Operating system

The traditional approach to providing a system to meet a wide range of requirements has been to incorporate all the requirements inside a general purpose operating system. The typical arrangement is illustrated in figure bellow.

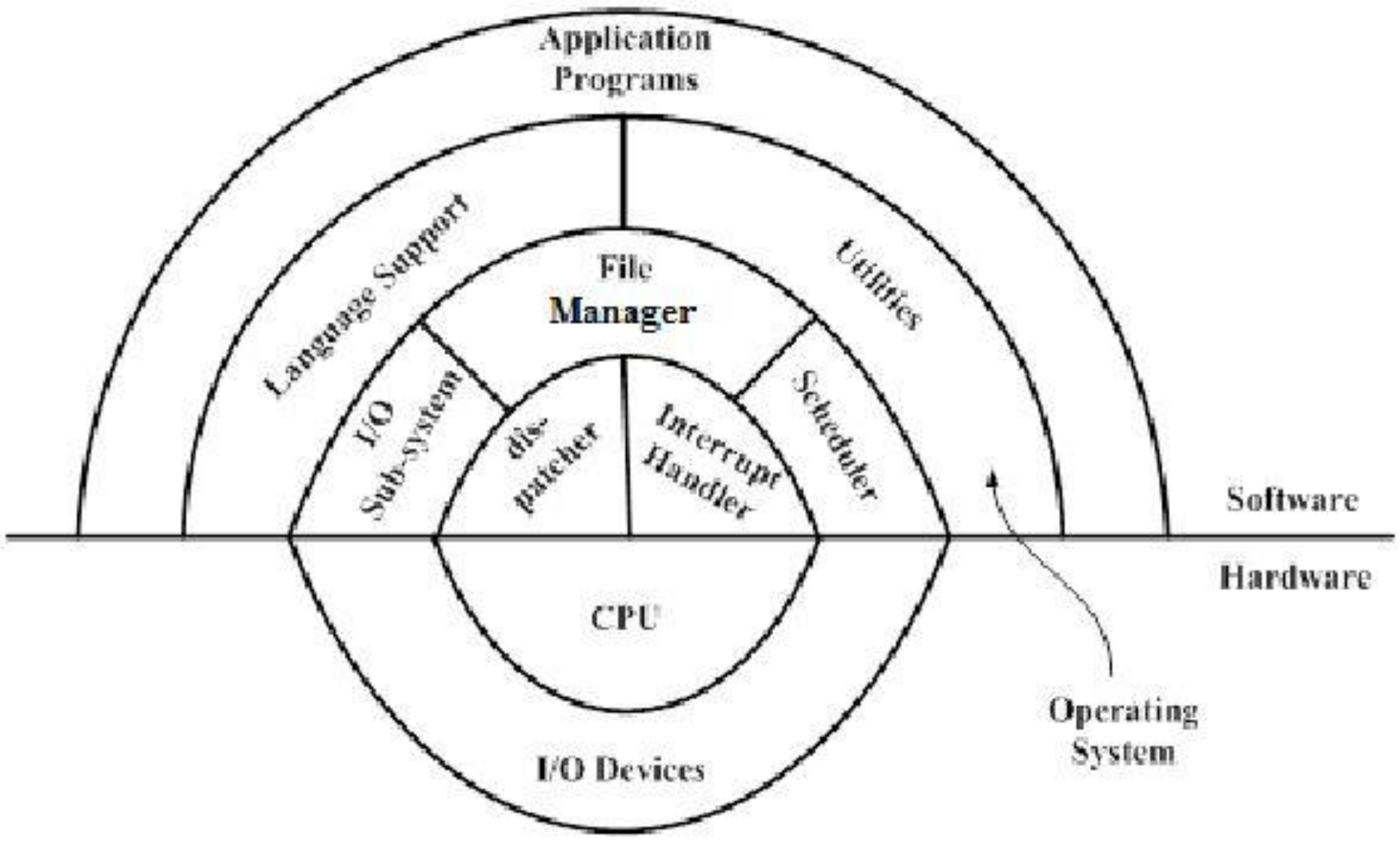

Fig .General Purpose Operating System

Access to the hardware of the system and of the I/O devices is through the operating system. In any real-time and multi-programming systems restriction of access is enforced by hardware and software traps.

