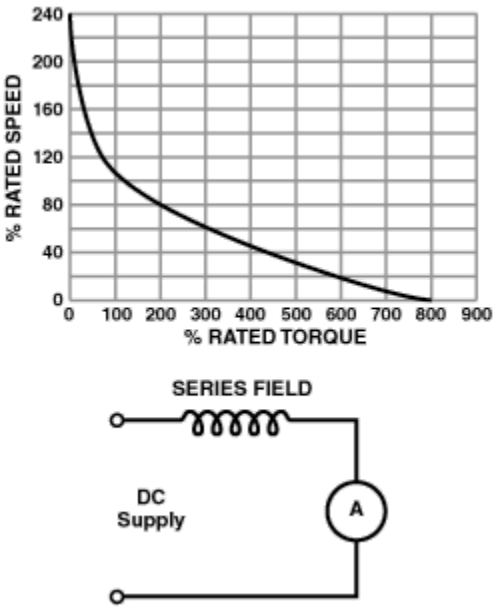


Fig: Diagram of DC series motor graph representation

When the load is increased then its efficiency increases with respect to the load applied. So these are on Electric Trains and elevators.

Specifications

- DC supply: 4 to 12V
- RPM: 300 at 12V
- Total length: 46mm

- Motor diameter: 36mm
- Motor length: 25mm
- Brush type: Precious metal
- Gear head diameter: 37mm
- Gear head length: 21mm
- Output shaft: Centred
- Shaft diameter: 6mm
- Shaft length: 22mm
- Gear assembly: Spur
- Motor weight: 105gms

We generally use 300RPM Centre Shaft Economy Series DC Motor which is high quality low cost DC geared motor. It has steel gears and pinions to ensure longer life and better wear and tear properties. The gears are fixed on hardened steel spindles polished to a mirror finish. The output shaft rotates in a plastic bushing. The whole assembly is covered with a plastic ring. Gearbox is sealed and lubricated with lithium grease and require no maintenance. The motor is screwed to the gear box from inside.

Although motor gives 300 RPM at 12V but motor runs smoothly from 4V to 12V and gives wide range of RPM, and torque. Tables below gives fairly good idea of the motor's performance in terms of RPM and no load current as a function of voltage and stall torque, stall current as a function of voltage.

3. DC Compound Motor:

A compound of Series and Shunt excitation for the fields is done in a Compound DC Motor. This gives the best of both series and shunt motors. Better torque as in a series motor, while the possibility to start the motor with no load.

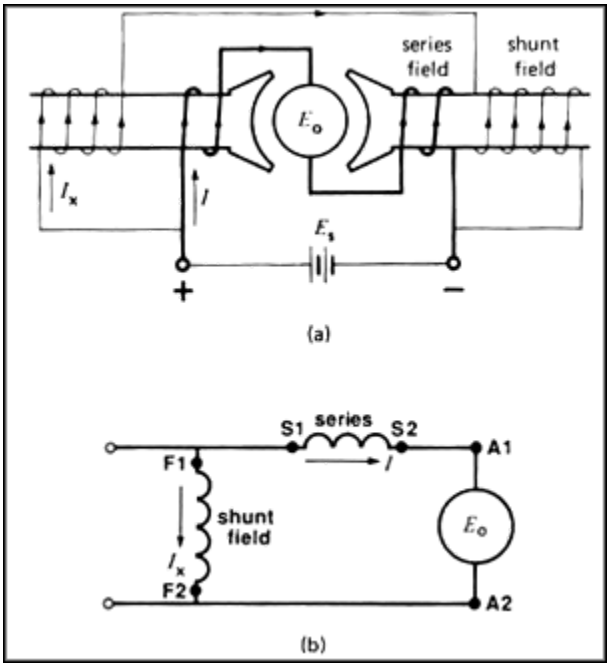


Fig: Diagram of DC compound motor

Above is the diagram of a long shunt motor, while in a short shunt, the shunt coil will be connected after the serial coil.

A Compound motor can be run as a shunt motor without connecting the serial coil at all but not vice versa.

DC Motor Driver:

The L293 and L293D are quadruple high-current half-H drivers. The L293 is designed to provide bidirectional drive currents of up to 1 A at voltages from 4.5 V to 36 V. The L293D is designed to provide bidirectional drive currents of up to 600-mA at voltages from 4.5 V to 36 V. Both devices are designed to drive inductive loads such as relays, solenoids, dc and bipolar stepping motors, as well as other high-current/high-voltage loads in positive-supply applications.

All inputs are TTL compatible. Each output is a complete totem-pole drive circuit, with a Darlington transistor sink and a pseudo-Darlington source. Drivers are enabled in pairs, with drivers 1 and 2 enabled by 1,2EN and drivers 3 and 4 enabled by 3,4EN.When an enable input is high, the associated drivers are enabled and their outputs are active and in phase with their inputs.

When the enable input is low, those drivers are disabled and their outputs are off and in the high-impedance state. With the proper data inputs, each pair of drivers forms a full-H (or bridge) reversible drive suitable for solenoid or motor applications. On the L293, external high-speed output clamp diodes should be used for inductive transient suppression. A VCC1 terminal, separate from VCC2, is provided for the logic inputs to minimize device power dissipation. The L293and L293D are characterized for operation from 0°C to 70°C.

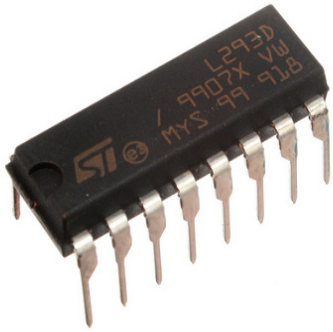


Fig 3.22: L293D IC

Pin Diagram of L293D motor driver:

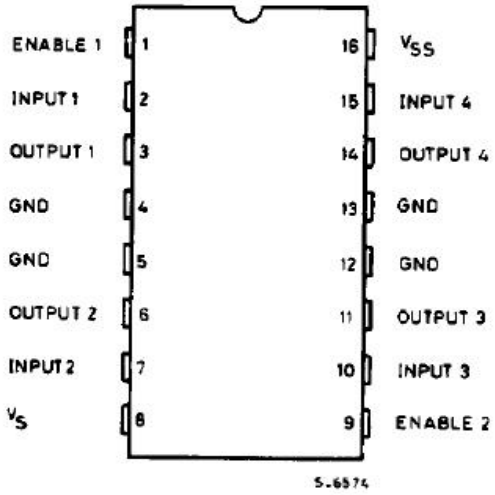


Fig 3.23: L293D pin diagram

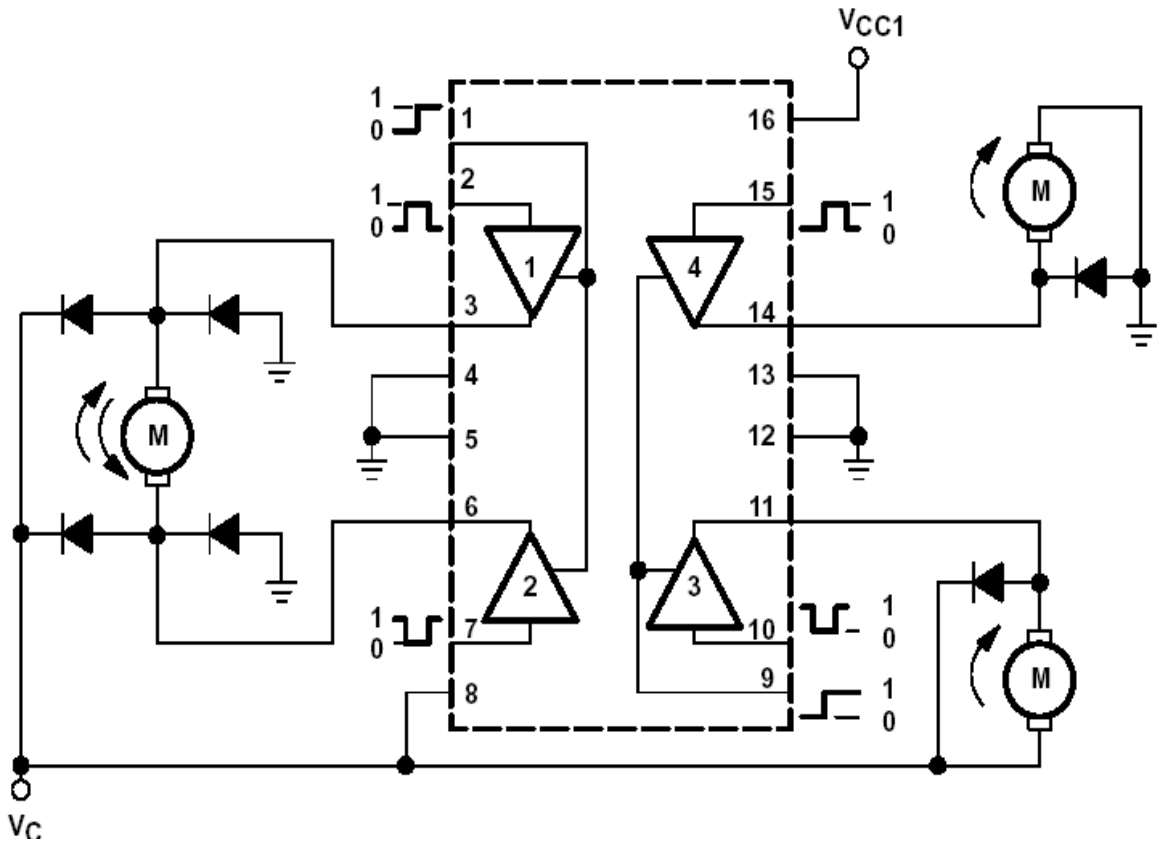


Fig 3.24: Internal structure of L293D.

Features of L293D:

- 600mA Output current capability per channel
- 1.2APeak output current (non repetitive) per channel
- Enable facility
- Over temperature protection
- Logical “0”input voltage up to 1.5 v
- High noise immunity
- Internal clamp diodes

Applications of DC Motors:

1. Electric Train: A kind of DC motor called the DC Series Motor is used in Electric Trains. The DC Series Motors have the property to deliver more power when they are loaded more. So the more the people get on a train, the more powerful the train becomes.

2. Elevators: The best bidirectional motors are DC motors. They are used in elevators. CompoundDC Motors are used for this application.

3. PC Fans, CD ROM Drives, and Hard Drives: All these things need motors, very miniature motors, with great precision. AC motors can never imagine any application in these places.

4. Starter Motors in Automobiles: An automobile battery supplies DC, so a DC motor is best suited here. Also, you cannot start an engine with a small sized AC motor,

5. Electrical Machines Lab in Colleges.

CHAPTER 4: SOFTWARE DESCRIPTION

This project is implemented using following software's and Linux Operating System:

- Express PCB – for designing circuit
- Linux OS
- Proteus 7 (Embedded C) – for simulation part

4.1 Express PCB:

Breadboards are great for prototyping equipment as it allows great flexibility to modify a design when needed; however the final product of a project, ideally should have a neat PCB, few

cables, and survive a shake test. Not only is a proper PCB neater but it is also more durable as there are no cables which can yank loose.

ExpressPCB is a software tool to design PCBs specifically for manufacture by the company ExpressPCB (no other PCB maker accepts ExpressPCB files). It is very easy to use, but it does have several limitations.

It can be likened to more of a toy then a professional CAD program.

It has a poor part library (which we can work around)

It cannot import or export files in different formats

It cannot be used to make prepare boards for DIY production

ExpressPCB has been used to design many PCBs (some layered and with surface-mount parts. Print out PCB patterns and use the toner transfer method with an Etch Resistant Pen to make boards. However, ExpressPCB does not have a nice print layout. Here is the procedure to design in ExpressPCB and clean up the patterns so they print nicely.

4.1.1 Preparing Express PCB for First Use:

ExpressPCB comes with a less then exciting list of parts. So before any project is started head over to Audio logic and grab the additional parts by morsel, ppl, and tangent, and extract them into your ExpressPCB directory. At this point start the program and get ready to setup the workspace to suit your style.

Click View -> Options. In this menu, setup the units for “mm” or “in” depending on how you think, and click “see through the top copper layer” at the bottom. The standard color scheme of red and green is generally used but it is not as pleasing as red and blue.

4.1.2 The Interface:

When a project is first started you will be greeted with a yellow outline. This yellow outline is the dimension of the PCB. Typically after positioning of parts and traces, move them to their final position and then crop the PCB to the correct size. However, in designing a board with a certain size constraint, crop the PCB to the correct size before starting.

Fig: 4.1 show the toolbar in which the each button has the following functions:

Fig 4.1: Tool bar necessary for the interface

- The select tool: It is fairly obvious what this does. It allows you to move and manipulate parts. When this tool is selected the top toolbar will show buttons to move traces to the top / bottom copper layer, and rotate buttons.
- The zoom to selection tool: does just that.
- The place pad: button allows you to place small soldier pads which are useful for board connections or if a part is not in the part library but the part dimensions are available. When this tool is selected the top toolbar will give you a large selection of round holes, square holes and surface mount pads.
- The place component: tool allows you to select a component from the top toolbar and then by clicking in the workspace places that component in the orientation chosen using the buttons next to the component list. The components can always be rotated afterwards with the select tool if the orientation is wrong.
- The place trace: tool allows you to place a solid trace on the board of varying thicknesses. The top toolbar allows you to select the top or bottom layer to place the trace on.
- The Insert Corner in trace: button does exactly what it says. When this tool is selected, clicking on a trace will insert a corner which can be moved to route around components and other traces.
- The remove a trace button is not very important since the delete key will achieve the same result.

4.1.3 Design Considerations:

Before starting a project there are several ways to design a PCB and one must be chosen to suit the project's needs.

Single sided, or double sided?

When making a PCB you have the option of making a single sided board, or a double sided board. Single sided boards are cheaper to produce and easier to etch, but much harder to design for large projects. If a lot of parts are being used in a small space it may be difficult to make a single sided board without jumper over traces with a cable. While there's technically nothing wrong with this, it should be avoided if the signal travelling over the traces is sensitive (e.g. audio signals).

A double sided board is more expensive to produce professionally, more difficult to etch on a DIY board, but makes the layout of components a lot smaller and easier. It should be noted that if a trace is running on the top layer, check with the components to make sure you can get to its pins with a soldering iron.

Large capacitors, relays, and similar parts which don't have axial leads can NOT have traces on top unless boards are plated professionally.

When using a double sided board you must consider which traces should be on what side of the board. Generally, put power traces on the top of the board, jumping only to the bottom if a part cannot be soldiered onto the top plane (like a relay), and vice- versa.

Some projects like power supplies or amps can benefit from having a solid plane to use for ground. In power supplies this can reduce noise, and in amps it minimizes the distance between parts and their ground connections, and keeps the ground signal as simple as possible. However, care must be taken with stubborn chips such as the TPA6120 amplifier from TI. The TPA6120 datasheet specifies not to run a ground plane under the pins or signal traces of this chip as the capacitance generated could effect performance negatively.

4.2 LINUXoperating System

Introduction to Operating Systems

It is a common occurrence to find users who are not even aware of what operating system they are running. On occasion, you may also find an administrator who knows the name of the operating

system, but nothing about the inner workings of it. In many cases, they have no time as they are often clerical workers or other personnel who were reluctantly appointed to be the system administrator.

Being able to run or work on a Linux system does not mean you must understand the intricate details of how it functions internally. However, there are some operating system concepts that will help you to interact better with the system. They will also serve as the foundation for many of the issues we're going to cover in this section.

In this chapter we are going to go through the basic composition of an operating system. First, we'll talk about what an operating system is and why it is important. We are also going to address how the different components work independently and together.

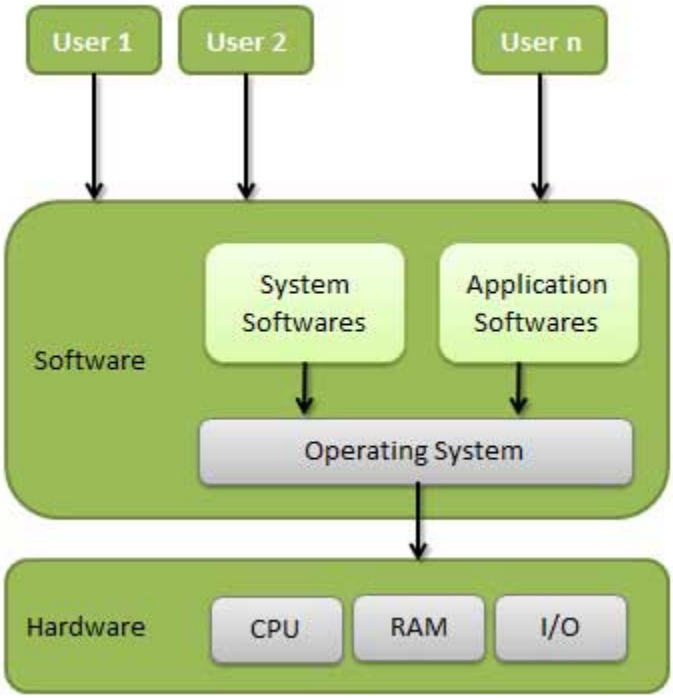
My goal is not to make you an expert on operating system concepts. Instead, I want to provide you with a starting point from which we can go on to other topics. If you want to go into more detail about operating systems, I would suggest*Modern Operating Systems* by Andrew Tanenbaum, published by Prentice Hall, and *Operating System Concepts* by Silberschatz, Peterson, and Galvin, published by Addison-Wesley. Another is *Inside Linux* by Randolph Bentson, which gives you a quick introduction to operating system concepts from the perspective of Linux.

An operating System (OS) is an intermediary between users and computer hardware. It provides users an environment in which a user can execute programs conveniently and efficiently.

In technical terms, It is a software which manages hardware. An operating System controls the allocation of resources and services such as memory, processors, devices and information.

Definition

An operating system is a program that acts as an interface between the user and the computer hardware and controls the execution of all kinds of programs.



Following are some of important functions of an operating System.

- Memory Management
- Processor Management
- Device Management
- File Management
- Security
- Control over system performance
- Job accounting
- Error detecting aids
- Coordination between other software and users

Memory Management

Memory management refers to management of Primary Memory or Main Memory. Main memory is a large array of words or bytes where each word or byte has its own address.

Main memory provides a fast storage that can be access directly by the CPU. So for a program to be executed, it must in the main memory. Operating System does the following activities for memory management.

- Keeps tracks of primary memory i.e. what part of it are in use by whom, what part are not in use.
- In multiprogramming, OS decides which process will get memory when and how much.
- Allocates the memory when the process requests it to do so.
- De-allocates the memory when the process no longer needs it or has been terminated.

Processor Management

In multiprogramming environment, OS decides which process gets the processor when and how much time. This function is called process scheduling. Operating System does the following activities for processor management.

- Keeps tracks of processor and status of process. Program responsible for this task is known as traffic controller.
- Allocates the processor(CPU) to a process.
- De-allocates processor when processor is no longer required.

Device Management

OS manages device communication via their respective drivers. Operating System does the following activities for device management.

- Keeps tracks of all devices. Program responsible for this task is known as the I/O controller.
- Decides which process gets the device when and for how much time.
- Allocates the device in the efficient way.
- De-allocates devices.

File Management

A file system is normally organized into directories for easy navigation and usage. These directories may contain files and other directions. Operating System does the following activities for file management.

- Keeps track of information, location, uses, status etc. The collective facilities are often known as file system.
- Decides who gets the resources.
- Allocates the resources.
- De-allocates the resources.

Following are some of the important activities that Operating System does.

- **Security** -- By means of password and similar other techniques, preventing unauthorized access to programs and data.
- **Control over system performance** -- Recording delays between request for a service and response from the system.
- **Job accounting** -- Keeping track of time and resources used by various jobs and users.
- **Error detecting aids** -- Production of dumps, traces, error messages and other debugging and error detecting aids.
- **Coordination between other software's and users** -- Coordination and assignment of compilers, interpreters, assemblers and other software to the various users of the computer systems.

Operating systems are there from the very first computer generation. Operating systems keep evolving over the period of time. Following are few of the important types of operating system which are most commonly used.

Batch operating system

The users of batch operating system do not interact with the computer directly. Each user prepares his job on an off-line device like punch cards and submits it to the computer operator. To speed up processing, jobs with similar needs are batched together and run as a group. Thus, the programmers left their programs with the operator. The operator then sorts programs into batches with similar requirements.

The problems with Batch Systems are following.

- Lack of interaction between the user and job.
- CPU is often idle, because the speeds of the mechanical I/O devices are slower than CPU.
- Difficult to provide the desired priority.

Time-sharing operating systems

Time sharing is a technique which enables many people, located at various terminals, to use a particular computer system at the same time. Time-sharing or multitasking is a logical extension of multiprogramming. Processor's time which is shared among multiple users simultaneously is termed as time-sharing. The main difference between Multiprogrammed Batch Systems and Time-Sharing Systems is that in case of Multiprogrammed batch systems, objective is to maximize processor use, whereas in Time-Sharing Systems objective is to minimize response time.

Multiple jobs are executed by the CPU by switching between them, but the switches occur so frequently. Thus, the user can receives an immediate response. For example, in a transaction processing, processor execute each user program in a short burst or quantum of computation. That is if n users are present, each user can get time quantum. When the user submits the command, the response time is in few seconds at most.

Operating system uses CPU scheduling and multiprogramming to provide each user with a small portion of a time. Computer systems that were designed primarily as batch systems have been modified to time-sharing systems.

Advantages of Timesharing operating systems are following

- Provide advantage of quick response.
- Avoids duplication of software.
- Reduces CPU idle time.

Disadvantages of Timesharing operating systems are following.

- Problem of reliability.

- Question of security and integrity of user programs and data.
- Problem of data communication.

Distributed operating System

Distributed systems use multiple central processors to serve multiple real time application and multiple users. Data processing jobs are distributed among the processors accordingly to which one can perform each job most efficiently.

The processors communicate with one another through various communication lines (such as high-speed buses or telephone lines). These are referred as loosely coupled systems or distributed systems. Processors in a distributed system may vary in size and function. These processors are referred as sites, nodes, computers and so on.

The advantages of distributed systems are following.

- With resource sharing facility user at one site may be able to use the resources available at another.
- Speedup the exchange of data with one another via electronic mail.
- If one site fails in a distributed system, the remaining sites can potentially continue operating.
- Better service to the customers.
- Reduction of the load on the host computer.
- Reduction of delays in data processing.

Network operating System

Network Operating System runs on a server and and provides server the capability to manage data, users, groups, security, applications, and other networking functions. The primary purpose of the network operating system is to allow shared file and printer access among multiple computers in a network, typically a local area network (LAN), a private network or to other networks. Examples of network operating systems are Microsoft Windows Server 2003, Microsoft Windows Server 2008, UNIX, Linux, Mac OS X, Novell NetWare, and BSD.

The advantages of network operating systems are following.

- Centralized servers are highly stable.
- Security is server managed.
- Upgrades to new technologies and hardwares can be easily integrated into the system.
- Remote access to servers is possible from different locations and types of systems.

The disadvantages of network operating systems are following.

- High cost of buying and running a server.
- Dependency on a central location for most operations.
- Regular maintenance and updates are required.

Real Time operating System

Real time system is defines as a data processing system in which the time interval required to process and respond to inputs is so small that it controls the environment. Real time processing is always on line whereas on line system need not be real time. The time taken by the system to respond to an input and display of required updated information is termed as response time. So in this method response time is very less as compared to the online processing.

Real-time systems are used when there are rigid time requirements on the operation of a processor or the flow of data and real-time systems can be used as a control device in a dedicated application. Real-time operating system has well-defined, fixed time constraints otherwise system will fail.For example Scientific experiments, medical imaging systems, industrial control systems, weapon systems, robots, and home-appliance controllers, Air traffic control system etc.

