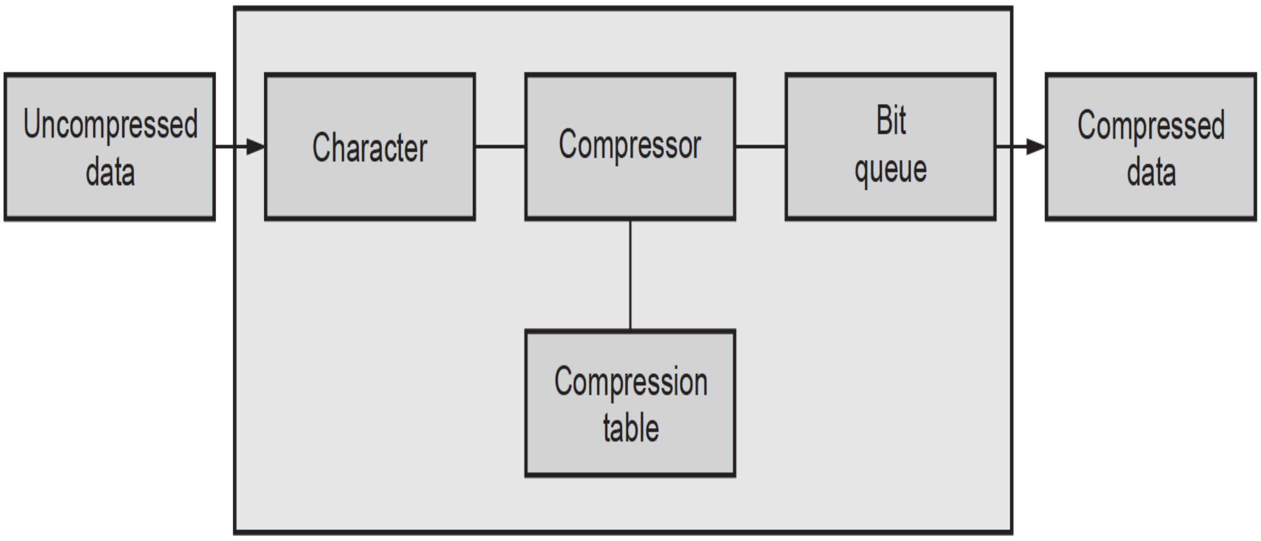
Multiple Tasks and Multiple Processes

Multiple Tasks and Multiple Processes

- Tasks are units of sequential code implementing the system actions and executed concurrently by an OS.
- Real time systems require that tasks be performed within a particular time frame. Task is related to the performance of the real time systems.
- A task, also called a thread, is a simple program that thinks it has the CPU all to itself. The design process for a real-time application involves splitting the work to be done into tasks responsible for a portion of the problem.
- Each task is assigned a priority, its own set of CPU registers and its own stack area.
- In the specified time constraint, system must produce its correct output. If system fail to meet the specified output, then the system is fail or quality decreases.
- Real time systems are used for space flights, air traffic control, high speed aircraft, telephone switching, electricity distribution, industrial processes etc.
- Real time system must be 100 % responsive 100 % of the time. Response time is measured in fractions of second, but this is an ideal not often achieved in the field.
- Real time database is updated continuously. In aircraft example, flight data is continuously changing so it is necessary to update. It includes speed, direction, location, height etc.
- A process is a sequential program in execution. Terms like job and task are also used to denote a process.
- A process is a dynamic entity that executes a program on a particular set of data. Multiple processes may be associated with one program.
- Task is a single instance of an executable program.

- In a multiprogramming environment, usually more programs to be executed than could possibly be rim at one time. In CPU scheduling, it switches from one process to another process. CPU resource management is commonly known as scheduling.
- Objective of the multiprogramming is to increases the CPU utilization. CPU scheduling is one kind of fundamental operating system functions.
- Each process has an execution state which indicates what process is currently doing. The process descriptor is the basic data structure used to represent the specific state for each process. A state diagram is composed of a set of states and transitions between states.



On-the-fly compression box

- Input and output of the compressor box is serial ports. It takes uncompressed data and processes it. Output of the box is compressed data. Given data is compressed using predefined compression table. Modem is used such type of box.
- The program's need to receive and send data at different rates. It is an example of rate control problems. It uses asynchronous input. You can provide a button for compressed mode and uncompressed mode.

- When user press uncompressed mode, the input data is passed through unchanged.
- The button will be depressed at a much lower rate than characters will be received, since it is not physically possible for a person to repeatedly depress a button at even slow serial line rates.
- Keeping up with the input and output data while checking on the button can introduce some very complex control code into the program.
- Sampling the button's state too slowly can cause the machine to miss a button depression entirely, but sampling it too frequently and duplicating a data value can cause the machine to incorrectly compress data.
- This problem is solved by maintaining counter.