In single-job operating systems access through operating system is not usually enforced.

In addition to supporting and controlling the basic activities, operating systems practically provide various utility programs, e.g. loaders, linkers, assemblers and debuggers, as well as run-time support for high-level languages.

(1) <u>Single-user , single-job operating system</u>

As an example of single-user, single-task , disk-based operating system, the CP/M80 system of digital research will be describe. This system is available for 8080, z80 based computer systems.

CP/M consist of three major sections:

1) Console command processor (CCP).
2) Basic input/output system (BIOS).
3) Basic Disk operating system (BDOS).

The relationship between the various sections of the operating system, the computer hardware and the user is illustrated in figure below:

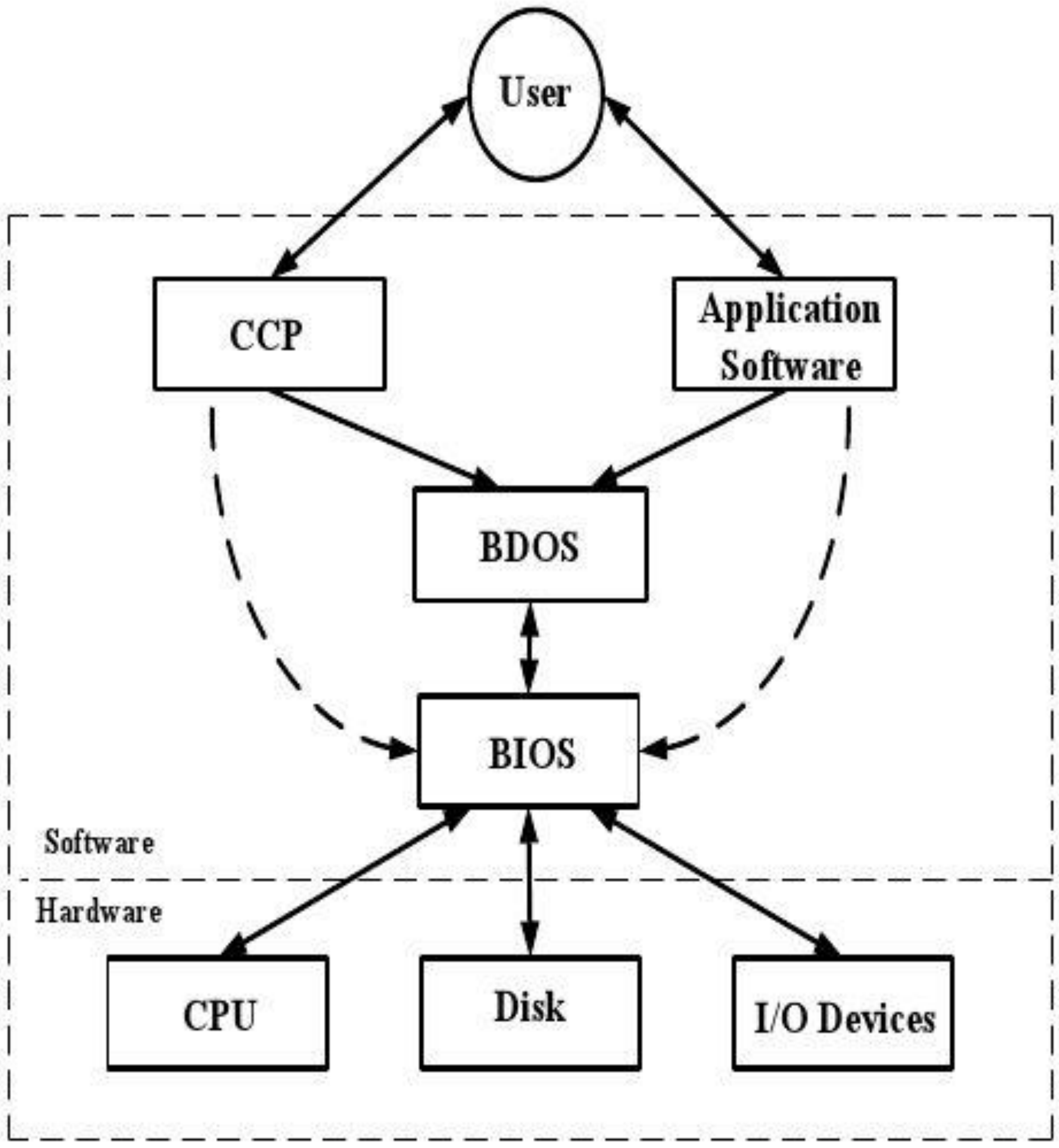

Fig .General Structure of CP/M80 Operating System

The console command processor provides a means by which the user communicate with the operating system from the computer console device. It is use to issue commands to the operating system and to provide the user with information about the actions being performed by the operating system.

The actual processing is the commands issued by the user is done by the BDOS which also handles the input and output and the file operations on the disks.

The BDOS makes the act management of the file and input/output operations transparent to the use. Application programs will normally communicate with the hardware of the system through 'system calls' which are processed by the BDOS.

The BIOS contains the various device drivers which manipulate the physical devices; this section of the operating system may vary from implementation to implementation as it has to operate directly with the underlying hardware of the computer. For example, depending on the manufacturer the physical addresses of the peripherals may vary, as may the type of peripherals, and the type of control used for the disk drive . All these differences will be accommodated in the codinet the BIOS.