There are two types of real-time operating systems.

HARD REAL-TIME SYSTEMS

Hard real-time systems guarantee that critical tasks complete on time. In hard real-time systems secondary storage is limited or missing with data stored in ROM. In these systems virtual memory is almost never found.

SOFT REAL-TIME SYSTEMS

Soft real time systems are less restrictive. Critical real-time task gets priority over other tasks and retains the priority until it completes. Soft real-time systems have limited utility than hard real-time systems.For example, Multimedia, virtual reality, Advanced Scientific Projects like undersea exploration and planetary rovers etc

An Operating System provides services to both the users and to the programs.

- It provides programs, an environment to execute.
- It provides users, services to execute the programs in a convenient manner.

Following are few common services provided by operating systems.

- Program execution
- I/O operations
- File System manipulation
- Communication
- Error Detection
- Resource Allocation
- Protection

Program execution

Operating system handles many kinds of activities from user programs to system programs like printer spooler, name servers, file server etc. Each of these activities is encapsulated as a process.

A process includes the complete execution context (code to execute, data to manipulate, registers, OS resources in use). Following are the major activities of an operating system with respect to program management.

- Loads a program into memory.
- Executes the program.

- Handles program's execution.
- Provides a mechanism for process synchronization.
- Provides a mechanism for process communication.
- Provides a mechanism for deadlock handling.

I/O Operation

I/O subsystem comprised of I/O devices and their corresponding driver software. Drivers hides the peculiarities of specific hardware devices from the user as the device driver knows the peculiarities of the specific device.

Operating System manages the communication between user and device drivers. Following are the major activities of an operating system with respect to I/O Operation.

- I/O operation means read or write operation with any file or any specific I/O device.
- Program may require any I/O device while running.
- Operating system provides the access to the required I/O device when required.

File system manipulation

A file represents a collection of related information. Computer can store files on the disk (secondary storage), for long term storage purpose. Few examples of storage media are magnetic tape, magnetic disk and optical disk drives like CD, DVD. Each of these media has its own properties like speed, capacity, data transfer rate and data access methods.

A file system is normally organized into directories for easy navigation and usage. These directories may contain files and other directions. Following are the major activities of an operating system with respect to file management.

- Program needs to read a file or write a file.
- The operating system gives the permission to the program for operation on file.
- Permission varies from read-only, read-write, denied and so on.

- Operating System provides an interface to the user to create/delete files.
- Operating System provides an interface to the user to create/delete directories.
- Operating System provides an interface to create the backup of file system.

Communication

In case of distributed systems which are a collection of processors that do not share memory, peripheral devices, or a clock, operating system manages communications between processes. Multiple processes with one another through communication lines in the network

OS handles routing and connection strategies, and the problems of contention and security. Following are the major activities of an operating system with respect to communication.

- Two processes often require data to be transferred between them.
- The both processes can be on the one computer or on different computer but are connected through computer network.
- Communication may be implemented by two methods either by Shared Memory or by Message Passing.

Error handling

Error can occur anytime and anywhere. Error may occur in CPU, in I/O devices or in the memory hardware. Following are the major activities of an operating system with respect to error handling.

- OS constantly remains aware of possible errors.
- OS takes the appropriate action to ensure correct and consistent computing.

Resource Management

In case of multi-user or multi-tasking environment, resources such as main memory, CPU cycles and files storage are to be allocated to each user or job. Following are the major activities of an operating system with respect to resource management.

- OS manages all kind of resources using schedulers.

- CPU scheduling algorithms are used for better utilization of CPU.

Protection

Considering a computer system having multiple users the concurrent execution of multiple processes, then the various processes must be protected from each another's activities.

Protection refers to mechanism or a way to control the access of programs, processes, or users to the resources defined by a computer system. Following are the major activities of an operating system with respect to protection.

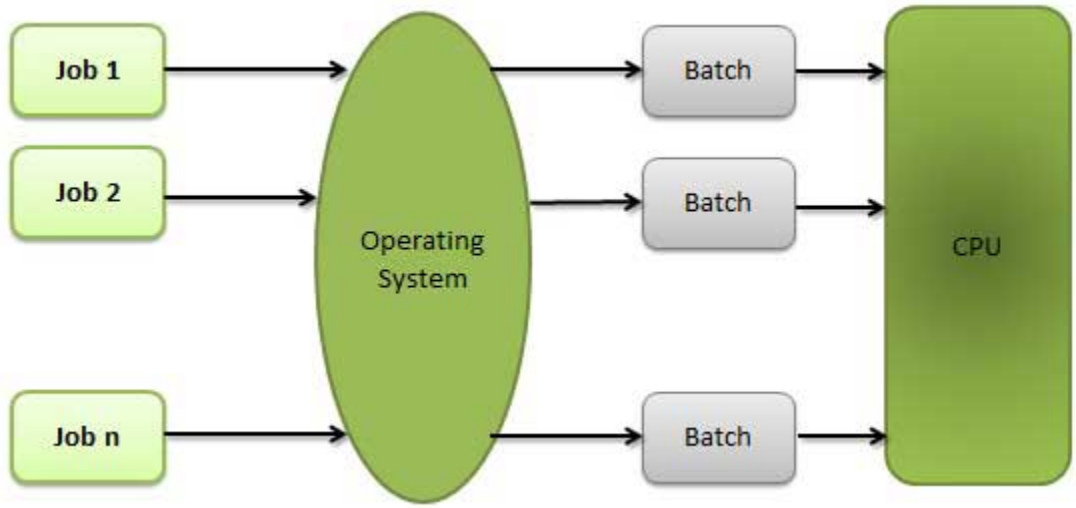
- OS ensures that all access to system resources is controlled.
- OS ensures that external I/O devices are protected from invalid access attempts.
- OS provides authentication feature for each user by means of a password.

Following are few of very important tasks that Operating System handles

Batch processing

Batch processing is a technique in which Operating System collects one programs and data together in a batch before processing starts. Operating system does the following activities related to batch processing.

- OS defines a job which has predefined sequence of commands, programs and data as a single unit.
- OS keeps a number a jobs in memory and executes them without any manual information.
- Jobs are processed in the order of submission i.e first come first served fashion.
- When job completes its execution, its memory is released and the output for the job gets copied into an output spool for later printing or processing.



ADVANTAGES

- Batch processing takes much of the work of the operator to the computer.
- Increased performance as a new job gets started as soon as the previous job finished without any manual intervention.

DISADVANTAGES

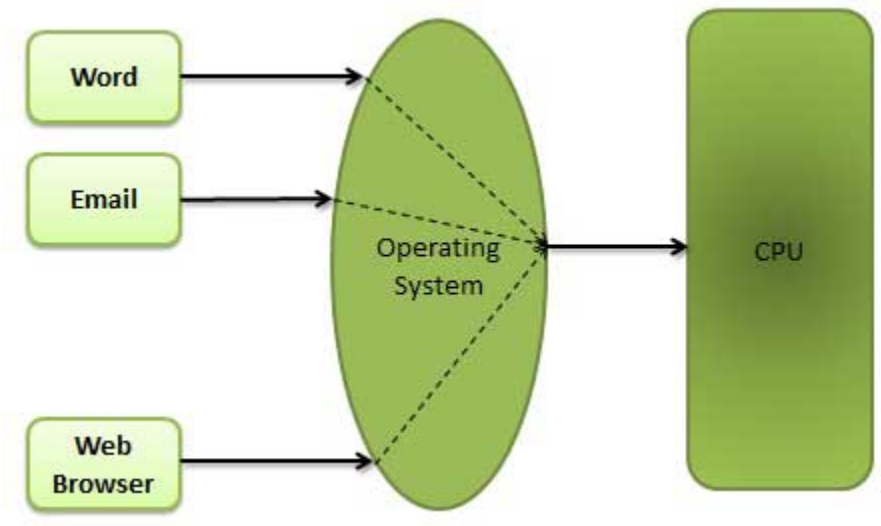
- Difficult to debug program.
- A job could enter an infinite loop.
- Due to lack of protection scheme, one batch job can affect pending jobs.

Multitasking

Multitasking refers to term where multiple jobs are executed by the CPU simultaneously by switching between them.Switches occur so frequently that the users may interact with each program while it is running. Operating system does the following activities related to multitasking.

- The user gives instructions to the operating system or to a program directly, and receives an immediate response.
- Operating System handles multitasking in the way that it can handle multiple operations / executes multiple programs at a time.

- Multitasking Operating Systems are also known as Time-sharing systems.
- These Operating Systems were developed to provide interactive use of a computer system at a reasonable cost.
- A time-shared operating system uses concept of CPU scheduling and multiprogramming to provide each user with a small portion of a time-shared CPU.
- Each user has at least one separate program in memory.

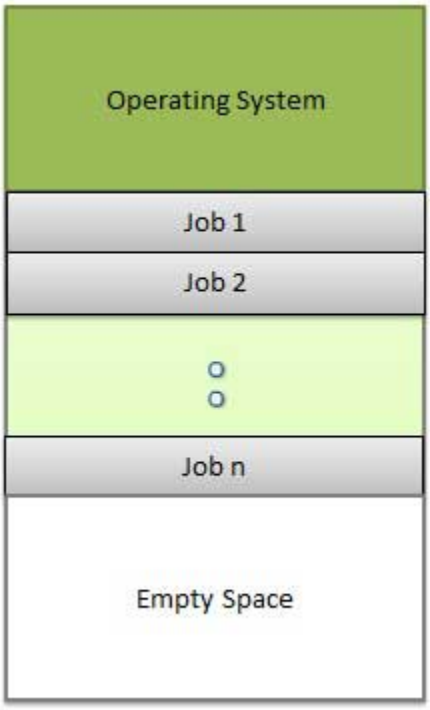


- A program that is loaded into memory and is executing is commonly referred to as a process.
- When a process executes, it typically executes for only a very short time before it either finishes or needs to perform I/O.
- Since interactive I/O typically runs at people speeds, it may take a long time to completed. During this time a CPU can be utilized by another process.
- Operating system allows the users to share the computer simultaneously. Since each action or command in a time-shared system tends to be short, only a little CPU time is needed for each user.
- As the system switches CPU rapidly from one user/program to the next, each user is given the impression that he/she has his/her own CPU, whereas actually one CPU is being shared among many users.

Multiprogramming

When two or more programs are residing in memory at the same time, then sharing the processor is referred to the multiprogramming. Multiprogramming assumes a single shared processor. Multiprogramming increases CPU utilization by organizing jobs so that the CPU always has one to execute.

Following figure shows the memory layout for a multiprogramming system.



Operating system does the following activities related to multiprogramming.

- The operating system keeps several jobs in memory at a time.
- This set of jobs is a subset of the jobs kept in the job pool.
- The operating system picks and begins to execute one of the job in the memory.
- Multiprogramming operating system monitors the state of all active programs and system resources using memory management programs to ensures that the CPU is never idle unless there are no jobs

ADVANTAGES

- High and efficient CPU utilization.
- User feels that many programs are allotted CPU almost simultaneously.

DISADVANTAGES

- CPU scheduling is required.
- To accommodate many jobs in memory, memory management is required.

Interactivity

Interactivity refers that a User is capable to interact with computer system. Operating system does the following activities related to interactivity.

- OS provides user an interface to interact with system.
- OS managers input devices to take inputs from the user. For example, keyboard.
- OS manages output devices to show outputs to the user. For example, Monitor.
- OS Response time needs to be short since the user submits and waits for the result.

Real Time System

Real time systems represents are usually dedicated, embedded systems. Operating system does the following activities related to real time system activity.

- In such systems, Operating Systems typically read from and react to sensor data.
- The Operating system must guarantee response to events within fixed periods of time to ensure correct performance.

Distributed Environment

Distributed environment refers to multiple independent CPUs or processors in a computer system. Operating system does the following activities related to distributed environment.

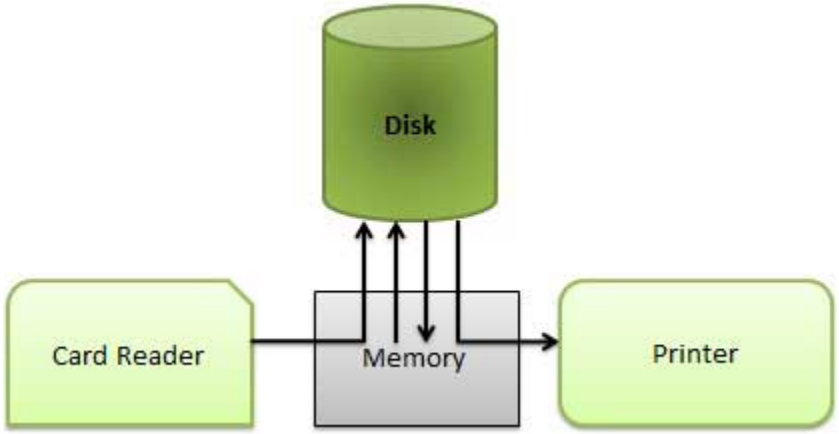
- OS Distributes computation logics among several physical processors.
- The processors do not share memory or a clock.
- Instead, each processor has its own local memory.

- OS manages the communications between the processors. They communicate with each other through various communication lines.

Spooling

Spooling is an acronym for simultaneous peripheral operations on line. Spooling refers to putting data of various I/O jobs in a buffer. This buffer is a special area in memory or hard disk which is accessible to I/O devices. Operating system does the following activities related to distributed environment.

- OS handles I/O device data spooling as devices have different data access rates.
- OS maintains the spooling buffer which provides a waiting station where data can rest while the slower device catches up.
- OS maintains parallel computation because of spooling process as a computer can perform I/O in parallel fashin. It becomes possible to have the computer read data from a tape, write data to disk and to write out to a tape printer while it is doing its computing task.



ADVANTAGES

- The spooling operation uses a disk as a very large buffer.
- Spooling is capable of overlapping I/O operation for one job with processor operations for another job.

Process

A process is a program in execution. The execution of a process must progress in a sequential fashion. Definition of process is following.

- A process is defined as an entity which represents the basic unit of work to be implemented in the system.

Components of process are following.

S.N.	Component & Description
1	**Object Program** Code to be executed.
2	**Data** Data to be used for executing the program.
3	**Resources** While executing the program, it may require some resources.
4	**Status** Verifies the status of the process execution.A process can run to completion only when all requested resources have been allocated to the process. Two or more processes could be executing the same program, each using their own data and resources.

Program

A program by itself is not a process. It is a static entity made up of program statement while process is a dynamic entity. Program contains the instructions to be executed by processor.

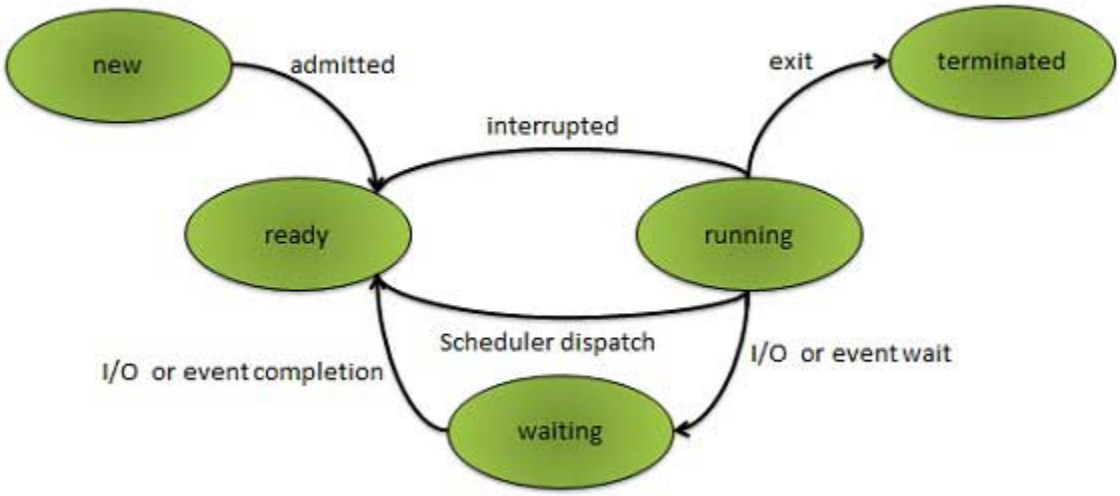
A program takes a space at single place in main memory and continues to stay there. A program does not perform any action by itself.

Process States

As a process executes, it changes state. The state of a process is defined as the current activity of the process.

Process can have one of the following five states at a time.

S.N.	State & Description
1	**New** The process is being created.
2	**Ready** The process is waiting to be assigned to a processor. Ready processes are waiting to have the processor allocated to them by the operating system so that they can run.
3	**Running** Process instructions are being executed (i.e. The process that is currently being executed).
4	**Waiting** The process is waiting for some event to occur (such as the completion of an I/O operation).
5	**Terminated** The process has finished execution.



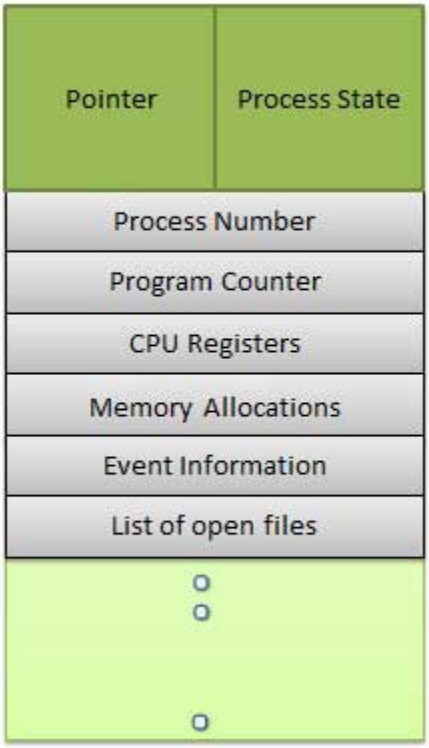
Process Control Block, PCB

Each process is represented in the operating system by a process control block (PCB) also called a task control block. PCB is the data structure used by the operating system. Operating system groups all information that needs about particular process.

PCB contains many pieces of information associated with a specific process which are described below.

S.N.	Information & Description
1	**Pointer** Pointer points to another process control block. Pointer is used for maintaining the scheduling list.
2	**ProcessState** Process state may be new, ready, running, waiting and so on.
3	**Program Counter** Program Counter indicates the address of the next instruction to be executed for this process.
4	**CPU registers** CPU registers include general purpose register, stack pointers, index registers and accumulators etc. number of register and type of register totally depends upon the computer architecture.
5	**Memory management information** This information may include the value of base and limit registers, the page tables, or the segment tables depending on the memory system used by the operating system. This information is useful for deallocating the memory when the process terminates.
6	**Accounting information** This information includes the amount of CPU and real time used, time limits,

	job or process numbers, account numbers etc.



Process control block includes CPU scheduling, I/O resource management, file management information etc.. The PCB serves as the repository for any information which can vary from process to process. Loader/linker sets flags and registers when a process is created. If that process gets suspended, the contents of the registers are saved on a stack and the pointer to the particular stack frame is stored in the PCB. By this technique, the hardware state can be restored so that the process can be scheduled to run again.

Definition

The process scheduling is the activity of the process manager that handles the removal of the running process from the CPU and the selection of another process on the basis of a particular strategy.

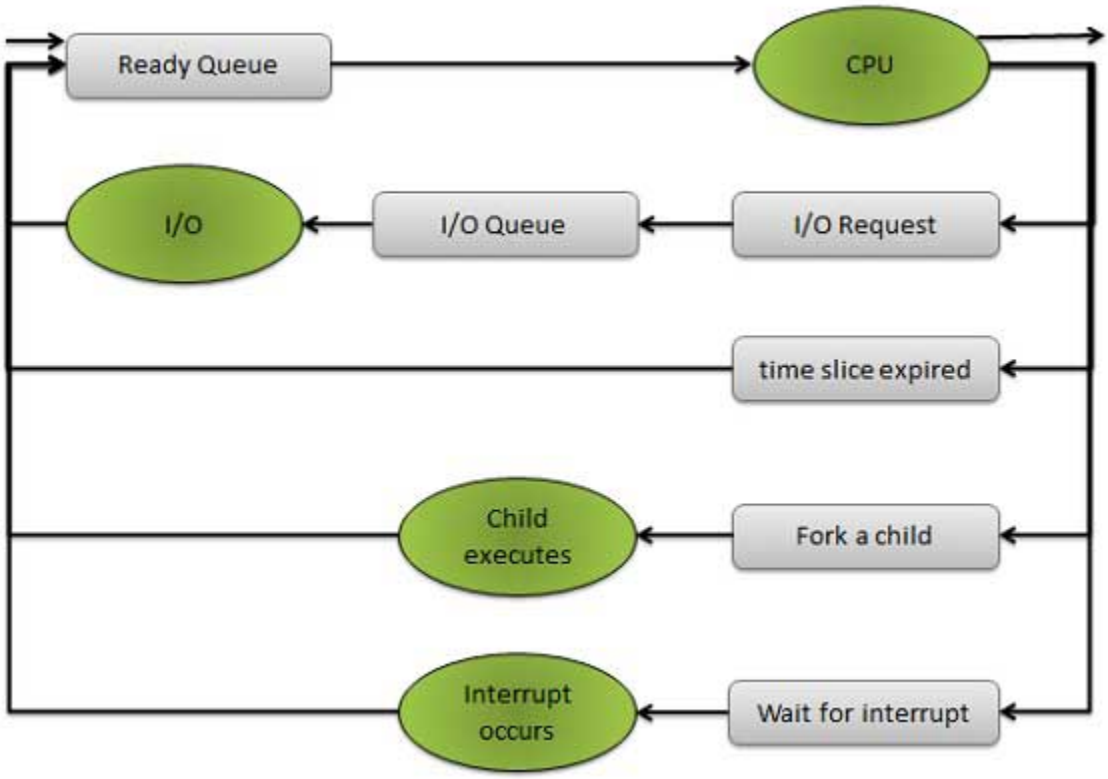
Process scheduling is an essential part of a Multiprogramming operating system. Such operating systems allow more than one process to be loaded into the executable memory at a time and loaded process shares the CPU using time multiplexing.

Scheduling Queues

Scheduling queues refers to queues of processes or devices. When the process enters into the system, then this process is put into a job queue. This queue consists of all processes in the system. The operating system also maintains other queues such as device queue. Device queue is a queue for which multiple processes are waiting for a particular I/O device. Each device has its own device queue.

This figure shows the queuing diagram of process scheduling.

- Queue is represented by rectangular box.
- The circles represent the resources that serve the queues.
- The arrows indicate the process flow in the system.



Queues are of two types

- Ready queue
- Device queue

A newly arrived process is put in the ready queue. Processes waits in ready queue for allocating the CPU. Once the CPU is assigned to a process, then that process will execute. While executing the process, any one of the following events can occur.

- The process could issue an I/O request and then it would be placed in an I/O queue.
- The process could create new sub process and will wait for its termination.
- The process could be removed forcibly from the CPU, as a result of interrupt and put back in the ready queue.

Two State Process Model

Two state process models refers to running and non-running states which are described below.

S.N.	State & Description
1	**Running** When new process is created by Operating System that process enters into the system as in the running state.
2	**Not Running** Processes that are not running are kept in queue, waiting for their turn to execute. Each entry in the queue is a pointer to a particular process. Queue is implemented by using linked list. Use of dispatcher is as follows. When a process is interrupted, that process is transferred in the waiting queue. If the process has completed or aborted, the process is discarded. In either case, the dispatcher then selects a process from the queue to execute.

