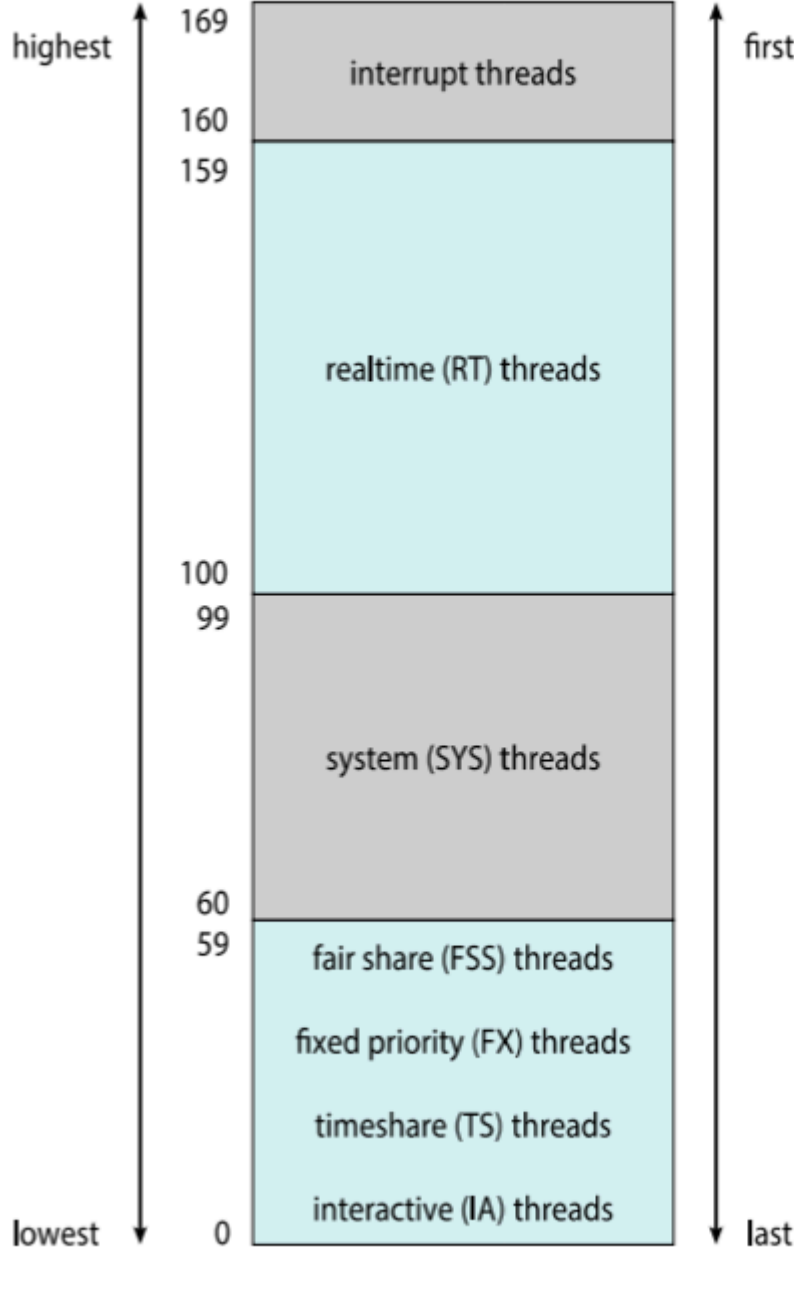


CPU Scheduling example

Solaris

- sun mircoSYSTEM에서 개발한 유닉스 계열 OS. 나중에 Oracle이 인수함
- priority-based scheduling
- 6개의 클래스가 존재
 - Time sharing (TS) : default
 - interactive (IA)
 - real time (RT)
 - system (SYS)
 - fair share (FSS)
 - fixed priority(FP)
- 각 스레드(프로세스?)는 하나의 클래스에만 존재 가능
- 각 class는 has its own scheduling algorithm
- Time sharing은 특별히 multi-level feedback queue임



<Solaris Scheduling>

solaris scheduling

- scheduler는 클래스마다 존재하는 priority를 각 스레드에게 global priority로 mapping
- 가장 높은 우선순위를 가진 thread가 실행
- block이 될 때 까지 실행, time slice만큼만 CPU 사용, 더 높은 thread에게 preempted
- 동일한 priority를 가진 멀티 스레드끼리는 RR 알고리즘 적용

Algorithm Evaluation

- 어떤 CPU scheduling algorithm 할 것인가?
- criteria를 먼저 정하고, 알고리즘을 평가

Deterministic modeling

- 일종의 analytic evaluation
- 임의의 workload를 정하고 각각의 algorithm을 시뮬레이션해서 성능 평가
- minimum average waiting time을 계산해야 함
- 간단하면서도 빠른 방법
- 예제 값을 생각해내는 게 좀 힘들

Queueing model

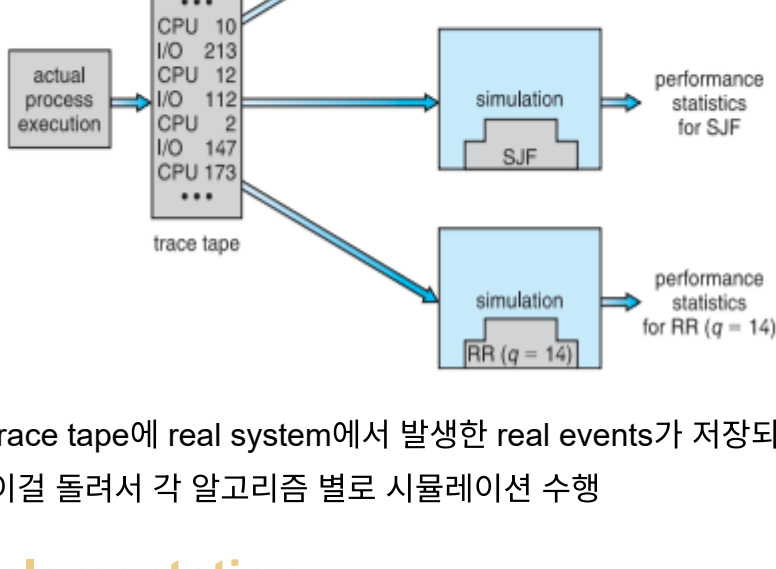
- 프로세스의 도착과, CPU-I/O burst가 상호작용하는 모습을 describe
- 마치 네트워크 서버처럼 환경이 구성 (계속 입력값이 들어오는 모습)
- 지수 분포를 따름 (랜덤이지만 평균)
- utilization, 평균 queue length, 평균 wait time 계산

Littles's formula

- in steady state, 큐를 떠나는 프로세스의 수는 도착하는 프로세스의 수와 같아야 함
- 평균 queue length = 평균 waiting time x 평균 arrival rate
- ex) 7초마다 프로세스가 들어오고 14개의 프로세스가 큐에 있다면 average wait time per process is 2 secs

Simulations

- Queueing model에는 한계가 존재
- 더 정확
- 프로그래밍된 컴퓨터 시스템 모델
- clock이 variable으로 역할



- trace tape에 real system에서 발생한 real events가 저장되어 있음
- 이걸 돌려서 각 알고리즘 별로 시뮬레이션 수행

Implementation

- 시뮬레이션도 limited accuracy
- 그냥 real system에 new scheduler를 implment 해서 test
- high cost, high risk

Example

시험 전에 풀어보깁