On starting the computer the three sub-systems which reside on the disk and loaded into the memory. The computer must, therefore, have some means to booting the operating system from a systems disk. The normal arrangement it is provide a small bootstrap loader in a ROM chip.

The combination of BDOS and BIOS frequently referred to as the FDOS (Functional Disk Operating System) and these units have to remain in memory.

<u>CCP direct Commands</u>

There are many direct commands available to the user; the routines to support these commands are held in the CCP area of memory.

1- DIR : Display a list of all filenames.
2- ERA : Erased a list of files.
3- REN < new > , < old > : rename the file <old> as <new>.
4- Type : listed the contents of file.

<u>Basic Disk Operating System BDOS</u>

The CCP facilities are useful in developing user programs, but one of the purposes of the operating system is to support user programs while they are running.

The facilities provided by the BDOS are used for this purpose and these are essentially concerned with operating the input and output devices attached to the system.

Input/output Devices

To types of I/O devices are supported:

1-Sequential, data is transferred one byte at a time.

2-Blocked, data is transferred in fixed length blocks or records of 128 bytes these are assumed to be disk drives in CP/M.

In dealing with I/O devices CP/M considers devices to be 'logical' 'physical'. Logical devices are software constructs used to simplify the user inter user programs perform input and output to logical devices, the BDOS connects the logical devices to the physical devices. The actual operation of the physical device performed by software in the BIOS.

(2) Real time multi-tasking operating system

The natural way to structure a typically computer control system is in the form of a number of different tasks which all apparently run in parallel.

The implementation of a design based on this approach is made easier if an operating system which supports multi-tasking can be used.

The traditional real-time operating systems are based on the assumption that all the tasks in the system will be executed an a single CPU or processor. In this section the same assumption will be made.

Confusion can arise between multi-user or multi-programming operating system and multi-tasking operating systems.