Schedulers

Schedulers are special system software's which handles process scheduling in various ways.Their main task is to select the jobs to be submitted into the system and to decide which process to run. Schedulers are of three types

- Long Term Scheduler

- Short Term Scheduler
- Medium Term Scheduler

Long Term Scheduler

It is also called job scheduler. Long term scheduler determines which programs are admitted to the system for processing. Job scheduler selects processes from the queue and loads them into memory for execution. Process loads into the memory for CPU scheduling. The primary objective of the job scheduler is to provide a balanced mix of jobs, such as I/O bound and processor bound. It also controls the degree of multiprogramming. If the degree of multiprogramming is stable, then the average rate of process creation must be equal to the average departure rate of processes leaving the system.

On some systems, the long term scheduler may not be available or minimal. Time-sharing operating systems have no long term scheduler. When process changes the state from new to ready, then there is use of long term scheduler.

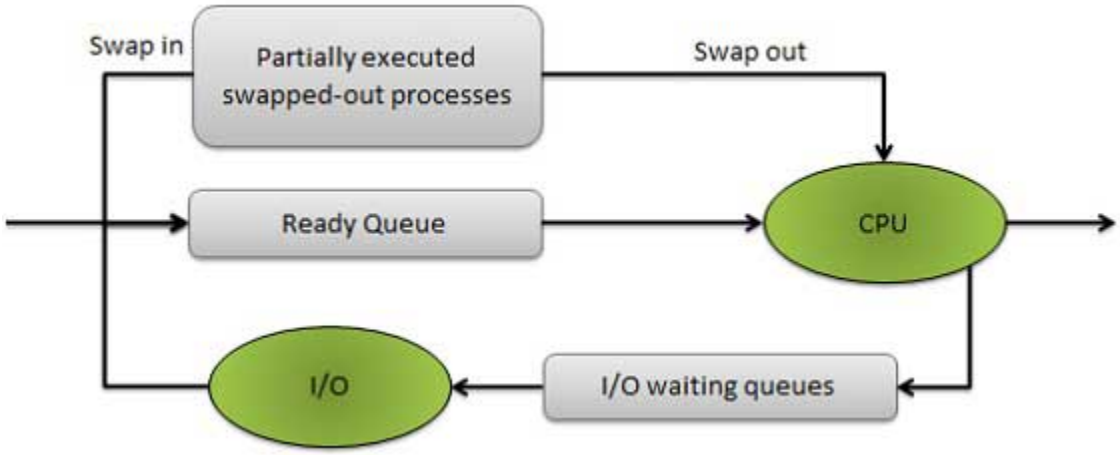
Short Term Scheduler

It is also called CPU scheduler. Main objective is increasing system performance in accordance with the chosen set of criteria. It is the change of ready state to running state of the process. CPU scheduler selects process among the processes that are ready to execute and allocates CPU to one of them.

Short term scheduler also known as dispatcher, execute most frequently and makes the fine grained decision of which process to execute next. Short term scheduler is faster than long term scheduler.

Medium Term Scheduler

Medium term scheduling is part of the swapping. It removes the processes from the memory. It reduces the degree of multiprogramming. The medium term scheduler is in-charge of handling the swapped out-processes.



Running process may become suspended if it makes an I/O request. Suspended processes cannot make any progress towards completion. In this condition, to remove the process from memory and make space for other process, the suspended process is moved to the secondary storage. This process is called swapping, and the process is said to be swapped out or rolled out. Swapping may be necessary to improve the process mix.

Comparison between Scheduler

S.N.	Long Term Scheduler	Short Term Scheduler	Medium Term Scheduler
1	It is a job scheduler	It is a CPU scheduler	It is a process swapping scheduler.
2	Speed is lesser than short term scheduler	Speed is fastest among other two	Speed is in between both short and long term scheduler.
3	It controls the degree of multiprogramming	It provides lesser control over degree of multiprogramming	It reduces the degree of multiprogramming.
4	It is almost absent or minimal in time sharing system	It is also minimal in time sharing system	It is a part of Time sharing systems.

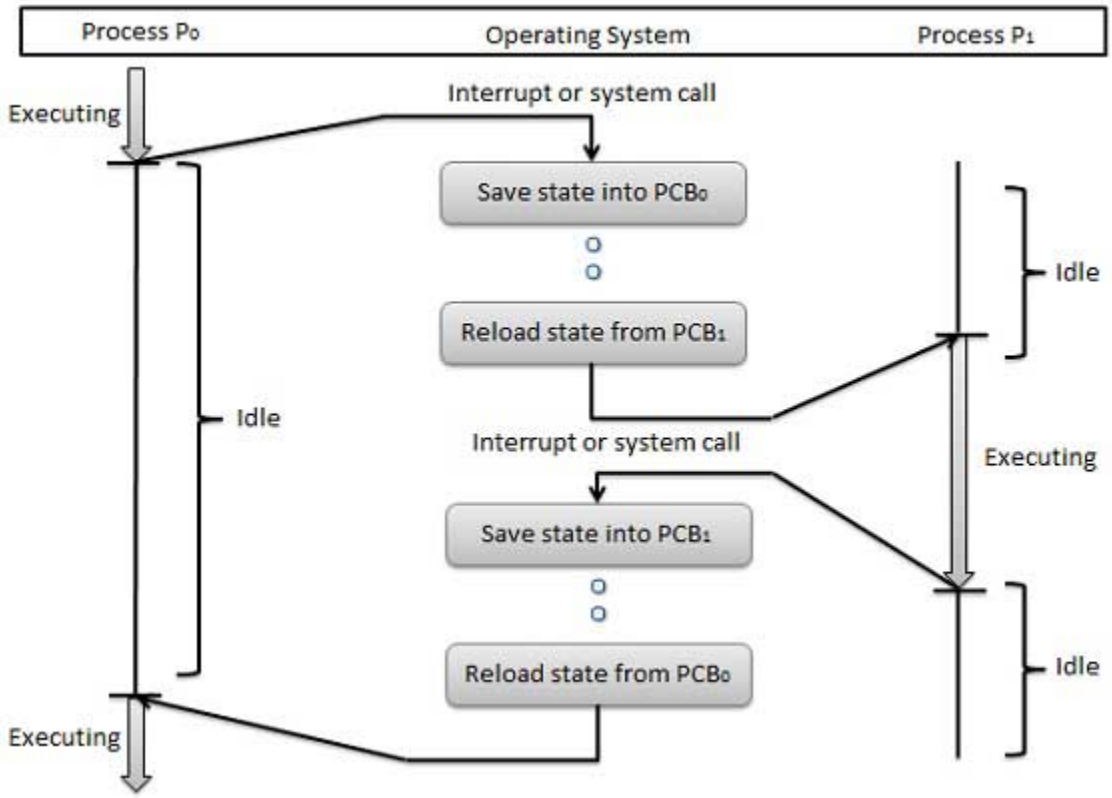
5	It selects processes from pool and loads them into memory for execution	It selects those processes which are ready to execute	It can re-introduce the process into memory and execution can be continued.

Context Switch

A context switch is the mechanism to store and restore the state or context of a CPU in Process Control block so that a process execution can be resumed from the same point at a later time. Using this technique a context switcher enables multiple processes to share a single CPU. Context switching is an essential part of a multitasking operating system features.

When the scheduler switches the CPU from executing one process to execute another, the context switcher saves the content of all processor registers for the process being removed from the CPU, in its process descriptor. The context of a process is represented in the process control block of a process.

Context switch time is pure overhead. Context switching can significantly affect performance as modern computers have a lot of general and status registers to be saved. Content switching times are highly dependent on hardware support. Context switch requires (n + m) bxK time units to save the state of the processor with n general registers, assuming b are the store operations are required to save n and m registers of two process control blocks and each store instruction requires K time units.



Some hardware systems employ two or more sets of processor registers to reduce the amount of context switching time. When the process is switched, the following information is stored.

- Program Counter
- Scheduling Information
- Base and limit register value
- Currently used register
- Changed State
- I/OState
- Accounting

We'll discuss four major scheduling algorithms here which are following

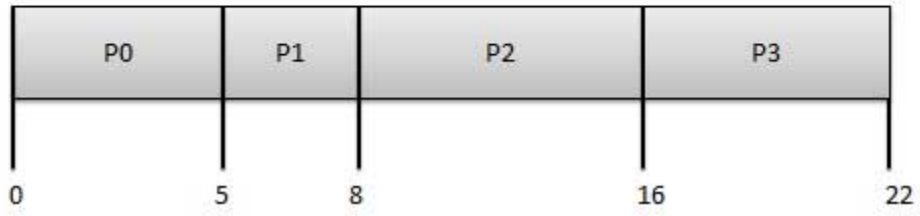
- First Come First Serve (FCFS) Scheduling
- Shortest-Job-First (SJF) Scheduling

- Priority Scheduling
- Round Robin(RR) Scheduling
- Multilevel Queue Scheduling

First Come First Serve (FCFS)

- Jobs are executed on first come, first serve basis.
- Easy to understand and implement.
- Poor in performance as average wait time is high.

Process	Arrival Time	Execute Time	Service Time
P0	0	5	0
P1	1	3	5
P2	2	8	8
P3	3	6	16



Wait time of each process is following

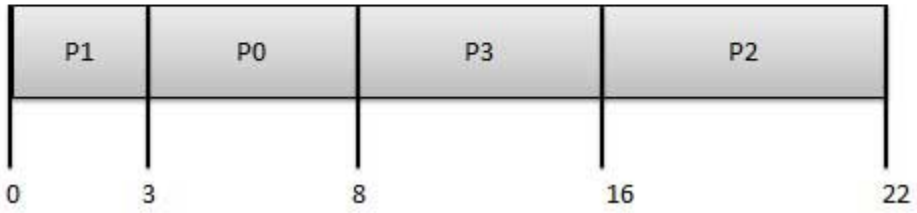
Process	Wait Time : Service Time - Arrival Time
P0	0 - 0 = 0
P1	5 - 1 = 4
P2	8 - 2 = 6
P3	16 - 3 = 13

Average Wait Time: (0+4+6+13) / 4 = 5.55

Shortest Job First (SJF)

- Best approach to minimize waiting time.
- Impossible to implement
- Processer should know in advance how much time process will take.

Process	Arrival Time	Execute Time	Service Time
P0	0	5	0
P1	1	3	3
P2	2	8	8
P3	3	6	16



Wait time of each process is following

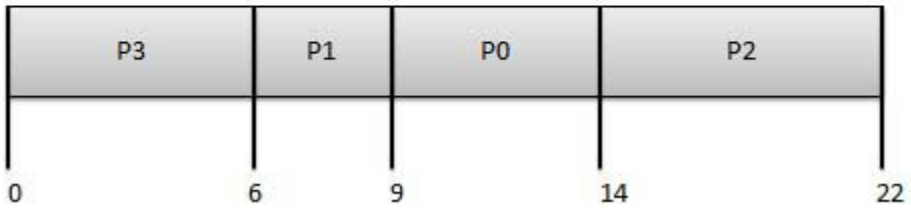
Process	Wait Time : Service Time - Arrival Time
P0	3 - 0 = 3
P1	0 - 0 = 0
P2	16 - 2 = 14
P3	8 - 3 = 5

Average Wait Time: (3+0+14+5) / 4 = 5.50

Priority Based Scheduling

- Each process is assigned a priority. Process with highest priority is to be executed first and so on.
- Processes with same priority are executed on first come first serve basis.
- Priority can be decided based on memory requirements, time requirements or any other resource requirement.

Process	Arrival Time	Execute Time	Priority	Service Time
P0	0	5	1	0
P1	1	3	2	3
P2	2	8	1	8
P3	3	6	3	16



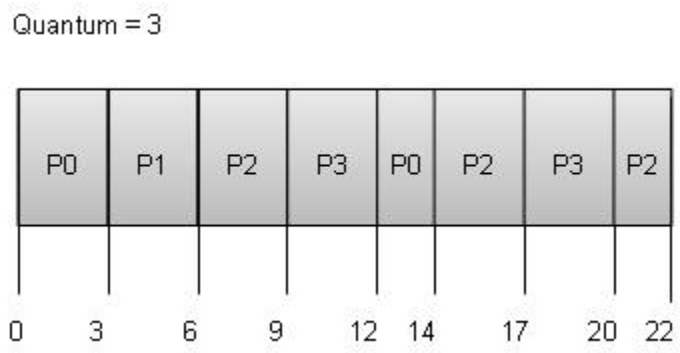
Wait time of each process is following

Process	Wait Time : Service Time - Arrival Time
P0	0 - 0 = 0
P1	3 - 1 = 2
P2	8 - 2 = 6
P3	16 - 3 = 13

Average Wait Time: (0+2+6+13) / 4 = 5.25

Round Robin Scheduling

- Each process is provided a fix time to execute called quantum.
- Once a process is executed for given time period. Process is preempted and other process executes for given time period.
- Context switching is used to save states of preempted processes.



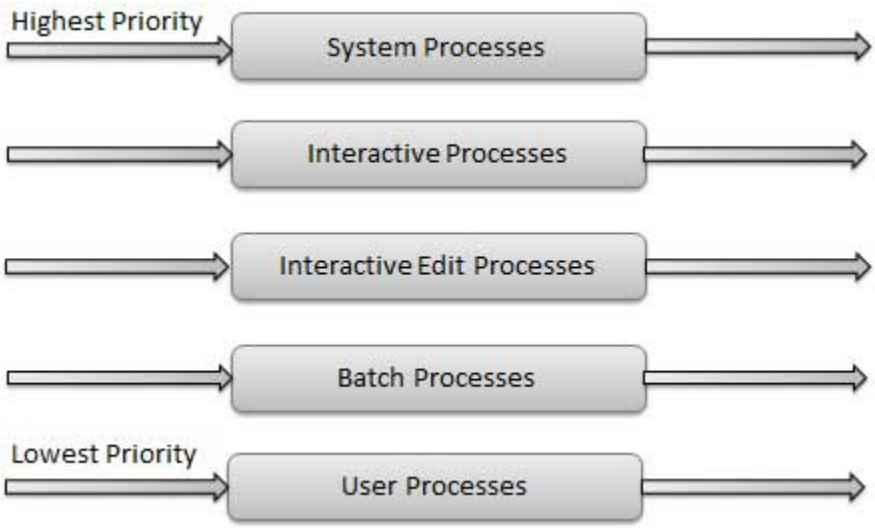
Wait time of each process is following

Process	Wait Time : Service Time - Arrival Time
P0	(0-0) + (12-3) = 9
P1	(3-1) = 2
P2	(6-2) + (14-9) + (20-17) = 12
P3	(9-3) + (17-12) = 11

Average Wait Time: (9+2+12+11) / 4 = 8.5

Multi Queue Scheduling

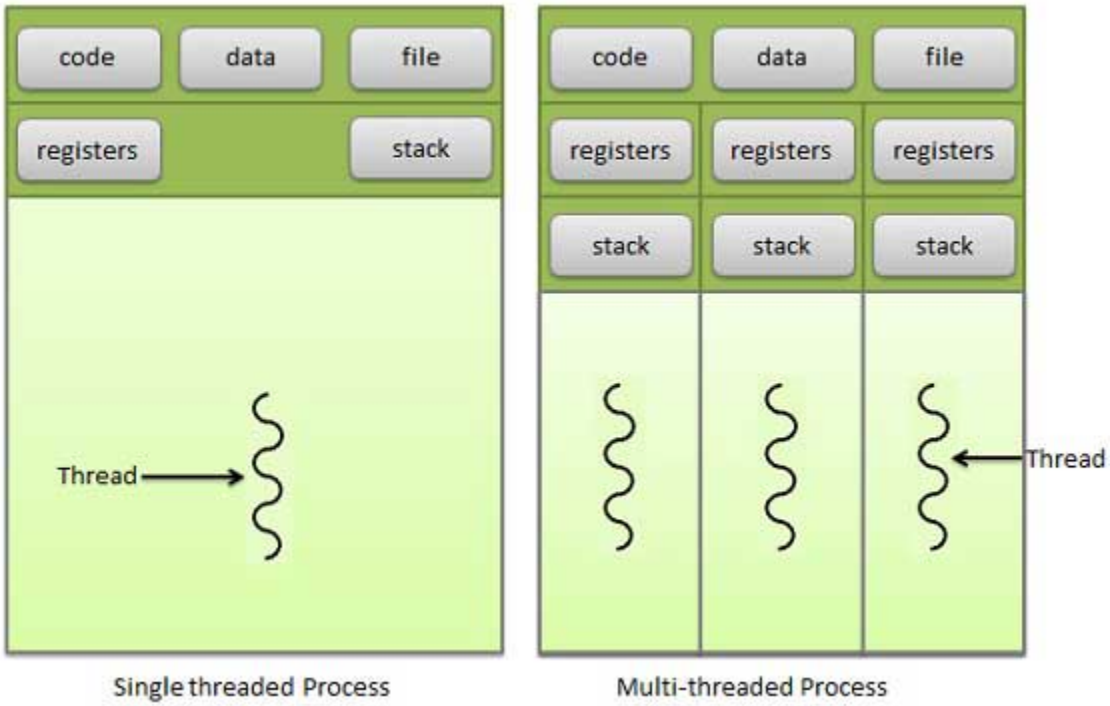
- Multiple queues are maintained for processes.
- Each queue can have its own scheduling algorithms.
- Priorities are assigned to each queue.



Thread:

A thread is a flow of execution through the process code, with its own program counter, system registers and stack. A thread is also called a light weight process. Threads provide a way to improve application performance through parallelism. Threads represent a software approach to improving performance of operating system by reducing the overhead thread is equivalent to a classical process.

Each thread belongs to exactly one process and no thread can exist outside a process. Each thread represents a separate flow of control.Threads have been successfully used in implementing network servers and web server. They also provide a suitable foundation for parallel execution of applications on shared memory multiprocessors. Following figure shows the working of the single and multithreaded processes.



Difference between Process and Thread

S.N.	Process	Thread
1	Process is heavy weight or resource intensive.	Thread is light weight taking lesser resources than a process.
1	Process switching needs interaction with operating system.	Thread switching does not need to interact with operating system.
1	In multiple processing environments each process executes the same code but has its own memory and file resources.	All threads can share same set of open files, child processes.
1	If one process is blocked then no other	While one thread is blocked and

	process can execute until the first process is unblocked.	waiting, second thread in the same task can run.
1	Multiple processes without using threads use more resources.	Multiple threaded processes use fewer resources.
1	In multiple processes each process operates independently of the others.	One thread can read, write or change another thread's data.

Advantages of Thread

- Thread minimizes context switching time.
- Use of threads provides concurrency within a process.
- Efficient communication.
- Economy- It is more economical to create and context switch threads.
- Utilization of multiprocessor architectures to a greater scale and efficiency.

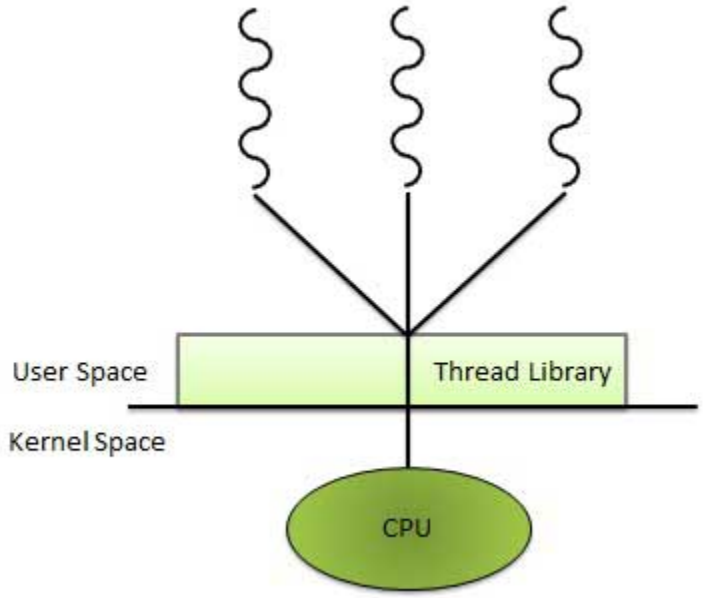
Types of Thread

Threads are implemented in following two ways

- **User Level Threads** -- User managed threads
- **Kernel Level Threads** -- Operating System managed threads acting on kernel, an operating system core.

User Level Threads

In this case, application manages thread management kernel is not aware of the existence of threads. The thread library contains code for creating and destroying threads, for passing message and data between threads, for scheduling thread execution and for saving and restoring thread contexts. The application begins with a single thread and begins running in that thread.



ADVANTAGES

- Thread switching does not require Kernel mode privileges.
- User level thread can run on any operating system.
- Scheduling can be application specific in the user level thread.
- User level threads are fast to create and manage.

DISADVANTAGES

- In a typical operating system, most system calls are blocking.
- Multithreaded application cannot take advantage of multiprocessing.

Kernel Level Threads

In this case, thread management done by the Kernel. There is no thread management code in the application area. Kernel threads are supported directly by the operating system. Any application can

be programmed to be multithreaded. All of the threads within an application are supported within a single process.

The Kernel maintains context information for the process as a whole and for individuals threads within the process. Scheduling by the Kernel is done on a thread basis. The Kernel performs thread creation, scheduling and management in Kernel space. Kernel threads are generally slower to create and manage than the user threads.

ADVANTAGES

- Kernel can simultaneously schedule multiple threads from the same process on multiple processes.
- If one thread in a process is blocked, the Kernel can schedule another thread of the same process.
- Kernel routines themselves can multithreaded.

DISADVANTAGES

- Kernel threads are generally slower to create and manage than the user threads.
- Transfer of control from one thread to another within same process requires a mode switch to the Kernel.

Multithreading Models

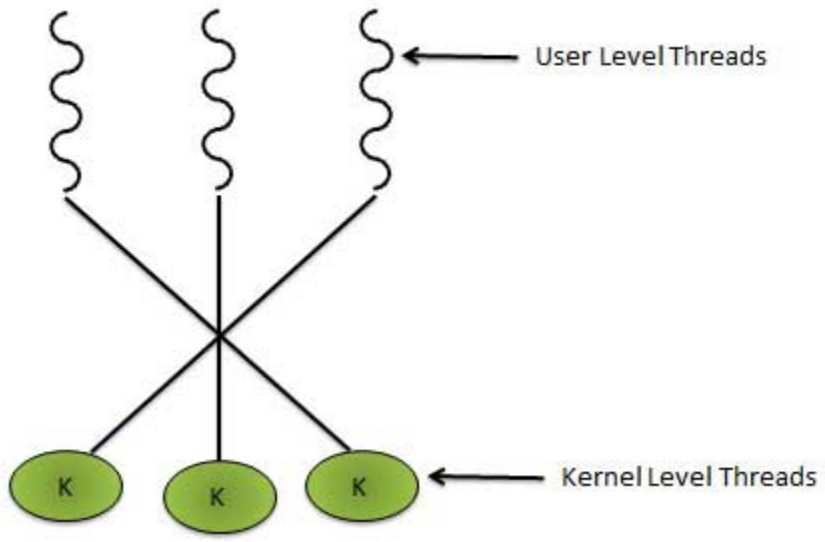
Some operating system provides a combined user level thread and Kernel level thread facility. Solaris is a good example of this combined approach. In a combined system, multiple threads within the same application can run in parallel on multiple processors and a blocking system call need not block the entire process. Multithreading models are three types

- Many to many relationship.
- Many to one relationship.
- One to one relationship.

Many to Many Model

In this model, many user level threads multiplexes to the Kernel thread of smaller or equal numbers. The number of Kernel threads may be specific to either a particular application or a particular machine.

