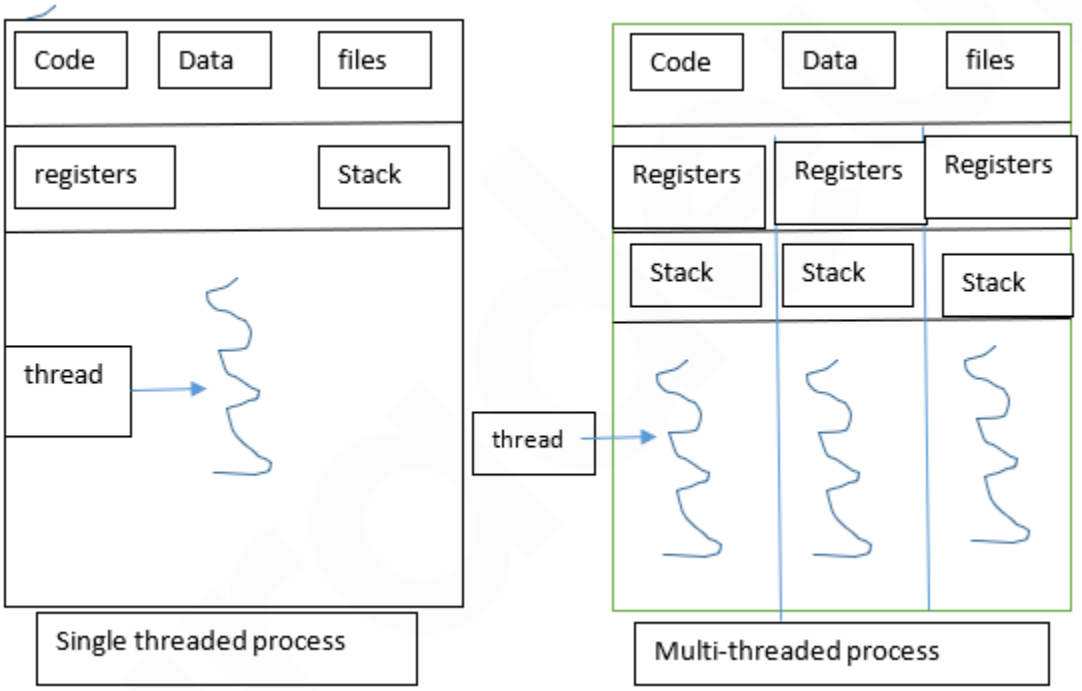
gradeup

Threads And CPU Scheduling

Content :-
1. Threads
2. Multi Level Queue
3. Multi Level Feedback Queue

1. **Threads :-**
Basic CPU utilization unit is known as thread; it comprises, a program counter, a thread ID, a stack and a register set. It shares with other threads belonging to the same process/job its data section, code section, and other operating- system resources, such as open files and signal. A traditional (or heavy weight) process/job has a single thread of control.

Motivation :-
Multithreaded models are run on many software package that are used in modern desktop PCs. An application typically is implemented as a separate process/job with several threads of control.



Benefits of multithreaded programming can be broken down into four major categories:-
1. Responsiveness
2. Resource sharing
3. Economy
4. Scalability

Multicore Programming : - A recent trend in the system design has been to place multiple computing cares on a single chip, where each core is appears as a separate processor to the operating system. Multithreaded programming provides the mechanism for more effective use of multiple cores and improved concurrency.
Five areas at present are challenges in programming for multicore systems:

1. **Dividing Activity :-** This involves examining applications to find the area that can be divided into separate , concurrent task and thus can run in parallel on individual cores.
2. **Balance :-** while identifying tasks that can run in parallel , programmers must also ensure that the tasks perform equal work to equal value.
3. **Data Splitting** : There is a separate cores for accessing and manipulating the data as there are different applications in which the tasks are divided .
4. **Data Dependency :** There must be examination of data dependency which is accessed by two or more tasks.
5. Testing and debugging: when the program is running in parallel on multi cores there are many different execution paths. Testing and debugging such concurrent programs in inherently more difficult than testing and debugging single- threaded application **.**