The function of a multi-user operating system is illustrated in figure below:

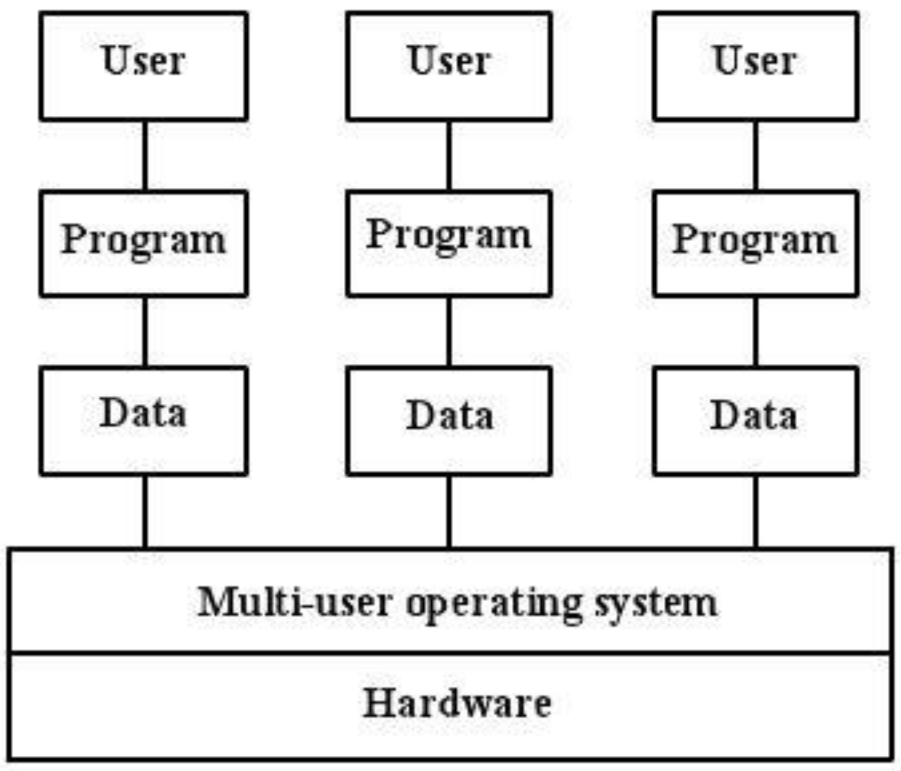

Fig. Multi-User Operating System

The operating system ensures that each user can run a single program as if they had the whole of the computer system for their program.

Although at any given instance it is not possible to predict which user will have the use of the CPU or even if the user's code is in the memory.

The operating system ensures that one user program cannot interfere with the operation of another user program.

Each user program runs in its own protected environment: a primary concern of the operating system is to prevent one program corrupting another , either deliberately or through error. In a multi-tasking operating system it is assumed that there is a single user and that the various tasks are to co-operate to serve the requirements of the user. Co-operation will require that the tasks communicate with each other and share common data. This is illustrated in figure below:

