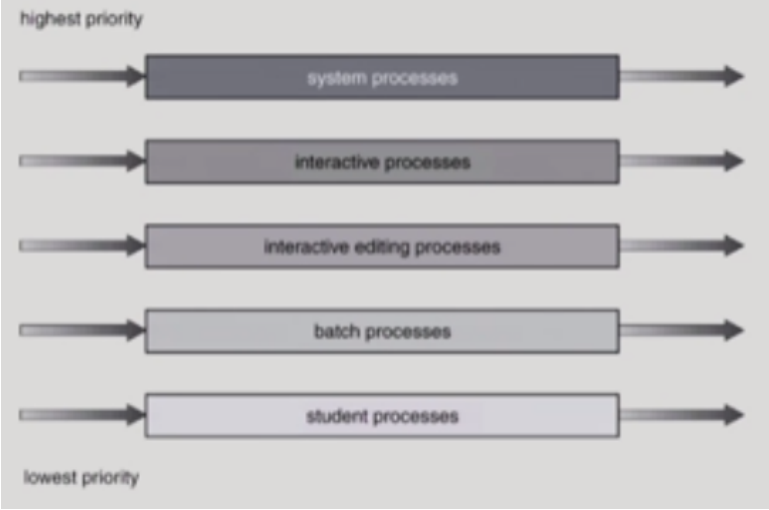
5. CPU Scheduling 2

속성

1. Multilevel Queue

- Ready Queue를 여러개로 분할한다.
    - 프로세스를 어느 줄(Queue)에 들어갈 것인가



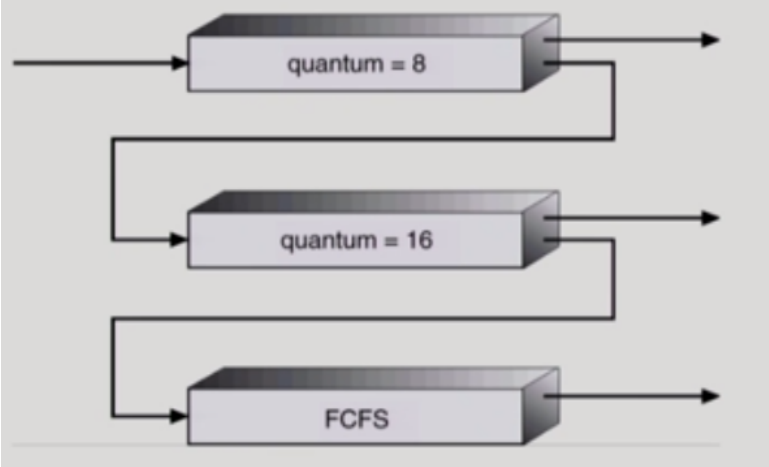
1. foreground (interactive) : `RR` 을 사용하는 것이 유리
2. background (bactch - no human interaction) : 사람과의 상호작용이 필요하지 않는 작업. `FCFS` 를 사용하는 것이 유리.

- 큐에 대한 스케줄링이 필요
    - Fixed priority scheduling

- Time slice : 각 큐에 CPU time을 적절한 비율로 전달

- **[단점]**
  - 차별적이다.

2. Multilevel Feedback Queue



- 프로세스가 다른 큐로 이동 가능
- 에이징(aging)을 이와 같은 방식으로 구현할 수 있다.
- Multilevel-feedback-queue scheduler를 정의하는 파라미터들
  - Queue의 수
  - 각 큐의 scheduling algorithm
  - Process를 상위 큐로 보내는 기준
  - Process를 하위 큐로 내쫓는 기준
  - 프로세스가 CPU 서비스를 받으려 할 때 들어갈 큐를 결정하는 기준

ex) CPU burst가 낮은 프로세스는 `quantum = 8` queue에 들어갔다가 바로 작업을 마무리 할 수 있다. 그러나 CPU burst가 높은 프로세스는 아래로 강등된다. `quantum = 16` 으로 할당시간은 늘어났지만 priority는 낮아진다. ➡️ RR 스케줄링 방법으로는 온전하지 않기 때문에 사용!

- **[장점]**
  - 미리 예측이 필요하지 않는다.
  - 짧은 burst 프로세서가 우대 받음.

3. Multi-Processor Scheduling

- CPU가 여러 개인 경우 스케줄링은 더욱 복잡해진다.

1. **Homegeneous processor**인 경우
    - Queue에 한줄로 세워서 각 프로세스를 가져갈 수 있다.
    - 반드시 특정 프로세서에서 수행되어야 하는 프로세스가 있는 경우에는 문제가 복잡해진다.
2. **Load sharing**
    - 일부 프로세서에 job이 몰리지 않도록 부하를 적절히 공유하는 메커니즘 필요
    - 별개의 큐를 두는 방법 vs 공동 큐를 사용하는 방법
3. **Symmetric Multiprocessing(SMP)**
    - 각 프로세서가 각자 알아서 스케줄링 결정
4. **Astmmetric multiprocessing**
    - 하나의 프로세서가 시스템 테이터의 접근과 공유를 책임지고 나머지 프로세서는 거기에 따름

4. Real-time Scheduling

- 데드라인이 보장되도록 보장한다.

1. **Hard real-time systems**
    a. 정해진 시간 안에 반드시 끝내도록 스케줄링해야 함
2. **soft real-time systems**
    a. 일반 프로세스에 비해 높은 priority를 갖도록 해야 함 (ex. 영화 _ 데드라인을 꼭 보장하지는 못함)

- Example of Non-Preemptive SJF

5. Thread Scheduling

1. **Local Scheduling**

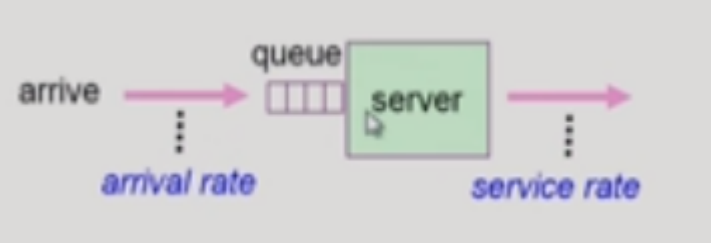
a. 사용자 수준의 thread library에 의해 어떤 thread를 스케줄할지 결정(프로세서가 결정)

2. **Global Scheduling**
   a. 일반 프로세스와 마찬가지로 커널의 단기 스케줄러가 어떤 thread를 스케줄할지 결정

6. Algorithm Evaluation

- 알고리즘 평가 방법

1. Queueing model



- 확률 분포로 주어지는 arrival rate(도착률)와 service rate(처리율) 등을 통해 각종 performane index 값을 계산

2. Implementation(구현) & Measurement(성능 측정)

- 실제 시스템에 알고리즘을 구현하여 실제 작업(workload)에 대해서 성능을 측정 비교

3. Simulation(모의 실험)

- 알고리즘을 모의 프로그램으로 작성 후 trace를 입력으로 하여 결과 비교