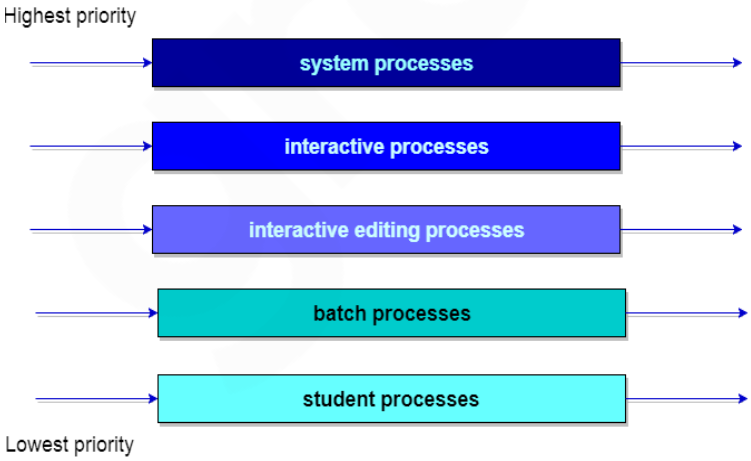
Multithreading Models
1. Many-to-One Model. :- The many-to-one model maps many user level threads to one kernel level thread. Thread management is done by the thread library in user space , so it is efficient ; but the entire process will be block if a thread makes the blocking system call
Ex-1

2. One-to-One Model:- In one-to-one model mapping of each user level thread to a kernel thread is done.The only drawback to this model is that creating a user that requires creating the corresponding kernel thread.
Ex-2

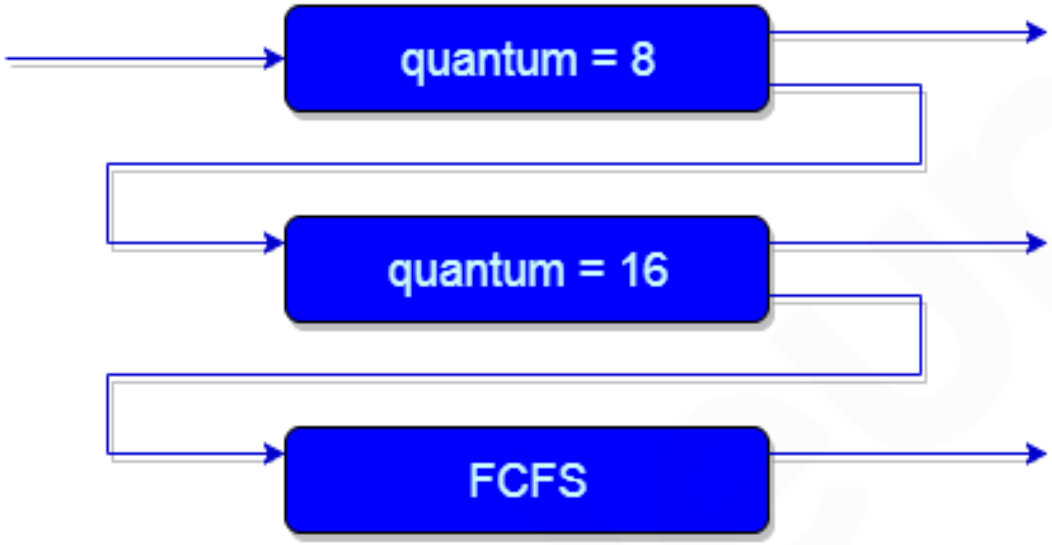
3. **Many-to-Many Model :-** The many-to-many model multiplexes many user-level threads to a smaller or equal number of kernel threads the number of kernel threads may be specific to either a particular applications or a particular machine.
Ex- 3

Note :- Kernel thread <= user Thread

Multi level Queue(MLQ) :-
In this the processes/jobs are easily classified into different groups.



Multi level Feedback Queue :-
In the multi level queue scheduling , the processes/jobs are permanently assigned to a queue as soon as they enter the system . The main advantage of this type of scheduling is that there is low scheduling overhead. The disadvantage is it is inflexible.



The parameters of multi level feedback queues are as follows :-
1. The number of queues
2. The scheduling algorithm of each queue
3. The method used to determine when to upgrade a process to a high-priority queue.
4. The method used to determine when to demote a process/job to low-priority queue
5. The method used to determine which queue a process will enter when that process needs service.

gradeup SUPER

Gradeup UGC NET
Super Superscription

Features:

1. 7+ Structured Courses for UGC NET Exam
2. 200+ Mock Tests for UGC NET & MHSET Exams
3. Separate Batches in Hindi & English
4. Mock Tests are available in Hindi & English
5. Available on Mobile & Desktop

Gradeup Super Subscription, Enroll Now