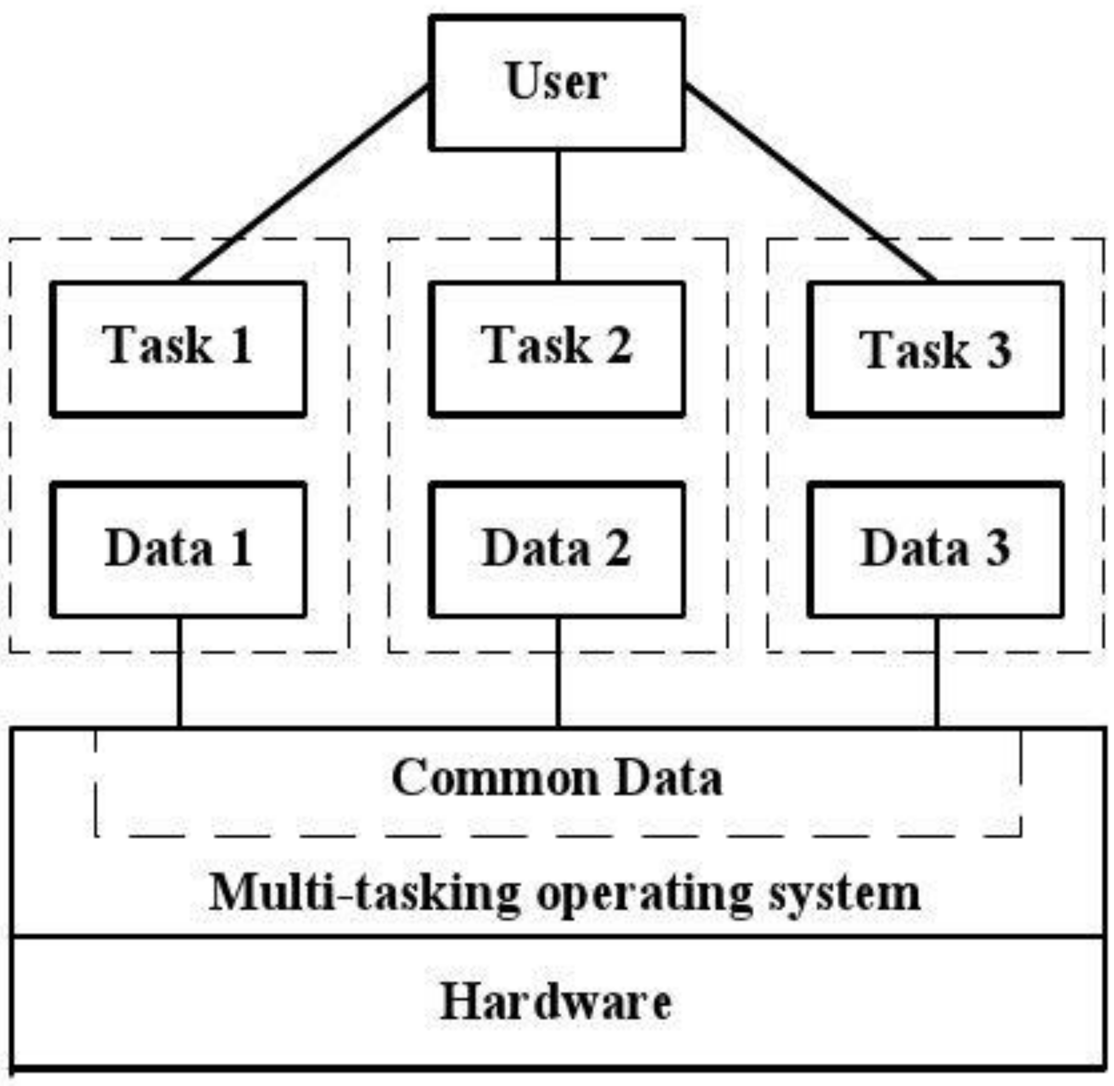

Fig. Multi-Tasking Operating System

In a good multi-tasking operating system the way in which tasks communicate and share data will be regulated such that the operating system is able to prevent inadvertent communication or data access (arising through an error in the coding of one task) and hence protect data which is private to a task.

A fundamental requirement of an operating system is to allocate the resources of the computer to the varies activities which have to be performed in a real time.
Operating system this allocation procedure is complicated by the fact that some of the activities are time critical and hence have a higher priority than others. There must therefore be some means of allocating priorities to tasks and of scheduling allocation of CPU time to the tasks according to some priority scheme.

A task may use another task, i.e. it may require certain activities which are contained in another task to be performed and it may itself be used by another task.
Thus tasks may need to communicate with each other. The operating system therefore has to have some means of enabling tasks, either to share memory for the exchange of data or to provide a mechanism by which tasks can send messages to each other. In addition, tasks may need to be invoked by external events; hence the operating system must support the use of interrupts.
Similarly, tasks may need to share data and they may require access to various hardware and software components; hence there has to be a mechanism for preventing two tasks from attempting to use the same resource at the same time.
In summary, a real time multi-tasking operating system (RTMTOS) has to support the resource sharing and the timing requirements of the tasks and the functions can be divided as follows:

- task scheduling
- interrupt handling
- memory management
- code sharing
- device sharing; and
- inter-task communication and data sharing