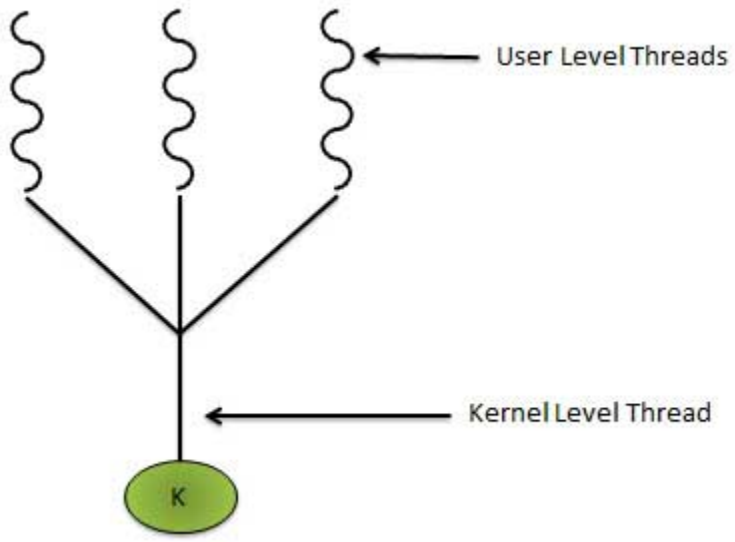
Following diagram shows the many to many model. In this model, developers can create as many user threads as necessary and the corresponding Kernel threads can run in parallels on a multiprocessor.



Many to One Model

Many to one model maps many user level threads to one Kernel level thread. Thread management is done in user space. When thread makes a blocking system call, the entire process will be blocks. Only one thread can access the Kernel at a time,so multiple threads are unable to run in parallel on multiprocessors.

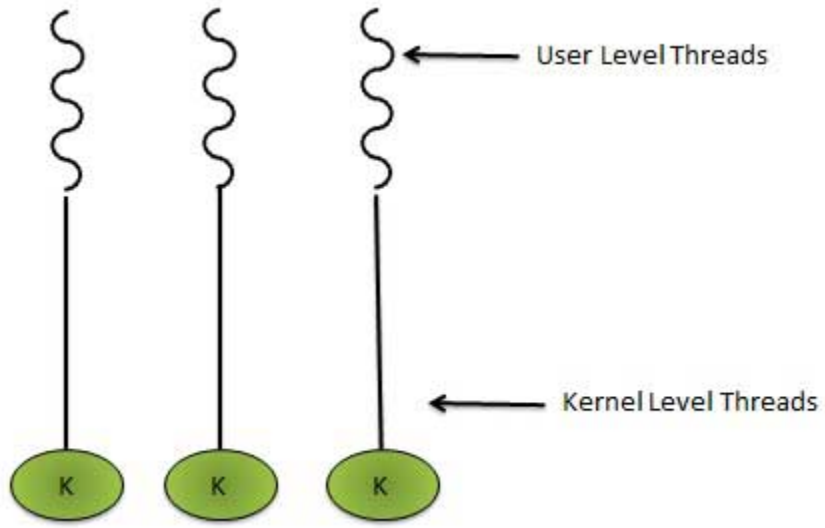
If the user level thread libraries are implemented in the operating system in such a way that system does not support them then Kernel threads use the many to one relationship modes.



One to One Model

There is one to one relationship of user level thread to the kernel level thread.This model provides more concurrency than the many to one model. It also has another thread to run when a thread makes a blocking system call. It supports multiple thread to execute in parallel on microprocessors.

Disadvantage of this model is that creating user thread requires the corresponding Kernel thread. OS/2, windows NT and windows 2000 use one to one relationship model.



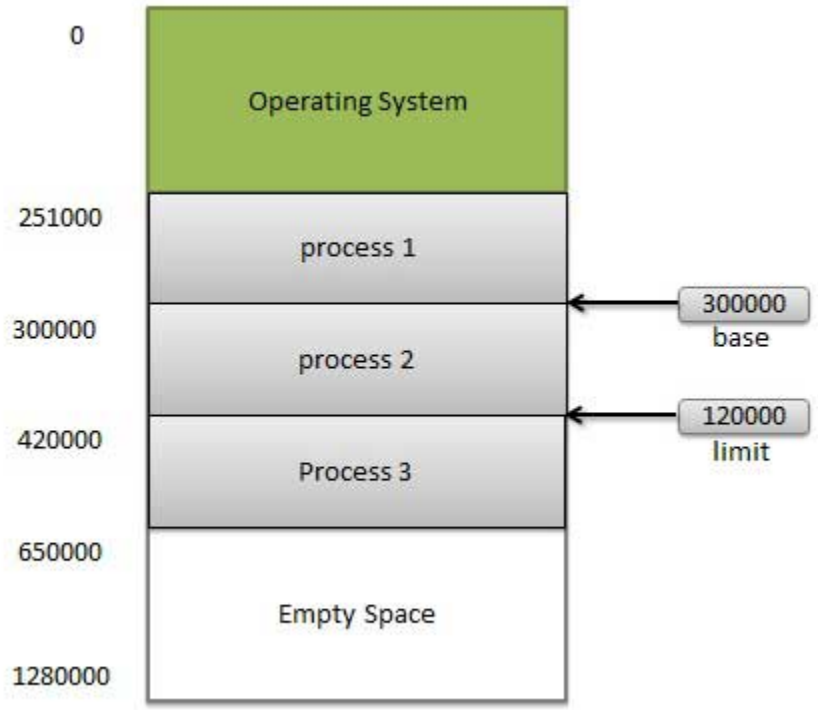
Difference between User Level & Kernel Level Thread

S.N.	User Level Threads	Kernel Level Thread

1	User level threads are faster to create and manage.	Kernel level threads are slower to create and manage.
2	Implementation is by a thread library at the user level.	Operating system supports creation of Kernel threads.
3	User level thread is generic and can run on any operating system.	Kernel level thread is specific to the operating system.
4	Multi-threaded application cannot take advantage of multiprocessing.	Kernel routines themselves can be multithreaded.

Memory management is the functionality of an operating system which handles or manages primary memory. Memory management keeps track of each and every memory location either it is allocated to some process or it is free. It checks how much memory is to be allocated to processes. It decides which process will get memory at what time. It tracks whenever some memory gets freed or unallocated and correspondingly it updates the status.

Memory management provides protection by using two registers, a base register and a limit register. The base register holds the smallest legal physical memory address and the limit register specifies the size of the range. For example, if the base register holds 300000 and the limit register is 1209000, then the program can legally access all addresses from 300000 through 411999.



Instructions and data to memory addresses can be done in following ways

- **Compile time** -- When it is known at compile time where the process will reside, compile time binding is used to generate the absolute code.
- **Load time** -- When it is not known at compile time where the process will reside in memory, then the compiler generates re-locatable code.
- **Execution time** -- If the process can be moved during its execution from one memory segment to another, then binding must be delayed to be done at run time

Dynamic Loading

In dynamic loading, a routine of a program is not loaded until it is called by the program. All routines are kept on disk in a re-locatable load format. The main program is loaded into memory and is executed. Other routines methods or modules are loaded on request. Dynamic loading makes better memory space utilization and unused routines are never loaded.

Dynamic Linking

Linking is the process of collecting and combining various modules of code and data into a executable file that can be loaded into memory and executed. Operating system can link system level libraries to

a program. When it combines the libraries at load time, the linking is called static linking and when this linking is done at the time of execution, it is called as dynamic linking.

In static linking, libraries linked at compile time, so program code size becomes bigger whereas in dynamic linking libraries linked at execution time so program code size remains smaller.

Logical versus Physical Address Space

An address generated by the CPU is a logical address whereas address actually available on memory unit is a physical address. Logical address is also known a Virtual address.

Virtual and physical addresses are the same in compile-time and load-time address-binding schemes. Virtual and physical addresses differ in execution-time address-binding scheme.

The set of all logical addresses generated by a program is referred to as a logical address space. The set of all physical addresses corresponding to these logical addresses is referred to as a physical address space.

The run-time mapping from virtual to physical address is done by the memory management unit (MMU) which is a hardware device. MMU uses following mechanism to convert virtual address to physical address

- The value in the base register is added to every address generated by a user process which is treated as offset at the time it is sent to memory. For example, if the base register value is 10000, then an attempt by the user to use address location 100 will be dynamically reallocated to location 10100.
- The user program deals with virtual addresses; it never sees the real physical addresses.

Swapping

Swapping is a mechanism in which a process can be swapped temporarily out of main memory to a backing store, and then brought back into memory for continued execution.

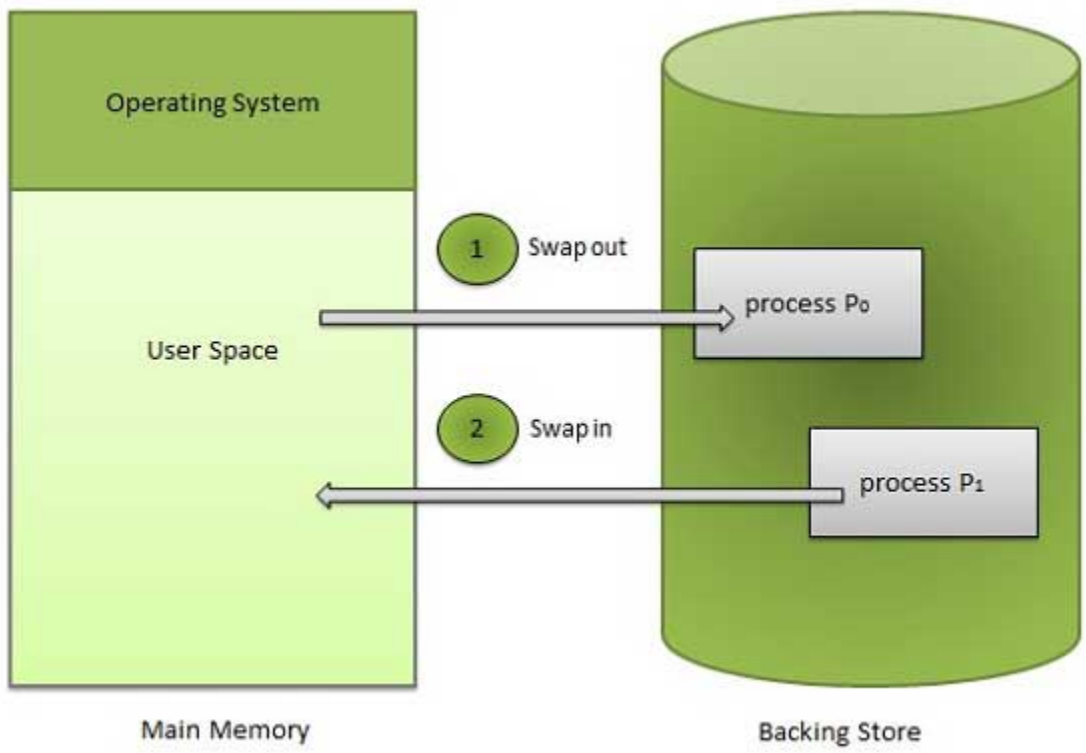
Backing store is a usually a hard disk drive or any other secondary storage which fast in access and large enough to accommodate copies of all memory images for all users. It must be capable of providing direct access to these memory images.

Major time consuming part of swapping is transfer time. Total transfer time is directly proportional to the amount of memory swapped. Let us assume that the user process is of size 100KB and the backing store is a standard hard disk with transfer rate of 1 MB per second. The actual transfer of the 100K process to or from memory will take

100KB / 1000KB per second

= 1/10 second

= 100 milliseconds



Memory Allocation

Main memory usually has two partitions

- **Low Memory** -- Operating system resides in this memory.
- **High Memory** -- User processes then held in high memory.

Operating system uses the following memory allocation mechanism.

S.N.	Memory Allocation	Description

1	**Single-partition allocation**	In this type of allocation, relocation-register scheme is used to protect user processes from each other, and from changing operating-system code and data. Relocation register contains value of smallest physical address whereas limit register contains range of logical addresses. Each logical address must be less than the limit register.
2	**Multiple-partition allocation**	In this type of allocation, main memory is divided into a number of fixed-sized partitions where each partition should contain only one process. When a partition is free, a process is selected from the input queue and is loaded into the free partition. When the process terminates, the partition becomes available for another process.

Fragmentation

As processes are loaded and removed from memory, the free memory space is broken into little pieces. It happens after sometimes that processes can not be allocated to memory blocks considering their small size and memory blocks remains unused. This problem is known as Fragmentation.

Fragmentation is of two types

S.N.	Fragmentation	Description
1	**External fragmentation**	Total memory space is enough to satisfy a request or to reside a process in it, but it is not contiguous so it can not be used.
2	**Internal fragmentation**	Memory block assigned to process is bigger. Some portion of memory is left unused as it can not be used by another process.

External fragmentation can be reduced by compaction or shuffle memory contents to place all free memory together in one large block. To make compaction feasible, relocation should be dynamic.

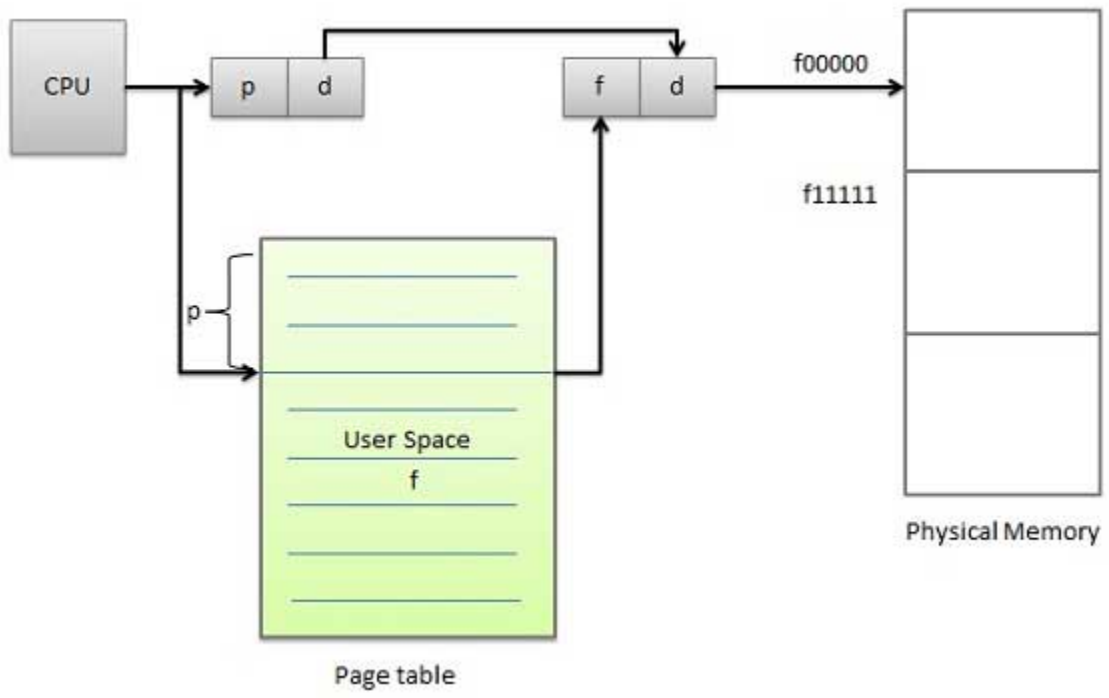
Paging

External fragmentation is avoided by using paging technique. Paging is a technique in which physical memory is broken into blocks of the same size called pages (size is power of 2, between 512 bytes and 8192 bytes). When a process is to be executed, it's corresponding pages are loaded into any available memory frames.

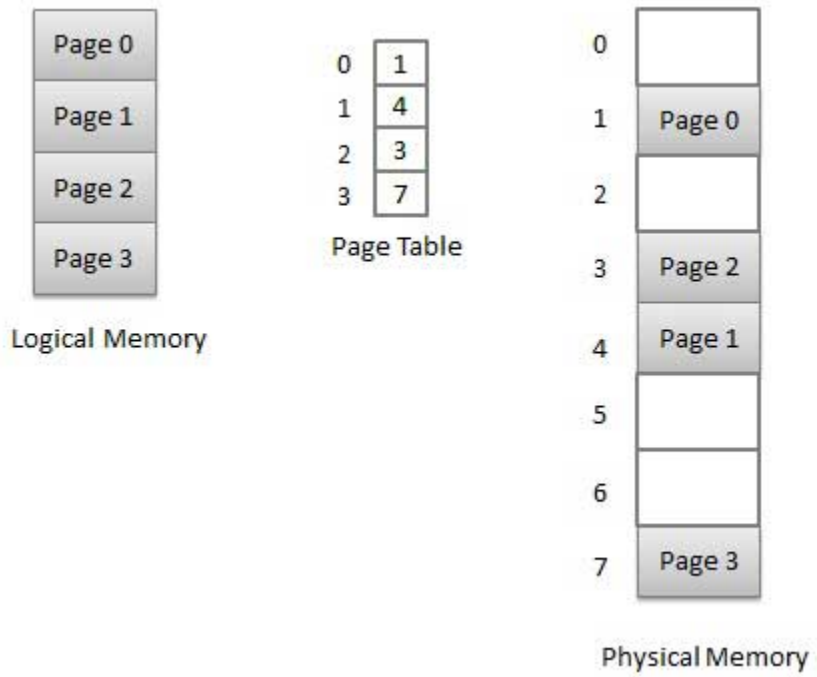
Logical address space of a process can be non-contiguous and a process is allocated physical memory whenever the free memory frame is available. Operating system keeps track of all free frames. Operating system needs n free frames to run a program of size n pages.

Address generated by CPU is divided into

- **Page number (p)** -- page number is used as an index into a page table which contains base address of each page in physical memory.
- **Page offset (d)** -- page offset is combined with base address to define the physical memory address.



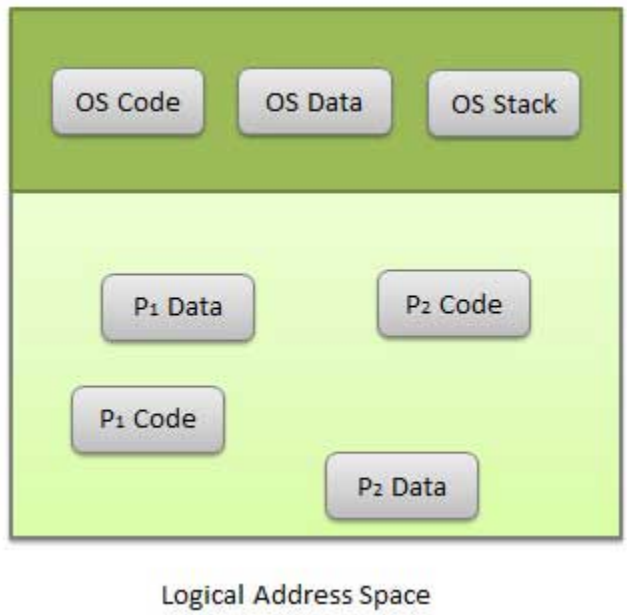
Following figure show the paging table architecture.



Segmentation

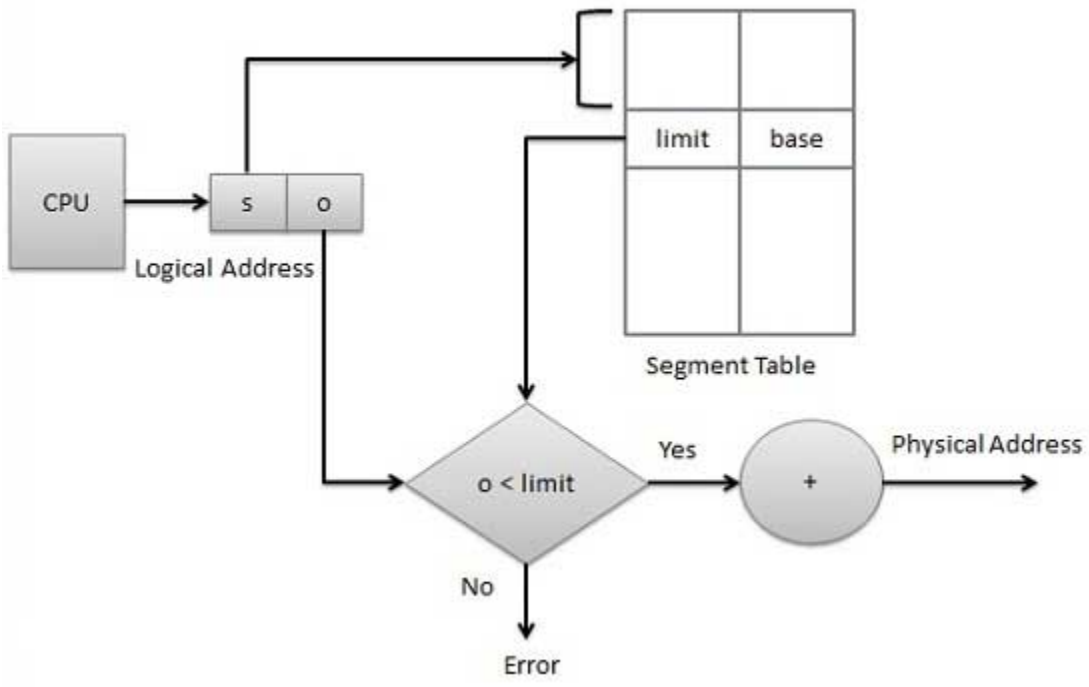
Segmentation is a technique to break memory into logical pieces where each piece represents a group of related information. For example, data segments or code segment for each process, data segment for operating system and so on. Segmentation can be implemented using or without using paging.

Unlike paging, a segment is having varying sizes and thus eliminates internal fragmentation. External fragmentation still exists but to lesser extent.



Address generated by CPU is divided into

- **Segment number (s)** -- segment number is used as an index into a segment table which contains base address of each segment in physical memory and a limit of segment.
- **Segment offset (o)** -- segment offset is first checked against limit and then is combined with base address to define the physical memory address.

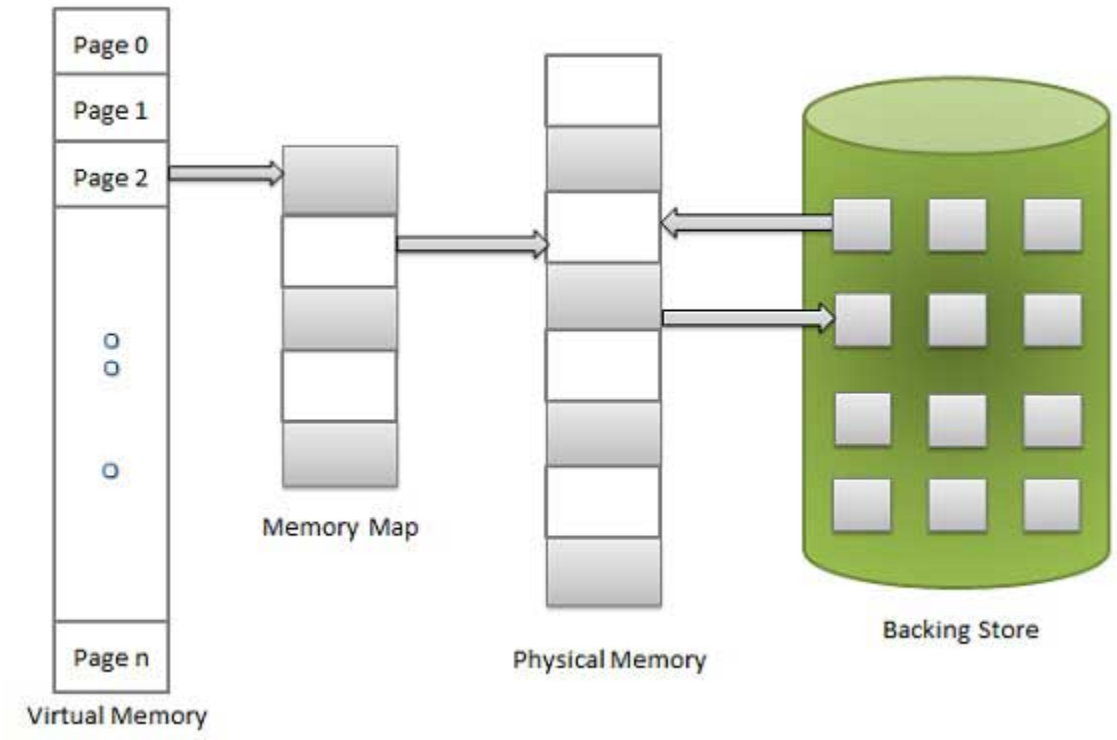


Virtual memory is a technique that allows the execution of processes which are not completely available in memory. The main visible advantage of this scheme is that programs can be larger than physical memory. Virtual memory is the separation of user logical memory from physical memory.

