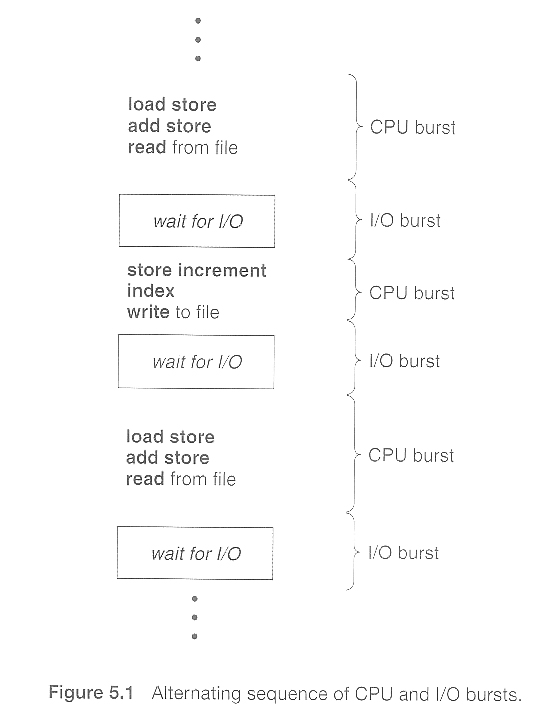
**CPU SCHEDULING**

**Scheduling:-**

* CPU scheduling is the basis of multiprogrammed operating systems.
* By switching the CPU among processes, the operating system can make the computer more productive.
* In a simple system running a single process, the time spent waiting for I/O is wasted, and for this time CPU is idle, which is causing performance degradation.
* A scheduling system allows one process to use the CPU while another is waiting for I/O, thereby making full use of otherwise lost CPU cycles.

**CPU and I/O Burst Cycles:-**

* Process execution consists of a cycle of **CPU execution** and **I/O wait**.
* CPU burst is when process is executed by CPU
* I/O Burst is when process is waiting for I/O operation.



**CPU Scheduler:-**

* Whenever the CPU becomes idle, it is the job of the CPU Scheduler to select another process from the ready queue to run next.
* This selection process is carried out by short term scheduler.
* The scheduler selects from the processes in memory that are ready to execute and allocates the CPU to that process.

**Dispatcher:-**

* The dispatcher is responsible for giving control of CPU to the process selected by the short-term scheduler.
* The time it takes for the dispatcher to stop one process and start another running is known as the dispatch latency.

**Preemptive Scheduling:-**

1. CPU scheduling decisions take place under one of four conditions:

* When a process switches from the running state to the waiting state, such as for an I/O request or invocation of the wait( ) system call.
* When a process switches from the running state to the ready state, for example in response to an interrupt.
* When a process switches from the waiting state to the ready state, say at completion of I/O or a return from wait( ).
* When a process terminates.

1. For conditions 1 and 4 there is no choice - A new process must be selected.
2. For conditions 2 and 3 there is a choice - To either continue running the current process, or select a different one.
3. If scheduling takes place only under conditions 1 and 4, the system is said to be non-preemptive, or cooperative. Under these conditions, once a process starts running it keeps running, until it either voluntarily blocks or until it finishes. Otherwise the system is said to be preemptive.
4. Windows used non-preemptive scheduling up to Windows 3.x, and started using pre-emptive scheduling with Win95.

### Scheduling Criteria:-

1. There are several different criteria to consider when trying to select the "best" scheduling algorithm for a particular situation and environment, including:

* **CPU utilization** - Ideally the CPU would be busy 100% of the time, so as to waste 0 CPU cycles. On a real system CPU usage should range from 40% ( lightly loaded ) to 90% ( heavily loaded. )
* **Throughput** - Number of processes completed per unit time. May range from 10 / second to 1 / hour depending on the specific processes.
* **Turnaround time** - Time required for a particular process to complete, from submission time to completion. ( Wall clock time. )
* **Waiting time** - How much time processes spend in the ready queue waiting their turn to get on the CPU.
* **Response time** - The time taken in an interactive program from the issuance of a command to the commence of a response to that command.

### Scheduling Algorithms:-