This separation allows an extremely large virtual memory to be provided for programmers when only a smaller physical memory is available. Following are the situations, when entire program is not required to be loaded fully in main memory.

- User written error handling routines are used only when an error occurred in the data or computation.
- Certain options and features of a program may be used rarely.
- Many tables are assigned a fixed amount of address space even though only a small amount of the table is actually used.

- The ability to execute a program that is only partially in memory would counter many benefits.
- Less number of I/O would be needed to load or swap each user program into memory.
- A program would no longer be constrained by the amount of physical memory that is available.
- Each user program could take less physical memory, more programs could be run the same time, with a corresponding increase in CPU utilization and throughput.



Virtual memory is commonly implemented by demand paging. It can also be implemented in a segmentation system. Demand segmentation can also be used to provide virtual memory.

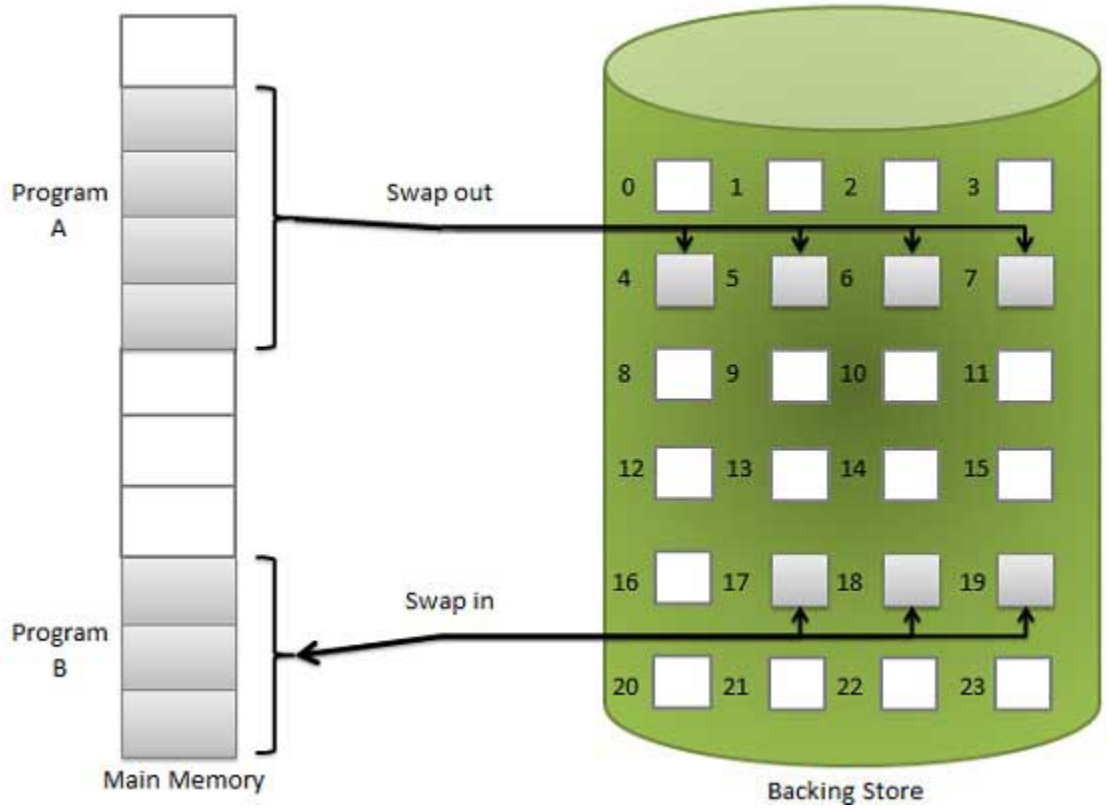
Demand Paging

A demand paging system is quite similar to a paging system with swapping. When we want to execute a process, we swap it into memory. Rather than swapping the entire process into memory, however, we use a lazy swapper called pager.

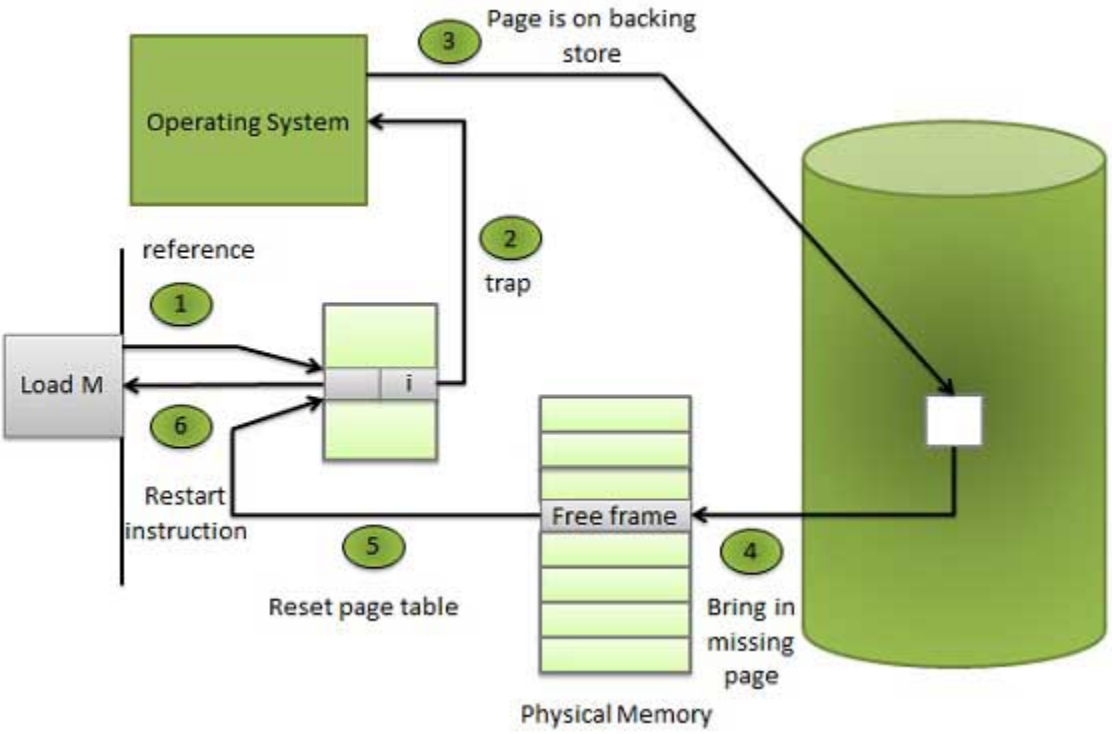
When a process is to be swapped in, the pager guesses which pages will be used before the process is swapped out again Instead of swapping in a whole process, the pager brings only those necessary

pages into memory. Thus, it avoids reading into memory pages that will not be used in anyway, decreasing the swap time and the amount of physical memory needed.

Hardware support is required to distinguish between those pages that are in memory and those pages that are on the disk using the valid-invalid bit scheme. Where valid and invalid pages can be checked by checking the bit Marking a page will have no effect if the process never attempts to access the page. While the process executes and accesses pages that are memory resident, execution proceeds normally.



Access to a page marked invalid causes a **page-fault trap**. This trap is the result of the operating system's failure to bring the desired page into memory. But page fault can be handled as following



Step	Description
Step 1	Check an internal table for this process, to determine whether the reference was a valid or it was an invalid memory access.
Step 2	If the reference was invalid, terminate the process. If it was valid, but page have not yet brought in, page in the latter.
Step 3	Find a free frame.
Step 4	Schedule a disk operation to read the desired page into the newly allocated frame.
Step 5	When the disk read is complete, modify the internal table kept with the process and the page table to indicate that the page is now in memory.
Step 6	Restart the instruction that was interrupted by the illegal address trap.

	The process can now access the page as though it had always been in memory. Therefore, the operating system reads the desired page into memory and restarts the process as though the page had always been in memory.

ADVANTAGES

Following are the advantages of Demand Paging

- Large virtual memory.
- More efficient use of memory.
- Unconstrained multiprogramming. There is no limit on degree of multiprogramming.

DISADVANTAGES

Following are the disadvantages of Demand Paging

- Number of tables and amount of processor overhead for handling page interrupts are greater than in the case of the simple paged management techniques.
- Due to the lack of an explicit constraint on jobs address space size.

Page Replacement Algorithm

Page replacement algorithms are the techniques using which Operating System decides which memory pages to swap out, write to disk when a page of memory needs to be allocated. Paging happens whenever a page fault occurs and a free page cannot be used for allocation purpose accounting to reason that pages are not available or the number of free pages is lower than required pages.

When the page that was selected for replacement and was paged out, is referenced again then it has to read in from disk, and this requires for I/O completion. This process determines the quality of the page replacement algorithm: the lesser the time waiting for page-ins, the better is the algorithm. A page replacement algorithm looks at the limited information about accessing the pages provided by hardware, and tries to select which pages should be replaced to minimize the total number of page

misses, while balancing it with the costs of primary storage and processor time of the algorithm itself. There are many different page replacement algorithms. We evaluate an algorithm by running it on a particular string of memory reference and computing the number of page faults.

Reference String

The string of memory references is called reference string. Reference strings are generated artificially or by tracing a given system and recording the address of each memory reference. The latter choice produces a large number of data, where we note two things.

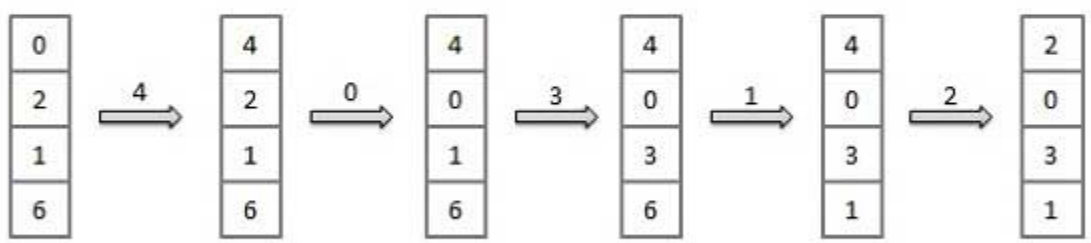
- For a given page size we need to consider only the page number, not the entire address.
- If we have a reference to a page p, then any immediately following references to page p will never cause a page fault. Page p will be in memory after the first reference; the immediately following references will not fault.
- For example, consider the following sequence of addresses - 123,215,600,1234,76,96
- If page size is 100 then the reference string is 1,2,6,12,0,0

First In First Out (FIFO) algorithm

- Oldest page in main memory is the one which will be selected for replacement.
- Easy to implement, keep a list, replace pages from the tail and add new pages at the head.

Reference String : 0, 2, 1, 6, 4, 0, 1, 0, 3, 1, 2, 1

Misses : x x x x x x x x x



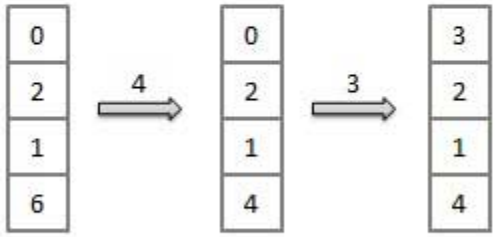
Fault Rate = 9 / 12 = 0.75

Optimal Page algorithm

- An optimal page-replacement algorithm has the lowest page-fault rate of all algorithms. An optimal page-replacement algorithm exists, and has been called OPT or MIN.
- Replace the page that will not be used for the longest period of time . Use the time when a page is to be used.

Reference String : 0, 2, 1, 6, 4, 0, 1, 0, 3, 1, 2, 1

Misses : x x x x x x



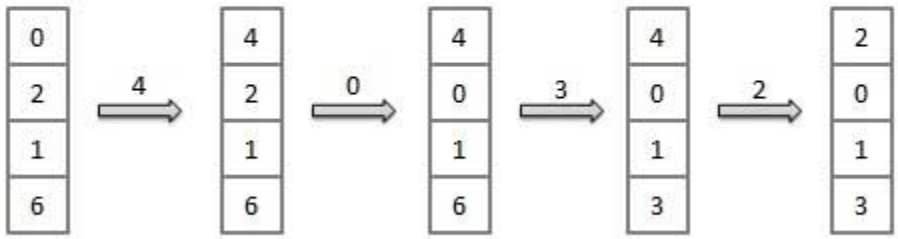
Fault Rate = 6 / 12 = 0.50

Least Recently Used (LRU) algorithm

- Page which has not been used for the longest time in main memory is the one which will be selected for replacement.
- Easy to implement, keep a list, replace pages by looking back into time.

Reference String : 0, 2, 1, 6, 4, 0, 1, 0, 3, 1, 2, 1

Misses : x x x x x x x x



Fault Rate = 8 / 12 = 0.67

Page buffering algorithm

- To get process start quickly, keep a pool of free frames.
- On page fault, select a page to be replaced.
- Write new page in the frame of free pool, mark the page table and restart the process.
- Now write the dirty page out of disk and place the frame holding replaced page in free pool.

Least frequently Used(LFU) algorithm

- Page with the smallest count is the one which will be selected for replacement.
- This algorithm suffers from the situation in which a page is used heavily during the initial phase of a process, but then is never used again.

Most frequently Used(LFU) algorithm

- This algorithm is based on the argument that the page with the smallest count was probably just brought in and has yet to be used.

Overview

Computers operate on many kinds of devices. General types include *storage devices* (disks, tapes),*transmission devices* (network cards, modems), and *human-interface devices* (screen, keyboard,

mouse). Other devices are more specialized. A device communicates with a computer system by sending signals over a cable or even through the air.

The device communicates with the machine via a connection point termed a *port* (for example, a serial port). If one or more devices use a common set of wires, the connection is called a *bus*.In other terms, a bus is a set of wires and a rigidly defined protocol that specifies a set of messages that can be sent on the wires.

Daisy chain

When device A has a cable that plugs into device B, and device B has a cable that plugs into device C, and device C plugs into a port on the computer, this arrangement is called a **daisy chain**. It usually operates as a bus.

Controller

A controller is a collection of electronics that can operate a port, a bus, or a device. A serial-port controller is an example of a simple device controller. This is a single chip in the computer that controls the signals on the wires of a serial port.

The SCSI bus controller is often implemented as a separate circuit board (a host adapter) that plugs into the computer. It contains a processor, microcode, and some private memory to enable it to process the SCSI protocol messages. Some devices have their own built-in controllers.

I/O port

An I/O port typically consists of four registers, called the **status** , **control**, **data-in**, and **data-out**registers.

S.N.	Register & Description
1	**Status Register** The status register contains bits that can be read by the host. These bits indicate states such as whether the current command has completed, whether a byte is available to be read from the data-in register, and whether there has been a device error.
2	**Control register** The control register can be written by the host to start a command or to change

	the mode of a device. For instance, a certain bit in the control register of a serial port chooses between full-duplex and half-duplex communication, another enables parity checking, a third bit sets the word length to 7 or 8 bits, and other bits select one of the speeds supported by the serial port.
3	**Data-in register** The data-in register is read by the host to get input.
4	**Data-out register** The data out register is written by the host to send output.

Polling

Polling is a process by which a host waits for controller response.It is a looping process, reading the status register over and over until the busy bit of status register becomes clear. The controller uses/sets the busy bit when it is busy working on a command, and clears the busy bit when it is ready to accept the next command. The host signals its wish via the command-ready bit in the command register. The host sets the command-ready bit when a command is available for the controller to execute.

In the following example, the host writes output through a port, coordinating with the controller by handshaking

- The host repeatedly reads the busy bit until that bit becomes clear.
- The host sets the write bit in the command register and writes a byte into the data-out register.
- The host sets the command-ready bit.
- When the controller notices that the command-ready bit is set, it sets the busy bit.
- The controller reads the command register and sees the write command.
- It reads the data-out register to get the byte, and does the I/O to the device.
- The controller clears the command-ready bit, clears the error bit in the status register to indicate that the device I/O succeeded, and clears the busy bit to indicate that it is finished.

I/O devices

I/O Devices can be categorized into following category.

S.N.	Category & Description
1	**Human readable** Human Readable devices are suitable for communicating with the computer user. Examples are printers, video display terminals, keyboard etc.
2	**Machine readable** Machine Readable devices are suitable for communicating with electronic equipment. Examples are disk and tape drives, sensors, controllers and actuators.
2	**Communication** Communication devices are suitable for communicating with remote devices. Examples are digital line drivers and modems.

Following are the differences between I/O Devices

S.N.	Criteria & Description
1	**Data rate** There may be differences of several orders of magnitude between the data transfer rates.
2	**Application** Different devices have different use in the system.
3	**Complexity of Control** A disk is much more complex whereas printer requires simple control interface.

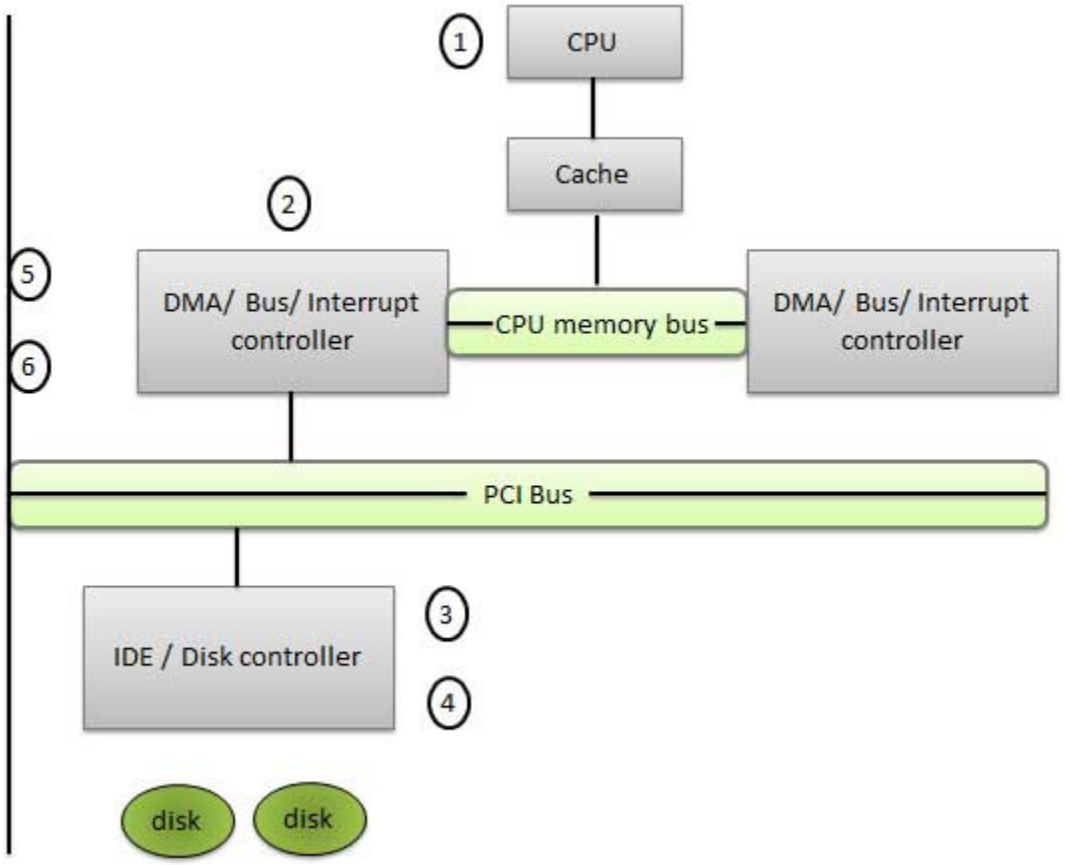
4	**Unit of transfer** Data may be transferred as a stream of bytes or characters or in larger blocks.
5	**Data representation** Different data encoding schemes are used for different devices.
6	**Error Conditions** The nature of errors differs widely from one device to another.

Direct Memory Access (DMA)

Many computers avoid burdening the main CPU with programmed I/O by offloading some of this work to a special purpose processor. This type of processor is called, a Direct Memory Access(DMA) controller. A special control unit is used to transfer block of data directly between an external device and the main memory, without intervention by the processor. This approach is called Direct Memory Access(DMA).

DMA can be used with either polling or interrupt software. DMA is particularly useful on devices like disks, where many bytes of information can be transferred in single I/O operations. When used with an interrupt, the CPU is notified only after the entire block of data has been transferred. For each byte or word transferred, it must provide the memory address and all the bus signals controlling the data transfer. Interaction with a device controller is managed through a device driver.

Handshaking is a process between the DMA controller and the device controller. It is performed via wires using terms DMA request and DMA acknowledge.



Step	Description
1	Device driver is instructed to transfer disk data to a buffer address X.
2	Device driver then instruct disk controller to transfer data to buffer.
3	Disk controller starts DMA transfer.
4	Disk controller sends each byte to DMA controller.
5	DMA controller transfers bytes to buffer, increases the memory address, decreases the counter C until C becomes zero.
6	When C becomes zero, DMA interrupts CPU to signal transfer completion.

Device Controllers

A computer system contains a many types of I/O devices and their respective controllers

- network card
- graphics adapter
- disk controller
- DVD-ROM controller
- serial port
- USB
- sound card

Interrupts

The CPU hardware uses an interrupt request line wire which helps CPU to sense after executing every instruction. When the CPU checks that a controller has put a signal on the interrupt request line, the CPU saves a state, such as the current value of the instruction pointer, and jumps to the interrupt handler routine at a fixed address. The interrupt handler part determines the cause of the interrupt, performs the necessary processing and executes a interrupt instruction to return the CPU to its execution state.

The basic mechanism of interrupt enables the CPU to respond to an asynchronous event, such as when a device controller becomes ready for service. Most CPUs have two interrupt request lines.

- **non-maskable interrupt** - Such kind of interrupts are reserved for events like unrecoverable memory errors.
- **maskable interrupt** - Such interrupts can be switched off by the CPU before the execution of critical instructions that must not be interrupted.

The interrupt mechanism accepts an address - a number that selects a specific interrupt handling routine/function from a small set.In most architectures, this address is an offset stored in a table called the interrupt vector table. This vector contains the memory addresses of specialized interrupt handlers.

Application I/O Interface

Application I/O Interface represents the structuring techniques and interfaces for the operating system to enable I/O devices to be treated in a standard, uniform way. The actual differences lies kernel level modules called device drivers which are custom tailored to corresponding devices but show one of the standard interfaces to applications. The purpose of the device-driver layer is to hide the differences among device controllers from the I/O subsystem of the kernel, such as the I/O system calls. Following are the characteristics of I/O interfaces with respected to devices.

- **Character-stream / block** - A character-stream device transfers bytes in one by one fashion, whereas a block device transfers a complete unit of bytes.
- **Sequential / random-access** - A sequential device transfers data in a fixed order determined by the device, random-access device can be instructed to seek position to any of the available data storage locations.
- **Synchronous / asynchronous** - A synchronous device performs data transfers with known response time where as an asynchronous device shows irregular or unpredictable response time.
- **Sharable / dedicated** - A sharable device can be used concurrently by several processes or threads but a dedicated device cannot be used.
- **Speed of operation** - Device speeds may range from a few bytes per second to a few gigabytes per second.
- **Read-write, read only, or write only** - Some devices perform both input and output, but others support only one data direction that is read only.

Clocks

Clocks are also called timers. The clock software takes the form of a device driver though a clock is neither a blocking device nor a character based device. The clock software is the clock driver. The exact function of the clock driver may vary depending on operating system. Generally, the functions of the clock driver include the following.

S.N.	Task	Description
1	Maintaining the time of the day	The clock driver implements the time of day or the real time clock function.It requires incrementing a counter at each clock tick.

2	Preventing processes from running too long	As a process is started, the scheduler initializes the quantum counter in clock ticks for the process. The clock driver decrements the quantum counter by 1, at every clock interrupt. When the counter gets to zero , clock driver calls the scheduler to set up another process. Thus clock driver helps in preventing processes from running longer than time slice allowed.
3	Accounting for CPU usage	Another function performed by clock driver is doing CPU accounting. CPU accounting implies telling how long the process has run.
4	Providing watchdog timers for parts of the system itself	Watchdog timers are the timers set by certain parts of the system. For example, to use a floppy disk, the system must turn on the motor and then wait about 500msec for it to comes up to speed.

Kernel I/O Subsystem

Kernel I/O Subsystem is responsible to provide many services related to I/O. Following are some of the services provided.

- **Scheduling** - Kernel schedules a set of I/O requests to determine a good order in which to execute them. When an application issues a blocking I/O system call, the request is placed on the queue for that device. The Kernel I/O scheduler rearranges the order of the queue to improve the overall system efficiency and the average response time experienced by the applications.
- **Buffering** - Kernel I/O Subsystem maintains a memory area known as buffer that stores data while they are transferred between two devices or between a device with an application operation. Buffering

is done to cope with a speed mismatch between the producer and consumer of a data stream or to adapt between devices that have different data transfer sizes.

- **Caching** - Kernel maintains cache memory which is region of fast memory that holds copies of data. Access to the cached copy is more efficient than access to the original.
- **Spooling and Device Reservation** A spool is a buffer that holds output for a device, such as a printer, that cannot accept interleaved data streams. The spooling system copies the queued spool files to the printer one at a time. In some operating systems, spooling is managed by a system daemon process. In other operating systems, it is handled by an in kernel thread.
- **Error Handling** An operating system that uses protected memory can guard against many kinds of hardware and application errors.

Device driver

Device driver is a program or routine developed for an I/O device. A device driver implements I/O operations or behaviors on a specific class of devices. For example a system supports one or a number of multiple brands of terminals, all slightly different terminals may have a single terminal driver. In the layered structure of I/O system, device driver lies between interrupt handler and device independent I/O software. The job of a device driver is following.

- To accept request from the device independent software above it.
- To see to it that the request is executed.

How a device driver handles a request is as follows: Suppose a request comes to read a block N. If the driver is idle at the time a request arrives, it starts carrying out the request immediately. Otherwise, if the driver is already busy with some other request, it places the new request in the queue of pending requests.

File

A file is a named collection of related information that is recorded on secondary storage such as magnetic disks, magnetic tapes and optical disks.In general, a file is a sequence of bits, bytes, lines or records whose meaning is defined by the files creator and user.

File Structure

File structure is a structure, which is according to a required format that operating system can understand.

- A file has a certain defined structure according to its type.
- A text file is a sequence of characters organized into lines.
- A source file is a sequence of procedures and functions.
- An object file is a sequence of bytes organized into blocks that are understandable by the machine.
- When operating system defines different file structures, it also contains the code to support these file structure. Unix, MS-DOS support minimum number of file structure.

File Type

File type refers to the ability of the operating system to distinguish different types of file such as text files source files and binary files etc. Many operating systems support many types of files. Operating system like MS-DOS and UNIX have the following types of files:

ORDINARY FILES

- These are the files that contain user information.
- These may have text, databases or executable program.
- The user can apply various operations on such files like add, modify, delete or even remove the entire file.

DIRECTORY FILES

- These files contain list of file names and other information related to these files.

SPECIAL FILES:

- These files are also known as device files.
- These files represent physical device like disks, terminals, printers, networks, tape drive etc.

These files are of two types

- **Character special files** - data is handled character by character as in case of terminals or printers.
- **Block special files** - data is handled in blocks as in the case of disks and tapes.

File Access Mechanisms

File access mechanism refers to the manner in which the records of a file may be accessed. There are several ways to access files

- Sequential access
- Direct/Random access
- Indexed sequential access

SEQUENTIAL ACCESS

A sequential access is that in which the records are accessed in some sequence i.e the information in the file is processed in order, one record after the other. This access method is the most primitive one. Example: Compilers usually access files in this fashion.

DIRECT/RANDOM ACCESS

- Random access file organization provides, accessing the records directly.
- Each record has its own address on the file with by the help of which it can be directly accessed for reading or writing.
- The records need not be in any sequence within the file and they need not be in adjacent locations on the storage medium.

INDEXED SEQUENTIAL ACCESS

- This mechanism is built up on base of sequential access.
- An index is created for each file which contains pointers to various blocks.
- Index is searched sequentially and its pointer is used to access the file directly.

Space Allocation

Files are allocated disk spaces by operating system. Operating systems deploy following three main ways to allocate disk space to files.

- Contiguous Allocation
- Linked Allocation
- Indexed Allocation

CONTIGUOUS ALLOCATION

- Each file occupy a contiguous address space on disk.
- Assigned disk address is in linear order.
- Easy to implement.
- External fragmentation is a major issue with this type of allocation technique.

LINKED ALLOCATION

- Each file carries a list of links to disk blocks.
- Directory contains link / pointer to first block of a file.
- No external fragmentation
- Effectively used in sequential access file.
- Inefficient in case of direct access file.

INDEXED ALLOCATION

- Provides solutions to problems of contigous and linked allocation.
- A index block is created having all pointers to files.
- Each file has its own index block which stores the addresses of disk space occupied by the file.
- Directory contains the addresses of index blocks of files.

Introduction:

Linux is a Unix-like and mostly POSIX-compliant computer operating system assembled under the model of free and open source software development and distribution. The defining component of Linux is the Linux kernel, an operating system kernel first released on 5 October 1991 by Linus Torvalds. The Free Software Foundation uses the name GNU/Linux, which has led to some controversy.

The **Linux Standard Base** (**LSB**) is a joint project by several Linux distributions under the organizational structure of the Linux Foundation to standardize the software system structure, including the file system hierarchy used in the GNU/Linux operating system. The LSB is based on the POSIX specification, the Single UNIX Specification, and several other open standards, but extends them in certain areas.

According to the LSB:

The goal of the LSB is to develop and promote a set of open standards that will increase compatibility among Linux distributions and enable software applications to run on any compliant system even in binary form. In addition, the LSB will help coordinate efforts to recruit software vendors to port and write products for Linux Operating Systems. The LSB compliance may be certified for a product by a certification procedure.

The LSB specifies for example: standard libraries, a number of commands and utilities that extend the POSIX standard, the layout of the file system hierarchy, run levels, the printing system, including spoolers such as CUPS and tools like Foomatic and several extensions to the X Window System.

The command **lsb_release -a** is available in many systems to get the LSB version details, or can be made available by installing lsb-release.

Linux was originally developed as a free operating system for Intel x86-based personal computers. It has since been ported to more computer hardware platforms than any other operating system. It is a leading operating system on servers and other big iron systems such as mainframe computers and supercomputers. As of June 2013, more than 95% of the world's 500 fastest supercomputers run some variant of Linux, including all the 44 fastest. Linux also runs on embedded systems, which are devices whose operating system is typically built into the firmware and is highly tailored to the system; this includes mobile phones, tablet computers, network routers, facility

automation controls, televisions and video game consoles. Android, which is a widely used operating system for mobile devices, is built on top of the Linux kernel.

The development of Linux is one of the most prominent examples of free and open source software collaboration. The underlying source code may be used, modified, and distributed—commercially or non-commercially—by anyone under licenses such as the GNU General Public License. Typically, Linux is packaged in a format known as a Linux distributionfor desktop and server use. Some popular mainstream Linux distributions include Debian, Ubuntu, Linux Mint, Fedora,Arch Linux, and the commercial Red Hat Enterprise Linux and SUSE Linux Enterprise Server. Linux distributions include the Linux kernel, supporting utilities and libraries and usually a large amount of application software to fulfill the distribution's intended use.

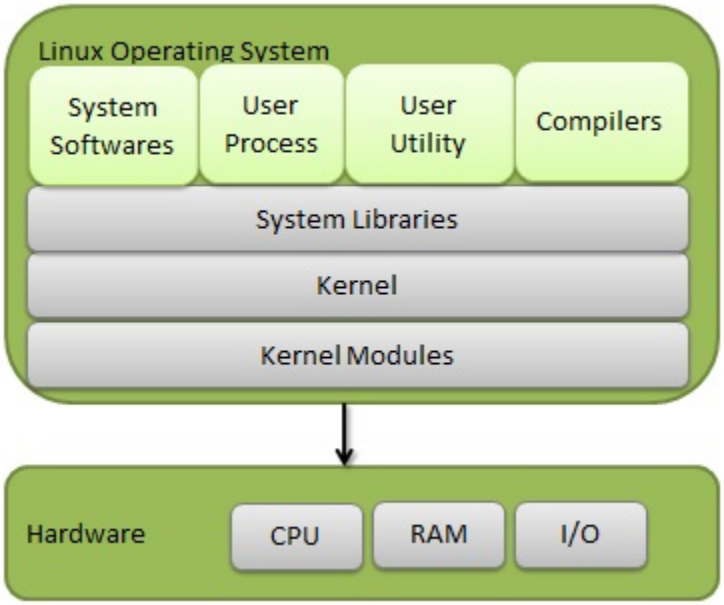
A distribution oriented toward desktop use will typically include X11 or Wayland as the windowing system, and an accompanying desktop environment such as GNOME or the KDE Software Compilation. Some such distributions may include a less resource intensive desktop such as LXDE or Xfce, for use on older or less powerful computers. A distribution intended to run as a server may omit all graphical environments from the standard install, and instead include other software to set up and operate a solution stack such as LAMP. Because Linux is freely redistributable, anyone may create a distribution for any intended use.

Linux is one of popular version of UNIX operating System. It is open source as its source code is freely available. It is free to use. Linux was designed considering UNIX compatibility. It's functionality list is quite similar to that of UNIX.

Components of Linux System

Linux Operating System has primarily three components

- **Kernel** - Kernel is the core part of Linux. It is responsible for all major activities of this operating system. It is consists of various modules and it interacts directly with the underlying hardware. Kernel provides the required abstraction to hide low level hardware details to system or application programs.
- **System Library** - System libraries are special functions or programs using which application programs or system utilities accesses Kernel's features. These libraries implements most of the functionalities of the operating system and do not requires kernel module's code access rights.
- **System Utility** - System Utility programs are responsible to do specialized, individual level tasks.



Kernel Mode vs User Mode

Kernel component code executes in a special privileged mode called **kernel mode** with full access to all resources of the computer. This code represents a single process, executes in single address space and do not require any context switch and hence is very efficient and fast. Kernel runs each processes and provides system services to processes, provides protected access to hardware's to processes.

Support code which is not required to run in kernel mode is in System Library. User programs and other system programs works in **User Mode** which has no access to system hardware's and kernel code. User programs/ utilities use System libraries to access Kernel functions to get system's low level tasks.

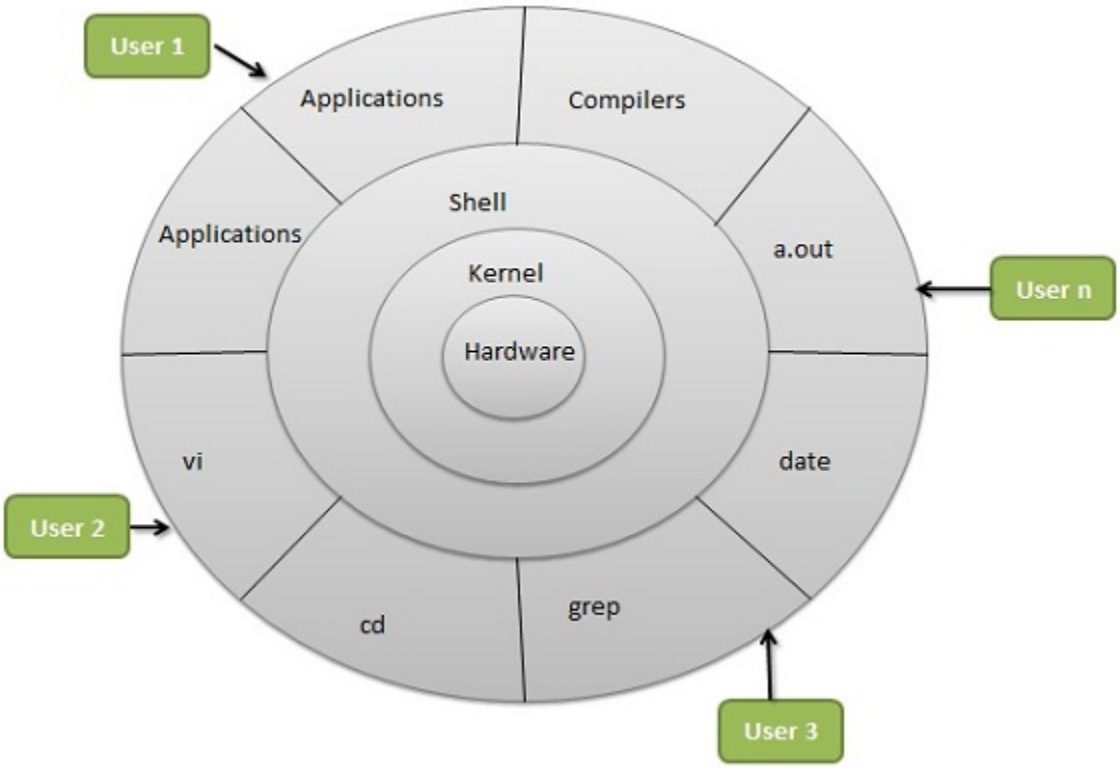
Basic Features

Following are some of the important features of Linux Operating System.

- **Portable -** Portability means software's can works on different types of hardware's in same way.Linux kernel and application programs supports their installation on any kind of hardware platform.
- **Open Source -** Linux source code is freely available and it is community based development project. Multiple teams works in collaboration to enhance the capability of Linux operating system and it is continuously evolving.
- **Multi-User -** Linux is a multiuser system means multiple users can access system resources like memory/ ram/ application programs at same time.

- **Multiprogramming** - Linux is a multiprogramming system means multiple applications can run at same time.
- **Hierarchical File System** - Linux provides a standard file structure in which system files/ user files are arranged.
- **Shell** - Linux provides a special interpreter program which can be used to execute commands of the operating system. It can be used to do various types of operations, call application programs etc.
- **Security** - Linux provides user security using authentication features like password protection/ controlled access to specific files/ encryption of data.

Architecture



Linux System Architecture is consists of following layers:

- **Hardware layer** - Hardware consists of all peripheral devices (RAM/ HDD/ CPU etc).
- **Kernel** - Core component of Operating System, interacts directly with hardware, provides low level services to upper layer components.
- **Shell** - An interface to kernel, hiding complexity of kernel's functions from users. Takes commands from user and executes kernel's functions.
- **Utilities** - Utility programs giving user most of the functionalities of an operating systems.

Linux Vs Windows

Linux is an open-source Operating System. People can change codes and add programs to Linux OS which will help use your computer better. Linux evolved as a reaction to the monopoly position of windows. you can't change any code for windows OS. You can't even see which processes do what and build your onw extension. Linux wants the programmers to extend and redesign it's OS. Linux user's can edit its OS and design new OS.

All flavors of Windows come from Microsoft. Linux come from different companies like LIndows , Lycoris, Red Hat, SuSe, Mandrake, Knopping, Slackware.

Linux is customizable but Windows is not. For example,NASlite is a version of Linux that runs off a single floppy disk and converts an old computer into a file server. This ultra small edition of Linux is capable of networking, file sharing and being a web server.

Linux is freely available for desktop or home use but Windows is expensive. For server use, Linux is cheap compared to Windows. Microsoft allows a single copy of Windows to be used on one computer. You can run Linux on any number of computers.

Linux has high security. You have to log on to Linux with a userid and password. You can login as root or as normal user. The root has full previlage.

Linux has a reputation for fewer bugs than Windows.

Windows must boot from a primary partition. Linux can boot from either a primary partition or a logical partition inside an extended partition. Windows must boot from the first hard disk. Linux can boot from any hard disk in the computer.

Windows uses a hidden file for its swap file. Typically this file resides in the same partition as the OS (advanced users can opt to put the file in another partition). Linux uses a dedicated partition for its swap file.

Windows separates directories with a back slash while Linux uses a normal forward slash.

Windows file names are not case sensitive. Linux file names are. For example "abc" and "aBC" are different files in Linux, whereas in Windows it would refer to the same file.

Windows and Linux have different concepts for their file hierarchy. Windows uses a volume-based file hierarchy while Linux uses a unified scheme. Windows uses letters of the alphabet to represent different devices and different hard disk partitions. eg: c: , d: , e: etc.. while in linux " / " is the main directory.

Linux and windows support the concept of hidden files. In linux hidden files begin with " . ", eg: .filename

In Linux each user will have a home directory and all his files will be save under it while in windows the user saves his files anywhere in the drive. This makes difficult to have backup for his contents. In Linux its easy to have backup's.

List of all the Directories in Linux

```
/                   ---------> Root or the main directory
|---root            ---------> Home directory for the root user
|---home            ---------> User's home directories
|    |----ftp       ---------> Users services are listed here
|    |----httpd
|    |----samba
|    |----user1
|    |----user2
|---bin             ---------> Commands needed during bootup and also needed by normal users
|---sbin            ---------> Like bin but commands are not intended for normal users.
|---proc            ---------> This filesystem is not on a disk.  virtual directory.
|    |
|    |----n         ---------> A directory with info about process number n.
|
|---usr             ---------> Contains all commands, libraries, man pages etc  for normal operation.
|    |
|    |----bin       --------->   Almost all user commands.
|    |----sbin      --------->   System admin commands not needed in the root filesystem
|    |
|    |----include   ---------> Header files for the C programming language.
|    |
|    |----lib       --------->   Unchanging data files for programs and subsystems
|    |----local     --------->   Locally installed software and other files.
|    |----man       -------->   Help pages
|    |----info      --------->    Info documents
```

```
|   |----doc          ---------->  Documentation for various packages
|   |----tmp
|   |----X11R6      ---------> The X windows system files.
|   |
|   |----X386          --------->     Like X11R6 but for X11 release 5
|---boot        ---------> Files used by the bootstrap loader, LILO.  Kernel images stored.
|---lib          --------->    Shared libraries needed by the programs on the root filesystem
|   |----modules    --------->    Loadable kernel modules.
|
|---dev        --------->     Device files for devices such as disk drives, serial ports, etc.
|---etc        --------->  Configuration files specific to the machine.
|   |----skel     --------->    For home directory creation.
|   |----sysconfig    --------->   Files that configure the linux system for networking etc.
|---var     --------->  Contains files that change for mail, news, temp files, etc
|   |----file
|   |----lib        --------->  Files that change while the system is running normally
|   |----local      ---------> Variable data for programs installed in /usr/local.
|   |----lock      --------->  Lock files.  Used by a program.
|   |----log       --------->  Log files from programs such as login and syslog etc.
|   |
|   |----run      ---------> Files that contain information about the system.
|   |
|   |----spool   --------->   Directories for mail, printer spools, news and other spooled work.
|   |----tmp      --------->   Temporary files that are needed to exist for longer time.
|   |
|---mnt    --------->  Mount points for temporary mounts by the system administrator.
|---tmp    --------->  Temporary files.  Programs running after bootup should use /var/tmp.
```

Programming on Linux

Most Linux distributions support dozens of programming languages. The original development tools used for building both Linux applications and operating system programs are found within the GNU toolchain, which includes the GNU Compiler Collection (GCC) and the GNU build

system. Amongst others, GCC provides compilers for Ada, C, C++, Java, and Fortran. First released in 2003, the LLVM project provides an alternative open-source compiler for many languages. Proprietary compilers for Linux include the Intel C++ Compiler, Sun Studio, and IBM XL C/C++ Compiler. BASIC in the form of Visual Basic is supported in such forms as Gambas, FreeBASIC, and XBasic, and in terms of terminal programming or QuickBASIC or Turbo BASIC programming in the form of QB64.

A common feature of Unix-like systems, Linux includes traditional specific-purpose programming languages targeted at scripting, text processing and system configuration and management in general. Linux distributions support shell scripts, awk, sed and make. Many programs also have an embedded programming language to support configuring or programming themselves. For example, regular expressions are supported in programs like grep, or locate, while advanced text editors, like GNU Emacs have a complete Lisp interpreter built-in.

Most distributions also include support for PHP, Perl, Ruby, Python and other dynamic languages. While not as common, Linux also supports C# (via Mono), Vala, and Scheme. A number of Java Virtual Machines and development kits run on Linux, including the original Sun Microsystems JVM (HotSpot), and IBM's J2SE RE, as well as many open-source projects like Kaffe and JikesRVM.

GNOME and KDE are popular desktop environments and provide a framework for developing applications. These projects are based on the GTK+ and Qt widget toolkits, respectively, which can also be used independently of the larger framework. Both support a wide variety of languages. There are a number of Integrated development environments available including Anjuta, Code::Blocks, CodeLite, Eclipse, Geany, ActiveState Komodo, KDevelop, Lazarus, MonoDevelop, NetBeans, and Qt Creator, while the long-established editors Vim, nano and Emacs remain popular.

Linux Basic Commands

Linux operating system has a beautiful graphical interface which most of us will be using. It will be good to learn the basic commands in Linux to work interactively with the Linux operating system. Linux has a back end access know as shell. You can control and activate all the process in Linux from the shell. So it is very important to learn few basic commands to work with Linux operating system.

First we will learn how to login for shell access. There are 7 terminals for Linux. 6 terminals are non -

GUI and 1 terminal for GUI access. You can login to each terminal using Alt + Ctrl + F1 , F2, .. F7. Each terminals will request your username and password for login. If you want to use the shell in the graphical interface (GUI), press Alt + F2 and type "konsole". As a user you will have permission to access only your /home/user directory and other directories in it.
*Note: user denotes the username.

Navigation Commands - [pwd, cd]

---

pwd
This command is used to find the current location or current working directory.
Eg:
[user@ws26 ~]$ pwd
/home/user
cd
This command id used to change the directory. You can move from one directory to another using this command. Few examples are given below.
Concider you have a directory structure /home/user/test/test1/ . test and test1 are directories in user home.
Example 1:
Consider you have a directory "test" in /home/user. Your current working directory is /home/user. You want to change your current working directory from /home/user to /home/user/test, use the following command
[user@ws26 ~]$ pwd
/home/user
[user@ws26 ~]$ cd test
[user@ws26 ~]$ pwd
/home/user/test
Example 2:
If you want to move back to /home/user, use the following command
[user@ws26 ~]$ cd ..
[user@ws26 ~]$ pwd
/home/user
Example 3:

To get back to the home directory of the user

[user@ws26 ~]$ cd ~

[user@ws26 ~]$ pwd

/home/user.

Listing contents in a directory - [ls, ls -l, ll]

---

ls

This command is used to list all the files and directory in the current directory.

Eg:

[user@ws26 ~]$ ls

1152696870.jpg book OperaDownloads snapshot52.png

image300.jpg Desktop test spiderman2.jpg

ls -l or ll

This command is also used to list all the files and directories. Here you will get more details about the files and directories present in the current directory. You will see the permission set, creation date, file / directory size etc.

Eg:

[user@ws26 ~]$ ls -l

total 5

-rw-rw-r-- 1 user user 33188 Dec 22 02:56 1152696870.jpg

-rw-rw-r-- 1 user user 17647 Aug 19 2006 534458.gif

-rw-rw-r-- 1 user user 26817 Aug 19 2006 534477.gif

drwxrwxr-x 2 user user 4096 Jan 13 04:20 book

drwx------ 3 user user 4096 Feb 4 02:34 Desktop

Reading files in Linux - [cat, more, less]

---

cat

This command is used to display the contents of a file. You can read the file contents using the cat command.

Example:

[user@ws26 ~]$ cat testfile

this is a test file.. You can read me..

Less and More

Both the less and more commands serve similar function. They are used to display file one screen at a time. You can press spacebar to continue reading the file. These commands are mainly used while reading large files.

Example 1:

[user@ws26 ~]$ less testfile

this is a test file.. You can read me..

Example 2:

[user@ws26 ~]$ more testfile

this is a test file.. You can read me..

Manipulating files - [cp, mv, rm, mkdir]

---

cp

This command is used to copy files/directory.

Example:

[user@ws26 ~]$ cp file1 directory1

This will copy the file1 into the directory1

cp -R

This command is used to copy directories recursively (copy all the files and folders inside the directory).

Example:

[user@ws26 ~]$ cp directory2 directory1

This will copy the whole directory2 into directory1

mv

This command is used to move the file or directory.

Example:

[user@ws26 ~]$ mv directory2 directory1

This will move the whole directory2 into directory1

rm

This command is used to remove or delete files and directories.

Example:

[user@ws26 ~]$ rm directory2

This will delete the directory - directory2

mkdir

This command is used to create a new directory

Example:

[user@ws26 ~]$ mkdir directory3

This will create a new directory - directory3.

Create new file in Linux - vi editor

The Vi editor is mainly used to create new files in Linux.

Example:

vi testfile

The command will open a new window and you can insert text to it. To begin press "i". After adding your contect press ":wq" to save the file.

Basics of linux:

1.1.2. Linus and Linux

By the beginning of the 90s home PCs were finally powerful enough to run a full blown UNIX. Linus Torvalds, a young man studying computer science at the university of Helsinki, thought it would be a good idea to have some sort of freely available academic version of UNIX, and promptly started to code.

He started to ask questions, looking for answers and solutions that would help him get UNIX on his PC. Below is one of his first posts in comp.os.minix, dating from 1991:

From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)

Newsgroups: comp.os.minix

Subject: Gcc-1.40 and a posix-question

Message-ID: <1991Jul3.100050.9886@klaava.Helsinki.FI>

Date: 3 Jul 91 10:00:50 GMT

Hello netlanders,

Due to a project I'm working on (in minix), I'm interested in the posix

standard definition. Could somebody please point me to a (preferably)
machine-readable format of the latest posix rules? Ftp-sites would be
nice.

From the start, it was Linus' goal to have a free system that was completely compliant with the original UNIX. That is why he asked for POSIX standards, POSIX still being the standard for UNIX.

In those days plug-and-play wasn't invented yet, but so many people were interested in having a UNIX system of their own, that this was only a small obstacle. New drivers became available for all kinds of new hardware, at a continuously rising speed. Almost as soon as a new piece of hardware became available, someone bought it and submitted it to the Linux test, as the system was gradually being called, releasing more free code for an ever wider range of hardware. These coders didn't stop at their PC's; every piece of hardware they could find was useful for Linux.

Back then, those people were called "nerds" or "freaks", but it didn't matter to them, as long as the supported hardware list grew longer and longer. Thanks to these people, Linux is now not only ideal to run on new PC's, but is also the system of choice for old and exotic hardware that would be useless if Linux didn't exist.

Two years after Linus' post, there were 12000 Linux users. The project, popular with hobbyists, grew steadily, all the while staying within the bounds of the POSIX standard. All the features of UNIX were added over the next couple of years, resulting in the mature operating system Linux has become today. Linux is a full UNIX clone, fit for use on workstations as well as on middle-range and high-end servers. Today, a lot of the important players on the hard- and software market each have their team of Linux developers; at your local dealer's you can even buy pre-installed Linux systems with official support - eventhough there is still a lot of hard- and software that is not supported, too.

1.1.3. Current application of Linux systems

Today Linux has joined the desktop market. Linux developers concentrated on networking and services in the beginning, and office applications have been the last barrier to be taken down. We don't like to admit that Microsoft is ruling this market, so plenty of alternatives have been started over the last couple of years to make Linux an acceptable choice as a workstation, providing an easy user interface and MS compatible office applications like word processors, spreadsheets, presentations and the like.

On the server side, Linux is well-known as a stable and reliable platform, providing database and trading services for companies like Amazon, the well-known online bookshop, US Post Office, the German army and many others. Especially Internet providers and Internet service providers have grown fond of Linux as firewall, proxy- and web server, and you will find a Linux box within reach of every UNIX system administrator who appreciates a comfortable management station. Clusters of Linux machines are used in the creation of movies such as "Titanic", "Shrek" and others. In post offices, they are the nerve centers that route mail and in large search engine, clusters are used to perform internet searches.These are only a few of the thousands of heavy-duty jobs that Linux is performing day-to-day across the world.

It is also worth to note that modern Linux not only runs on workstations, mid- and high-end servers, but also on "gadgets" like PDA's, mobiles, a shipload of embedded applications and even on experimental wristwatches. This makes Linux the only operating system in the world covering such a wide range of hardware.

1.1.2. Linus and Linux

By the beginning of the 90s home PCs were finally powerful enough to run a full blown UNIX. Linus Torvalds, a young man studying computer science at the university of Helsinki, thought it would be a good idea to have some sort of freely available academic version of UNIX, and promptly started to code.

He started to ask questions, looking for answers and solutions that would help him get UNIX on his PC. Below is one of his first posts in comp.os.minix, dating from 1991:

```
From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
Newsgroups: comp.os.minix
Subject: Gcc-1.40 and a posix-question
Message-ID: <1991Jul3.100050.9886@klaava.Helsinki.FI>
Date: 3 Jul 91 10:00:50 GMT
Hello netlanders,
Due to a project I'm working on (in minix), I'm interested in the posix
standard definition. Could somebody please point me to a (preferably)
machine-readable format of the latest posix rules? Ftp-sites would be
```

nice.

From the start, it was Linus' goal to have a free system that was completely compliant with the original UNIX. That is why he asked for POSIX standards, POSIX still being the standard for UNIX.

In those days plug-and-play wasn't invented yet, but so many people were interested in having a UNIX system of their own, that this was only a small obstacle. New drivers became available for all kinds of new hardware, at a continuously rising speed. Almost as soon as a new piece of hardware became available, someone bought it and submitted it to the Linux test, as the system was gradually being called, releasing more free code for an ever wider range of hardware. These coders didn't stop at their PC's; every piece of hardware they could find was useful for Linux.

Back then, those people were called "nerds" or "freaks", but it didn't matter to them, as long as the supported hardware list grew longer and longer. Thanks to these people, Linux is now not only ideal to run on new PC's, but is also the system of choice for old and exotic hardware that would be useless if Linux didn't exist.

Two years after Linus' post, there were 12000 Linux users. The project, popular with hobbyists, grew steadily, all the while staying within the bounds of the POSIX standard. All the features of UNIX were added over the next couple of years, resulting in the mature operating system Linux has become today. Linux is a full UNIX clone, fit for use on workstations as well as on middle-range and high-end servers. Today, a lot of the important players on the hard- and software market each have their team of Linux developers; at your local dealer's you can even buy pre-installed Linux systems with official support - eventhough there is still a lot of hard- and software that is not supported, too.

1.1.3. Current application of Linux systems

Today Linux has joined the desktop market. Linux developers concentrated on networking and services in the beginning, and office applications have been the last barrier to be taken down. We don't like to admit that Microsoft is ruling this market, so plenty of alternatives have been started over the last couple of years to make Linux an acceptable choice as a workstation, providing an easy user interface and MS compatible office applications like word processors, spreadsheets, presentations and the like.

On the server side, Linux is well-known as a stable and reliable platform, providing database and trading services for companies like Amazon, the well-known online bookshop, US Post Office, the

German army and many others. Especially Internet providers and Internet service providers have grown fond of Linux as firewall, proxy- and web server, and you will find a Linux box within reach of every UNIX system administrator who appreciates a comfortable management station. Clusters of Linux machines are used in the creation of movies such as "Titanic", "Shrek" and others. In post offices, they are the nerve centers that route mail and in large search engine, clusters are used to perform internet searches.These are only a few of the thousands of heavy-duty jobs that Linux is performing day-to-day across the world.

It is also worth to note that modern Linux not only runs on workstations, mid- and high-end servers, but also on "gadgets" like PDA's, mobiles, a shipload of embedded applications and even on experimental wristwatches. This makes Linux the only operating system in the world covering such a wide range of hardware.

1.4. Properties of Linux

1.4.1. Linux Pros

A lot of the advantages of Linux are a consequence of Linux' origins, deeply rooted in UNIX, except for the first advantage, of course:

- Linux is free:

  As in free beer, they say. If you want to spend absolutely nothing, you don't even have to pay the price of a CD. Linux can be downloaded in its entirety from the Internet completely for free. No registration fees, no costs per user, free updates, and freely available source code in case you want to change the behavior of your system.

  Most of all, Linux is free as in free speech:

  The license commonly used is the GNU Public License (GPL). The license says that anybody who may want to do so, has the right to change Linux and eventually to redistribute a changed version, on the one condition that the code is still available after redistribution. In practice, you are free to grab a kernel image, for instance to add support for teletransportation machines or time travel and sell your new code, as long as your customers can still have a copy of that code.

- Linux is portable to any hardware platform:

  A vendor who wants to sell a new type of computer and who doesn't know what kind of OS his new machine will run (say the CPU in your car or washing machine), can take a Linux kernel and make it work on his hardware, because documentation related to this activity is freely available.

- Linux was made to keep on running:

  As with UNIX, a Linux system expects to run without rebooting all the time. That is why a lot of tasks are being executed at night or scheduled automatically for other calm moments, resulting in higher availability during busier periods and a more balanced use of the hardware. This property allows for Linux to be applicable also in environments where people don't have the time or the possibility to control their systems night and day.

- Linux is secure and versatile:

  The security model used in Linux is based on the UNIX idea of security, which is known to be robust and of proven quality. But Linux is not only fit for use as a fort against enemy attacks from the Internet: it will adapt equally to other situations, utilizing the same high standards for security. Your development machine or control station will be as secure as your firewall.

- Linux is scalable:

  From a Palmtop with 2 MB of memory to a petabyte storage cluster with hundreds of nodes: add or remove the appropriate packages and Linux fits all. You don't need a supercomputer anymore, because you can use Linux to do big things using the building blocks provided with the system. If you want to do little things, such as making an operating system for an embedded processor or just recycling your old 486, Linux will do that as well.

- The Linux OS and most Linux applications have very short debug-times:

  Because Linux has been developed and tested by thousands of people, both errors and people to fix them are usually found rather quickly. It sometimes happens that there are only a couple of hours between discovery and fixing of a bug.

1.4.2. Linux Cons

- There are far too many different distributions:

  "Quot capites, tot rationes", as the Romans already said: the more people, the more opinions. At first glance, the amount of Linux distributions can be frightening, or ridiculous, depending on your point of view. But it also means that everyone will find what he or she needs. You don't need to be an expert to find a suitable release.

  When asked, generally every Linux user will say that the best distribution is the specific version he is using. So which one should you choose? Don't worry too much about that: all releases contain more or less the same set of basic packages. On top of the basics, special third party software is added making, for example, TurboLinux more suitable for the small and medium enterprise, RedHat for servers and SuSE for workstations. However, the differences are likely to be very superficial. The best strategy is to test a couple of distributions; unfortunately not everybody has the time for this. Luckily, there is plenty of advice on the subject of choosing your Linux. A quick search on Google, using the keywords "choosing your distribution" brings up tens of links to good advise. The Installation HOWTO also discusses choosing your distribution.

- Linux is not very user friendly and confusing for beginners:

  It must be said that Linux, at least the core system, is less userfriendly to use than MS Windows and certainly more difficult than MacOS, but... In light of its popularity, considerable effort has been made to make Linux even easier to use, especially for new users. More information is being released daily, such as this guide, to help fill the gap for documentation available to users at all levels.

- Is an Open Source product trustworthy?

  How can something that is free also be reliable? Linux users have the choice whether to use Linux or not, which gives them an enormous advantage compared to users of proprietary software, who don't have that kind of freedom. After long periods of testing, most Linux users come to the conclusion that Linux is not only as good, but in many cases better and faster that the traditional solutions. If Linux were not trustworthy, it would have been long gone, never

knowing the popularity it has now, with millions of users. Now users can influence their systems and share their remarks with the community, so the system gets better and better every day. It is a project that is never finished, that is true, but in an ever changing environment, Linux is also a project that continues to strive for perfection.

1.3. Does Linux have a future?

1.3.1. Open Source

The idea behind Open Source software is rather simple: when programmers can read, distribute and change code, the code will mature. People can adapt it, fix it, debug it, and they can do it at a speed that dwarfs the performance of software developers at conventional companies. This software will be more flexible and of a better quality than software that has been developed using the conventional channels, because more people have tested it in more different conditions than the closed software developer ever can.

The Open Source initiative started to make this clear to the commercial world, and very slowly, commercial vendors are starting to see the point. While lots of academics and technical people have already been convinced for 20 years now that this is the way to go, commercial vendors needed applications like the Internet to make them realize they can profit from Open Source. Now Linux has grown past the stage where it was almost exclusively an academic system, useful only to a handful of people with a technical background. Now Linux provides more than the operating system: there is an entire infrastructure supporting the chain of effort of creating an operating system, of making and testing programs for it, of bringing everything to the users, of supplying maintenance, updates and support and customizations, etcetera. Today, Linux is ready to accept the challenge of a fast-changing world.

1.3.2. Ten years of experience at your service

While Linux is probably the most well-known Open Source initiative, there is another project that contributed enormously to the popularity of the Linux operating system. This project is called SAMBA, and its achievement is the reverse engineering of the Server Message Block (SMB)/Common Internet File System (CIFS) protocol used for file- and print-serving on PC-related machines, natively supported by MS Windows NT and OS/2, and Linux. Packages are now available

for almost every system and provide interconnection solutions in mixed environments using MS Windows protocols: Windows-compatible (up to and includingWinXP) file- and print-servers.

Maybe even more successful than the SAMBA project is the Apache HTTP server project. The server runs on UNIX, Windows NT and many other operating systems. Originally known as "A PAtCHy server", based on existing code and a series of "patch files", the name for the matured code deserves to be connoted with the native American tribe of the Apache, well-known for their superior skills in warfare strategy and inexhaustible endurance. Apache has been shown to be substantially faster, more stable and more feature-full than many other web servers. Apache is run on sites that get millions of visitors per day, and while no official support is provided by the developers, the Apache user community provides answers to all your questions. Commercial support is now being provided by a number of third parties.

In the category of office applications, a choice of MS Office suite clones is available, ranging from partial to full implementations of the applications available on MS Windows workstations. These initiatives helped a great deal to make Linux acceptable for the desktop market, because the users don't need extra training to learn how to work with new systems. With the desktop comes the praise of the common users, and not only their praise, but also their specific requirements, which are growing more intricate and demanding by the day.

The Open Source community, consisting largely of people who have been contributing for over half a decade, assures Linux' position as an important player on the desktop market as well as in general IT application. Paid employees and volunteers alike are working diligently so that Linux can maintain a position in the market. The more users, the more questions The Open Source community makes sure answers keep coming, and watches the quality of the answers with a suspicious eye, resulting in ever more stability and accessibility.

Listing all the available Linux software is beyond the scope of this guide, as there are tens of thousands of packages. Throughout this course we will present you with the most common packages, which are almost all freely available. In order to take away some of the fear of the beginning user, here's a screenshot of one of your most-wanted programs. You can see for yourself that no effort has been spared to make users who are switching from Windows feel at home:

BASIC COMMANDS OF LINUX

Below is a listing of each of the Unix and Linux commands currently listed on Computer Hope and a brief explanation of what each of the commands do. This is a full listing which means not all the below commands will work with your distribution and may also not work because of your privileges. Clicking on any of the commands will display additional help and information about that command.

Command	Description
a2p	Creates a Perl script from an awk script.
ac	Prints statistics about user connection time.
access	A system function which checks real user's permissions to access a file.
addgroup	Adds a new group to the system.
adduser	Adds a new user to the system.
agrep	Version of the grep utility which also matches approximate patterns.
alias	Creates another name for a command or command string.
apropos	Searches the manual pages for a keyword or regular expression.
apt-cache	Queries the APT software package cache.

apt-get	Command line tool for working with APT software packages.
aptitude	Text-based front-end for the APT package management system.
ar	Creates, modifies, and extracts from archives.
arch	Displays the architecture of the current host.
arp	Manipulate the system ARP cache.
as	An assembler.
aspell	Interactive spell checker.
at	Command scheduler.
awk	Awk script processing program.
basename	Deletes any specified prefix from a string.
bash	Command Bourne interpreter
bc	Calculator.
bdiff	Compare large files.
bfs	Editor for large files.
bg	Continues a program running in the background.

biff	Enable and disable incoming mail notifications.
break	Break out of while, for, foreach, or until loop.
bs	Battleship game.
bye	Alias often used for the exit command.
cal	Calendar.
calendar	Display appointments and reminders.
cancel	Cancels a print job.
cat	View or modify a file.
cc	C compiler.
cd	Change directory.
cfdisk	A more user-friendly version of the fdisk disk partitioning utility.
chdir	Change directory.
checkeq	Language processors to assist in describing equations.
checknr	Check nroff and troff files for any errors.

chfn	Modify your own information or if superuser or root modify another users information.
chgrp	Change a groups access to a file or directory.
chkey	Change the secure RPC key pair.
chmod	Change the permission of a file.
chown	Change the ownership of a file.
chroot	Run a command or shell from another directory, and treats that directory as root.
chsh	Change login shell.
cksum	Display and calculate a CRC for files.
clear	Clears screen.
cmp	Compare files.
col	Reverse line-feeds filter.
comm	Compare files and select or reject lines that are common.
compress	Compress files on a computer.
continue	Break out of while, for, foreach, or until loop.

cp	Copy files.
cpio	Creates archived CPIO files.
crontab	Create and list files that you want to run on a regular schedule.
crypt	Function used to encrypt passwords.
csh	Execute the C shell command interpreter
csplit	Split files based on context.
ctags	Create a tag file for use with ex and vi.
cu	Calls or connects to another Unix system, terminal or non-Unix system.
curl	Transfer a URL.
cut	Cut out selected fields of each line of a file.
date	Tells you the date and time in Unix.
dc	An arbitrary precision arithmetic package.
dd	Convert and copy a file.
delgroup	Remove a group from the system.
deluser	Remove a user from the system.

depmod	Generates a list of kernel module dependences, modules.dep, and associated map files.
deroff	Removes nroff/troff, tbl, and eqn constructs.
df	Display the available disk space for each mount.
dhclient	Dynamic Host Configuration Protocol Client.
diff	Displays two files and prints the lines that are different.
dig	DNS lookup utility.
dircmp	Lists the different files when comparing directories.
dirname	Deliver portions of path names.
dmesg	Print or control the kernel ring buffer.
dos2unix	Converts text files between DOS and Unix formats.
dpkg	Queries, installs, removes, and maintains Debian software packages and their dependencies.
dpost	Translates files created by troff into PostScript.
du	Tells you how much space a file occupies.
echo	Displays text after echo to the terminal.

ed	Line oriented file editor.
edit	Text editor.
egrep	Search a file for a pattern using full regular expressions.
eject	Ejects removable media.
elm	Program command used to send and receive e-mail.
emacs	Text editor.
enable	Enables and disables LP printers.
env	Displays environment variables.
eqn	Language processors to assist in describing equations.
ex	Line-editor mode of the vi text editor.
exit	Exit from a program, shell or log you out of a Unix network.
expand	Expand copies of files.
expr	Evaluate arguments as an expression.
fc	Lists, edits, or re-executes commands previously entered.

fdisk	A disk partioning utility.
fg	Continues a stopped job by running it in the foreground
fgrep	Search a file for a fixed-character string.
file	Tells you if the object you are looking at is a file or a directory.
find	Finds files within a directory hierarchy.
findsmb	List info about machines that respond to SMB name queries on a subnet.
finger	Lists information about the user.
fmt	Simple text formatters.
fold	Filter for folding lines.
for	Shell built-in functions to repeatedly execute action(s) for a selected number of times.
foreach	Shell built-in functions to repeatedly execute action(s) for a selected number of times.
free	Display amount of free and used memory in the system
fsck	Check and repair a Linux file system.
ftp	Enables ftp access to another terminal.

fuser	Identify processes using files or sockets.
gawk	Powerful pattern-matching and processing language.
getfacl	Display discretionary file information.
gethostname	System call to get the hostname of the current processor.
gpasswd	Administer /etc/group and /etc/gshadow.
gprof	The gprof utility produces an execution profile of a program.
grep	Finds text within a file.
groupadd	Creates a new group account.
groupdel	Enables a superuser or root to remove a group.
groupmod	Enables a superuser or root to modify a group.
gunzip	Expand compressed files.
gview	A programmer's text editor.
gvim	A programmer's text editor.
gzip	Compress files.
halt	Stop the computer.

hash	Remove internal hash tables.
hashstat	Evaluates the effectiveness of internal hash tables.
head	Displays the first ten lines of a file, unless otherwise stated.
help	Displays help for built-in shell commands.
history	Display the command history.
host	DNS lookup utility.
hostid	Prints the numeric identifier for the current host.
hostname	Set or print name of current host system.
id	Shows you the numeric user and group ID on BSD.
ifconfig	Sets up network interfaces.
ifdown	Take a network interface down.
ifup	Bring a network interface up.
info	Read Info documents.
init	Process control initialization.

iostat	Reports Central Processing Unit (CPU) statistics and input/output statistics for devices and partitions.
ip	Show and manipulate routing, devices, policy routing and tunnels.
isalist	Display the native instruction sets executable on this platform.
iwconfig	Configure a wireless network interface.
jobs	List the jobs currently running in the background.
join	Joins command forms together.
keylogin	Decrypt the user's secret key.
kill	Cancels a job.
killall	Kills processes by name.
ksh	Korn shell command interpreter.
last	Displays a listing of the most recently logged-in users.
ld	Link-editor for object files.
ldd	List dynamic dependencies of executable files or shared objects.
less	Opposite of the more command.

lex	Generate programs for lexical tasks.
link	Calls the link function to create a link to a file.
ln	Creates a link to a file.
lo	Allows you to exit from a program, shell or log you out of a Unix network.
locate	List files in databases that match a pattern.
login	Signs into a new system.
logname	Returns users login name.
logout	Logs out of a system.
losetup	Sets up and controls loop devices.
lp	Prints a file on System V systems.
lpadmin	Configure the LP print service.
lpc	Line printer control program.
lpq	Lists the status of all the available printers.
lpr	Submits print requests.

lprm	Removes print requests from the print queue.
lpstat	Lists status of the LP print services.
ls	Lists the contents of a directory.
lsof	Lists open files.
lzcat	View compressed .lzma files.
lzma	Compress files to .lzma file.
mach	Display the processor of the current host.
mail	One of the ways that allows you to read/send E-Mail.
mailcompat	Provide SunOS 4.x compatibility for the Solaris mailbox format.
mailx	Mail interactive message processing system.
make	Executes a list of shell commands associated with each target.
man	Display the documentation (manual page) of a given command.
merge	Performs a merge of the contents of three files.
mesg	Control if non-root users can send text messages to you.
mii-tool	View, manipulate media-independent interface status.

mkdir	Create a directory.
mkfs	Build a Linux file system, usually a hard disk partition.
mkswap	Sets up a Linux swap area.
modprobe	Adds and removes modules from the linux kernel.
more	Displays text one screen at a time.
mount	Creates a file systems and remote resources.
mt	Magnetic tape control.
mv	Renames a file or moves it from one directory to another directory.
myisamchk	Checks, repairs, optimises, or fetches information about a MySQL database.
mysql	An open-source relational database management system.
nc	TCP/IP swiss army knife.
neqn	Language processors to assist in describing equations.
netstat	Shows network status.
newalias	Install new elm aliases for user or system.

newform	Change the format of a text file.
newgrp	Log into a new group.
nice	Invokes a command with an altered scheduling priority.
niscat	Display NIS+ tables and objects.
nischmod	Change access rights on a NIS+ object.
nischown	Change the owner of a NIS+ object on a system running Solaris.
nischttl	Change the time to live value of a NIS+ object.
nisdefaults	Display NIS+ default values.
nisgrep	Utilities for searching NIS+ tables.
nismatch	Utilities for searching NIS+ tables.
nispasswd	Change NIS+ password information.
nistbladm	NIS+ table administration command.
nl	Numbers the lines in a file.
nmap	Network exploration tool and security port scanner.

nohup	Runs a command even if the session is disconnected or the user logs out.
nroff	Formats documents for display or line-printer.
nslookup	Queries a name server for a host or domain lookup.
od	Dump files in octal and other formats.
on	Execute a command on a remote system, but with the local environment.
onintr	Shell built-in functions to respond to (hardware) signals.
optisa	Determine which variant instruction set is optimal to use.
pack	Shrinks file into a compressed file.
pagesize	Display the size of a page of memory in bytes, as returned by getpagesize.
parted	A disk partition manipulation program.
partprobe	Informs the operating system about changes to the partition table.
passwd	Allows you to change your password.
paste	Merge corresponding or subsequent lines of files.

pax	Read/write and writes lists of the members of archive files and copy directory hierarchies.
pcat	Compresses file.
perl	Perl is a programming language optimized for scanning arbitrary text files, extracting information from those text files.
pg	Files perusal filters for CRTs.
pgrep	Examine the active processes on the system and reports the process IDs of the processes
pico	Simple and very easy to use text editor in the style of the Pine Composer.
pine	Command line program for Internet News and Email.
ping	Sends ICMP ECHO_REQUEST packets to network hosts.
pkill	Examine the active processes on the system and reports the process IDs of the processes
poweroff	Stop the computer.
pr	Formats a file to make it look better when printed.
priocntrl	Displays or sets scheduling parameters of processes.
printenv	Prints all or part of environment.

printf	Write formatted output.
ps	Reports the process status.
pstree	Displays processes in tree format.
pvs	Display the internal version information of dynamic objects within an ELF file.
pwd	Print the current working directory.
quit	Allows you to exit from a program, shell or log you out of a Unix network.
rcp	Copies files from one computer to another computer.
readlink	Prints the value of a symbolic link or canonical file name.
reboot	Stop the computer.
red	The "restricted" version of a line-oriented file editor.
rehash	Recomputes the internal hash table of the contents of directories listed in the path.
rename	Renames multiple files using a regular expression.
renice	Alters the priority of running processes.

repeat	Shell built-in functions to repeatedly execute action(s) for a selected number of times.
replace	A string-replacement utility.
rgview	A programmer's text editor.
rgvim	A programmers text editor.
rlogin	Establish a remote connection from your terminal to a remote machine.
rm	Deletes a file without confirmation (by default).
rmdir	Deletes a directory.
rn	Reads newsgroups.
route	Show and manipulate the IP routing table.
rpcinfo	Report RPC information.
rsh	Runs a command on another computer.
rsync	Faster, flexible replacement for rcp.
rview	A programmers text editor.
rvim	A programmers text editor.

s2p	Convert a sed script into a Perl script.
sag	Graphically displays the system activity data stored in a binary data file by a previous sar run.
sar	Displays the activity for the CPU.
scp	Transfers files securely over a network connection.
screen	Screen manager with VT100/ANSI terminal emulation.
script	Records everything printed on your screen.
sdiff	Compares two files, side-by-side.
sed	Allows you to use pre-recorded commands to make changes to text.
sendmail	Sends mail over the Internet.
service	Runs a System V init script.
set	Set the value of an environment variable.
setenv	Set the value of an environment variable.
setfacl	Modify the Access Control List (ACL) for a file or files.
sethostname	System calls or set the hostname of the current processor.

sfdisk	A low-level disk partitioning program.
sftp	Secure file transfer program.
sh	Runs or processes jobs through the Bourne shell.
shred	Delete a file securely, first overwriting it to hide its contents.
shutdown	Turn off the computer immediately or at a specified time.
sleep	Waits a x amount of seconds.
slogin	OpenSSH SSH client (remote login program).
smbclient	An ftp-like client to access SMB/CIFS resources on servers.
sort	Sorts the lines in a text file.
spell	Looks through a text file and reports any words that it finds in the text file that are not in the dictionary.
split	Split a file into pieces.
startx	Starts an X Window System session.
stat	Display file or filesystem status.
stop	Control process execution.
strftime	Formats strings that represent the system date and time.

strip	Discard symbols from object files.
stty	Sets options for your terminal.
su	Become superuser or another user.
sudo	Executes any command as the superuser.
swapoff	Disables a Linux swap area.
swapon	Enables a Linux swap area.
sysinfo	Get and set system information strings.
sysklogd	Linux system logging utilities.
tabs	Set tabs on a terminal.
tail	Delivers the last part of the file.
talk	Talk with other logged in users.
tac	Concatenate and print files in reverse.
tar	Create tape archives and add or extract files.
tbl	Preprocessor for formatting tables for nroff or troff.

tcopy	Copy a magnetic tape.
tcpdump	Dump traffic on a network.
tee	Read from an input and write to a standard output or file.
telinit	Process control initialization.
telnet	Uses the telnet protocol to connect to another remote computer.
test	Check file types and compare values.
time	Used to time a simple command.
timex	The timex command times a command; reports process data and system activity.
todos	Converts text files between DOS and Unix formats.
top	Display Linux tasks.
touch	Change file access and modification time.
tput	Initialize a terminal or query terminfo database.
tr	Translate characters.
traceroute	Print the route packets take to network host.
trap	A function which "traps" signals and interrupts, and reacts to

them.

tree	List the contents of a file hierarchy visually in a tree format.
troff	Typeset or format documents.
tty	Print the file name of the terminal connected to standard input.
ul	Reads the named filenames or terminal and does underlining.
umask	Get or set the file mode creation mask.
unalias	Remove an alias.
unhash	Remove internal hash table.
uname	Print name of current system.
uncompress	Uncompressed compressed files.
uniq	Report or filter out repeated lines in a file.
unlink	Call the unlink function to remove the specified file.
unlzma	Decompress .lzma file.
unmount	Disconnects a file systems and remote resources.
unpack	Expands a compressed file.

until	Execute a set of actions while/until conditions are evaluated TRUE.
unxz	Decompress .xz files.
unzip	List, test and extract compressed files in a ZIP archive.
uptime	Display information about how long the system has been running.
useradd	Create a new user or updates default new user information.
userdel	Remove a users account.
usermod	Modify a users account.
vacation	Reply to mail automatically.
vgrind	Grind nice program listings
vi	Screen-oriented (visual) display editor based on ex.
vim	A programmer's text editor.
view	A programmer's text editor.
vipw	A special command to safely edit password files.
visudo	A special command to safely edit the "sudoers" file.

vmstat	Reports statistics about virtual memory usage.
w	Show who is logged on and what they are doing.
wait	Await process completion.
wall	Send a message to everybody's terminal.
wc	Displays a count of lines, words, and characters in a file
wget	Downloads files via HTTP or FTP, such as web pages.
whatis	Displays short manual page descriptions.
whereis	Locate a binary, source, and manual page files for a command.
while	Repetitively execute a set of actions while/until conditions are evaluated TRUE.
which	Locate a command.
who	Displays who is on the system.
whoami	Print effective userid.
whois	Internet user name directory service.
write	Send a message to another user.

X	Execute the X Window System.
Xorg	The executable of the X Window System server.
xargs	Build and execute complex commands across multiple files.
xfd	Display all the characters in an X font.
xhost	Server access control program for X.
xinit	The initializer of the X Window System.
xlsfonts	Server font list displayer for X.
xset	User preference utility for X.
xterm	Terminal emulator for X.
xrdb	X server resource database utility.
xz	Compress files to .xz files.
xzcat	View compressed .xz files.
yacc	Short for yet another compiler-compiler, yacc is a compiler.
yes	Repeatedly output a line with all specified STRING(s), or 'y'.
yppasswd	Changes network password in the NIS database.

yum	Interactive rpm based package manager.
zcat	Compress files.
zip	Compression and file packaging utility.
zipinfo	Display technical information about a zip file.

MS-DOS vs. Linux and Unix

If you are able to navigate a computer using MS-DOS or the Windows command line, you should be able to quickly pick up on the navigation of Linux and Unix. In the below chart is a listing of common MS-DOS commands with their Linux and Unix counterpart.

MS-DOS	Linux and Unix
attrib	chmod
backup	tar
dir	ls
cls	clear
copy	cp
del	rm
deltree	rm -R rmdir

edit	vi pico
format	fdformat, mount, and umount
move and rename	mv
Type	less <file>
cd	cd chdir
more < file	more file
md	mkdir
win	startx

Linux ABCs

Developed by Linus Torvalds and further elaborated by a number of developers throughout the world, Linux (lee'nuhks/ or /li'nuks/,_not_/li:'nuhks) is a freely available multitasking and multiuser operating system. From the outset, Linux was placed under General Public License (GPL). The system can be distributed, used, and expanded free of charge. In this way, developers have access to all the source codes, thus being able to integrate new functions or to find and eliminate programming bugs quickly. Thereby drivers for new adapters (SCSI controller, graphics cards, etc.) can be integrated very rapidly.

Linux and Unix Commands

	* See the Linux and Unix overview page for a brief description on all commands on one page.
A	a2p | ac | access | alias | agrep | apropos | apt-cache | apt-get | aptitude | ar | arch | arp | as |aspell | at | awk
B	basename | bash | bc | bdiff | bfs | bg | biff | break | bs | bye
C	cal | calendar | cancel | cat | cc | cd | chdir | checkeq | checknr | chfn | chgrp | chkey | chmod |chown | chroot | chsh | cksum | clear | cls | cmp | col | comm | compress | continue | copy | cp |cpio | crontab | crypt | csh | csplit | ctags | cu | curl | cut
D	date | dc | dd | df | depmod | deroff | dhclient | diff | dig | dircmp | dirname | dmesg | dos2unix| dpkg | dpost | du
E	echo | ed | edit | egrep | eject | elm | emacs | enable | env | eqn | ex | exit | expand | expr
F	fc | fdisk | fg | fgrep | file | find | findsmb | finger | fmt | fold | for | foreach | fromdos | fsck |ftp | fuser
G	gawk | getfacl | gethostname | gpasswd | gprof | grep | groupadd | groupdel | groupmod |gunzip | gview | gvim | gzip
H	halt | hash | hashstat | head | help | history | host | hostid | hostname
I	id | ifconfig | ifdown | ifup | ip | init | info | iostat | isalist | iwconfig
J	jobs | join
K	keylogin | kill | killall | ksh
L	last | ld | ldd | less | lex | link | ln | lo | locate | login | logname | logout | losetup | lp |lpadmin | lpc | lpq | lpr | lprm |

	lpstat | ls | lsof | lzcat | lzma
M	mach | mail | mailcompat | mailx | make | man | merge | mesg | mii-tool | mkdir | mkfs |modprobe | more | mount | mt | mv | myisamchk | mysql
N	nc | neqn | netstat | newalias | newform | newgrp | nice | niscat | nischmod | nischown |nischttl | nisdefaults | nisgrep | nismatch | nispasswd | nistbladm | nl | nmap | nohup | nroff |nslookup
O	od | on | onintr | optisa
P	pack | pagesize | passwd | paste | pax | pcat | perl | pg | pgrep | pico | pine | ping | pkill |poweroff | pr | printenv | priocntl | printf | ps | pstree | pvs | pwd
Q	quit
R	rcp | readlink | reboot | red | rehash | remsh | renice | repeat | replace | rgview | rgvim | rlogin| rm | rmail | rmdir | rn | route | rpcinfo | rsh | rsync | rview | rvim
S	s2p | sag | sar | scp | screen | script | sdiff | sed | sendmail | service | set | setenv | setfacl |sethostname | settime | sftp | sh | shred | shutdown | signals | sleep | slogin | smbclient | sort| spell | split | stat | stop | strftime | strip | stty | su | sudo | sysinfo | sysklogd
T	tabs | tac | tail | talk | tar | tbl | tcopy | tcpdump | tee | telinit | telnet | test | time | timex |todos | top | touch | tput | tr | traceroute | tree | troff | tty
U	ul | umask | unalias | uname | uncompress | unhash | uniq | unlzma | unmount | unpack |untar | until | unxz | unzip

	|uptime | useradd | userdel | usermod
V	vacation | vedit | vgrind | vi | view | vim | vipw | vmstat
W	w | wait | wall | wc | wget | whereis | whatis | which | while| who | whoami | whois | write
X	X | xargs | xfd | xhost | xlsfonts | xorg | xset | xterm | xrdb | xz | xzcat
Y	yacc | yes | yppasswd | yum
Z	zcat| zip | zipcloak | zipinfo | zipnote| zipsplit

* In addition to each of the above explanations, additional information about a specified command for your Unix or Linux variant can be found by using the mancommand.

Advantages of Linux

- Low cost: There is no need to spend time and huge amount money to obtain licenses since Linux and much of it's software come with the GNU General Public License. There is no need to worry about any software's that you use in Linux.
- Stability: Linux has high stability compared with other operating systems. There is no need to reboot the Linux system to maintain performance levels. Rarely it freeze up or slow down. It has continuous up-times of hundreds of days or more.
- Performance: Linux provides high performance on various networks. It has the ability to handle large numbers of users simultaneously.
- Networking: Linux provides a strong support for network functionality; client and server systems can be easily set up on any computer running Linux. It can perform tasks like network backup faster than other operating systems.

- Flexibility: Linux is very flexible. Linux can be used for high performance server applications, desktop applications, and embedded systems. You can install only the needed components for a particular use. You can also restrict the use of specific computers.
- Compatibility: It runs all common Unix software packages and can process all common file formats.
- Wider Choice: There is a large number of Linux distributions which gives you a wider choice. Each organization develop and support different distribution. You can pick the one you like best; the core function's are the same.
- Fast and easy installation: Linux distributions come with user-friendly installation.
- Better use of hard disk: Linux uses its resources well enough even when the hard disk is almost full.
- Multitasking: Linux is a multitasking operating system. It can handle many things at the same time.
- Security: Linux is one of the most secure operating systems. File ownership and permissions make linux more secure.
- Open source: Linux is an Open source operating systems. You can easily get the source code for linux and edit it to develop your personal operating system.
- Today, Linux is widely used for both basic home and office uses. It is the main operating system used for high performance business and in web servers. Linux has made a high impact in this world.

CHAPTER 5: PROJECT DESCRIPTION

In this chapter, schematic diagram and interfacing of ARM-11 microprocessor with each module is considered.

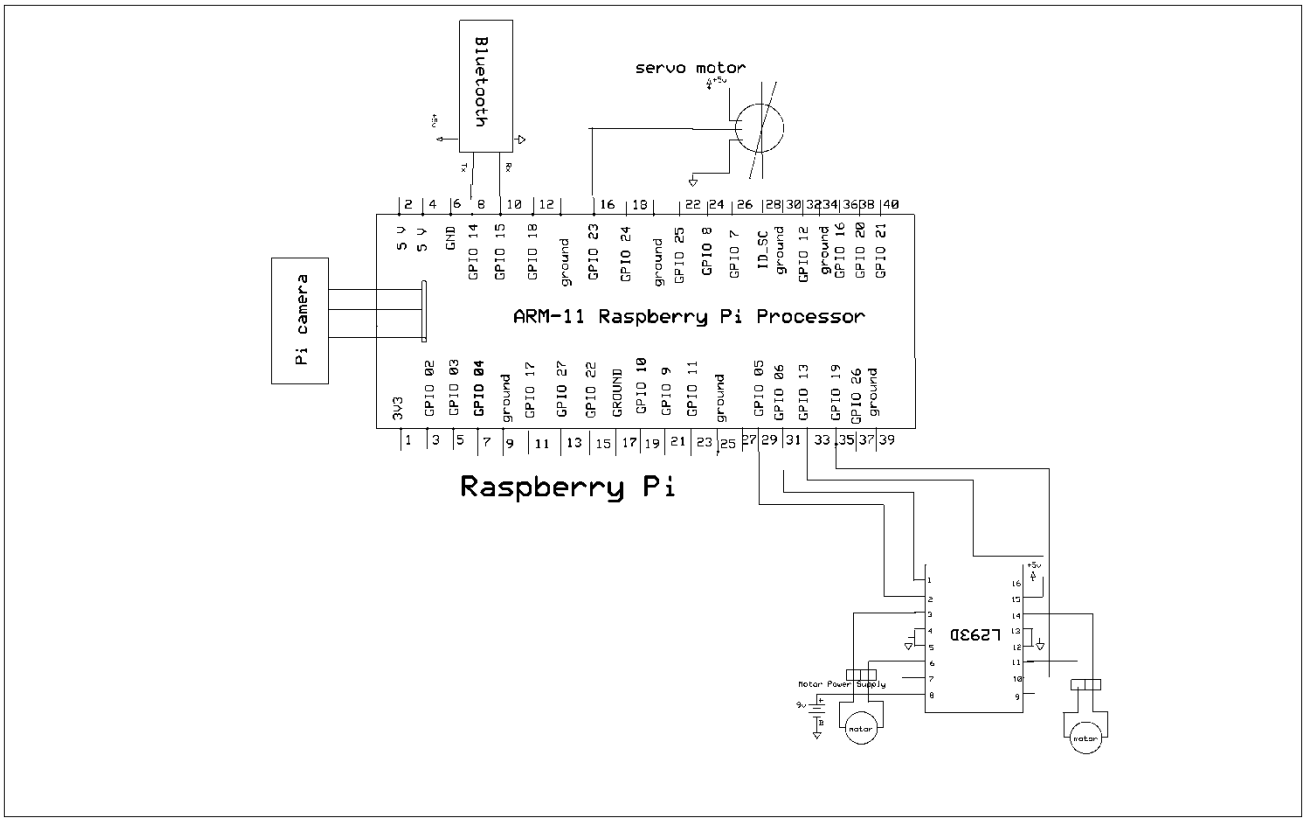


Fig 5.1(i): Virtual Telepresence Robot Using Raspberry Pi

The above schematic diagram of **Virtual Telepresence Robot Using Raspberry Pi** explains the interfacing section of each component with Raspberry Pi and input output modules.

CHAPTER 6: ADVANTAGES AND DISADVANTAGES

Advantages:

1. Highly efficient and user-friendly design.
2. Easy to operate.
3. Low power consumption.
4. Wi-Fi based communication along with obstacle detection.

Disadvantages:

1. Interfacing fingerprint scanner to ARM-11 processor is highly sensitive.
2. Attaching mail with the help of Raspberry Pi is difficult.

Applications:

1. Examination centers.
2. Home security.
3. Industrial security

CHAPTER 7: RESULTS

7.1 Result:

The project "**Virtual Telepresence Robot Using Raspberry Pi"** the aim of the project is achieved by using Raspberry pi. This robot with a Pi camera is placed in a remote location to capture the environment in visual form using Raspberry Pi. The captured visuals are displayed on the user's virtual reality (VR) headset along with obstacle detection. The robot can also be moved in any direction (in which the user turns his head, say, right or left) through an app installed in the user's smartphone.

7.2 Conclusion:

Integrating features of all the hardware components used have been developed in it. Presence of every module has been reasoned out and placed carefully, thus contributing to the best working of the unit. Secondly, using highly advanced IC's with the help of growing technology, the project has been successfully implemented. Thus the project has been successfully designed and tested.

7.3 Future Scope:

Our project "**Virtual Telepresence Robot Using Raspberry Pi**" is mainly intended to give direction to the robot using VR head set.

The controlling device of the whole system is a Raspberry Pi processor. Arduino, Pi Camera, Servomotor, Bluetooth module are interfaced to the ARM CORTEX A5 1.2 Ghz processor Raspberry Pi. The data received by the fingerprint scanner is fed to theARM CORTEX A5 1.2 Ghz processor. The processor acts accordingly and either opens or does not open the door. In achieving the task the controller is loaded with a program written using Embedded Linux programming language.

This project can be extended using high efficiency GSM module using which we can send the user about the unauthorized access. The GSM module gives the SMS messages of the authentication through SMS. Using GSM we can know which direction the camera is moved and we can get direction in the form of message.

REFERENCES

The sites which were used while doing this project:

- www.wikipedia.com
- www.allaboutcircuits.com
- www.microchip.com
- www.howstuffworks.com
- http://openrelief.org/
- http://www.theregister.co.uk/2012/06/12/raspberry_pi_drone/
- http://www.zdnet.com/raspberry-pi-designer-hints-at-future- version-4010025712/
- http://janbierens.com/2012/05/22/raspberry-pi-and-the-future/
- http://en.wikipedia.org/wiki/Raspberry_Pi
- http://www.pcpro.co.uk/reviews/desktops/374290/raspberry-pi-model-b
- http://www.techrepublic.com/blog/european-technology/raspberry-pi-five-ways-business-can-use-it/610

Books referred:

1. Raj kamal –Microcontrollers Architecture, Programming, Interfacing and System Design.

2. PCB Design Tutorial –David.L.Jones.

Program Code:

The program code which is dumped in the microcontroller of our project is shown below.

```
import serial
from time import sleep

import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BCM)
GPIO.setup(4,GPIO.OUT)
GPIO.setup(27,GPIO.OUT)

p.start(8) # 8 middle - for up increase
p1.start(6.5) # 6.5 middle - for left increase value

time.sleep(3)

while 1:
    try:
        s.bind(('0.0.0.0', 5555))

        time.sleep(0.2)
        message, address = s.recvfrom(8192)

        print(message)         #whole message signal
        print(z1 ,z2)
        a1=float(z1)
        b1=float(z2)
        s.close();


        if(a1>7):       #uppper limit
            a=4 #12.3

        elif(a1>6):
            a=5 #11.9
```

```
elif(a1>5):
    a=6 #11.2

elif(a1>3):
    a= 6.5 #10.8

elif(a1>2):
    a= 7 #10.5

elif(a1>1):
    a=7.5

elif(a1>0):
    a=8 #9      #middle limit

elif(a1>-1):
    a=8.5 #8

elif(a1>-2):
    a=9#7

elif(a1>-3):
    a=10#7.5

elif(a1>-4):
    a=11#7.5

else:
    a=12.3 #6.9
time.sleep(0.2)


if(b1>27):
    b=2

elif(b1>24):
    b=2.5

elif(b1>21):
    b=3

elif(b1>18):
    b= 3.5

elif(b1>15):
    b=4

elif(b1>12):
    b=4.5

elif(b1>9):
    b=5

elif(b1>5):
```

```
    b=5.5

elif(b1>2):
    b=6

elif(b1>0):
    b=6.5

elif(b1>-2):
    b=7

elif(b1>-5):
    b=7.5

elif(b1>-9):
    b=8

elif(b1>-12):
    b=8.5

elif(b1>-15):
    b=9

elif(b1>-18):
     b=9.5

elif(b1>-21):
    b= 10

elif(b1>-24):
    b=10.5

elif(b1>-27):
    b=11

p1.ChangeDutyCycle(b)
time.sleep(0.4)
p.ChangeDutyCycle(a)
time.sleep(0.4)
```