1. **다음 운영체제 유형의 필수 속성을 제시해라**
2. **Batch**

I/O routine supplied by the system

시스템에 의해 I/O 루틴이 제공된다.

메모리 관리-시스템은 job들에 메모리를 할당해야 한다.

CPU스케줄링

Memory management – the system must allocate the memory to several jobs

CPU scheduling – the system must choose among several jobs ready to run ν Allocation of devices

1. **Time sharing**

ν The CPU is multiplexed among several jobs that are kept in memory and on disk (the CPU is allocated to a job only if the job is in memory) ν A job swapped in and out of memory to the disk ν On-line communication between the user and the system is provided; when the operating system finishes the execution of one command, it seeks the next “control statement” from the user’s keyboard ν On-line system must be available for users to access data and code

1. **Distributed**

Distribute the computation among several physical processors ν Loosely coupled system – each processor has its own local memory; processors communicate with one another through various communications lines, such as high-speed buses or telephone lines

Clock과 메모리를 공유하지 않음

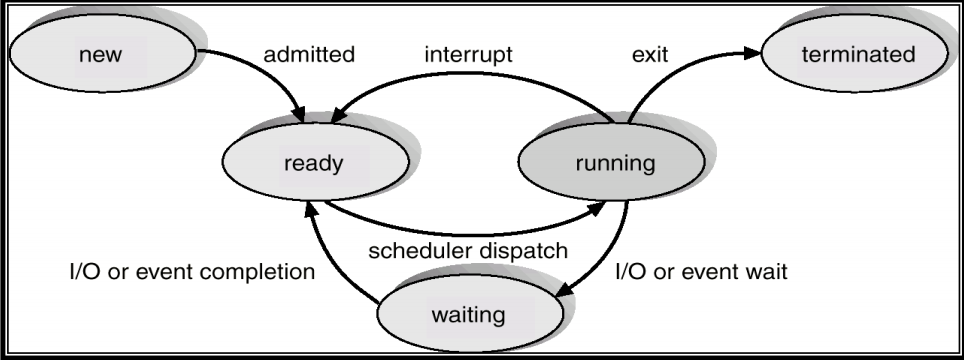
네트워킹 구조가 필요

1. **Parallel**

Multiprocessor systems with more than one CPU in close communication ν Tightly coupled system – processors share memory and a clock; communication usually takes place through the shared memory

Clock과 메모리를 공유

1. **Interrupt에 의해 취해진 동작을 설명해라**

****

프로세스가 시스템 상에서 new 상태에서 할당되기를 기다리다가



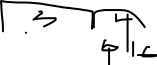
1. **UNIX 에서 fork() 시스템 콜을 설명해라**



1. **평균대기시간을 구해라 using shortest job remaining time first scheduling policy.**



****



**P1: 0—0**

**P2: 2-2**

**P3:**

1. **동기화 프로세스**
2. **상호배제가 concurrent 프로그래밍에서 필요한 이유. Micro instruction level에서 설명해라.**

상호 배제를 보장하지 못하면 임계영역에 둘 이상의 프로세스가 접근하여 logical 에러가 발생할 수 있다.

* Logical error 발생하는 예 보여주기

1. **“test & set(lock)”하드웨어적 방법 설명**
2. **교착상태에 빠진 세마포어 예를 보여라**
3. **아래 소프트웨어 솔루션이 상호배제를 보장하는 안하는지 설명하고 아니라면 왜 안하는지 설명해라.**

**초기 설정: turn=I, flag(f,f)**

****

Flag(t,f)

Flag(t,t)

Turn=j

Turn=i

Pi가 임계영역에 들어감

Pj는 while문에 빠짐

Flag(f,t)

Remainder section으로 pi가 들어감

While문에 pj는 빠져 임계영역에 들어가지 못함

상호배제는 보장하지만 progress 위배

1. **Monitor problem**

**모니터 큐의 역할을 설명해라**

1. **Entry queues**
2. **Condition queues**
3. **Next queue**

**1. Describe the actions taken by an interrupt.**

**2. Describe boots-up.**

**3. Describe DMA.**

**4. block이 버퍼캐시에 들어갈 때 방법 5가지를 수도코드로 작성하라.**

**5. Describe why Mutual Exclusion is needed in concurrent programming enviroment. You have to show detailed example by using micro instruction level.**

**6. Shortest job으로 가정하고 평균 waiting time 계산**

**7. Rate monotonic algorithm을 설명하고 이 세가지가 잘 작동하는가?**

**T1 = 3 C1 = 1 / T2 = 5 C2 = 1 / T3 = 15 C4 = 4**

**8. Describe whether the following software solution to guarantee mutual exclusion is correct or not. If not please explain why it is incorrect?**

**9. fork()와 exec()를 수도코드로 작성하라.**

**10. semaphore의 정의와 deadlock이 일어나는 경우를 설명하시오.**

예시 답안

1. Describe the actions taken by an interrupt.

2. Describe boots-up.

3. Describe DMA.

// 이까지는 알아서. 대부분 5~6단계정도 ➀➁이런 것을 통해서 설명됨

4. block이 버퍼캐시에 들어갈 때 방법 5가지를 수도코드로 작성하라.

// 따로 나눠준 두꺼운 프린트에 5가지 방법 외워두기. ex 뭐 프리리스트에 값이 있으면 어쩌고 저쩌고. 수도코드로 외우면 좋지만 정 안되면 한글로 작성해도 무관함(처음에는 한글 작성으로 깎여있었는데 그 다음에 맞은 거 보면 코드로 작성하는게 좋아보임)

5. Describe why Mutual Exclusion is needed in concurrent programming enviroment. You have to show detailed example by using micro instruction level.

// micro instruction level이라는 말이 조금 중요한데, fetch decode operation store이 4단계가 micro instruction level임.

예시로 t0 t1 t2 t3 t5 t6 t7이렇게 든 부분을 그대로 적으면 되는데 간단하게 살ᄍᆞᆨ 적어보면

t0: P1 fetch x => register X = 10

t1: P1 decoding Add 1

t2: p1 operation x = 11

여기서 만약에 P2라는 것이 fetch를 x로 하면 x = 10이 되어서 값이 바뀐다는 그런 이야기.뒤는 생략. 그래서 결과적으로 f d o s는 최소 단계로 하나의 단계가 되어야함

6. Shortest job으로 가정하고 평균 waiting time 계산

// 이는 알아서

7. Rate monotonic algorithm을 설명하고 이 세가지가 잘 작동하는가?

T1 = 3 C1 = 1 / T2 = 5 C2 = 1 / T3 = 15 C4 = 4

// 그림으로 그리면서 풀어보기. 주기가 짧은 것이 우선 순위가 높다는 것만 생각(T1이 짧으니 이를 바탕으로.)

8. Describe whether the following software solution to guarantee mutual exclusion is correct or not. If not please explain why it is incorrect?

//이건 보기가 없는데,

1 2

3 4

5 6

critical critical -> 동시에 들어갔을 때 deadlock이 걸렸음

1 4

2 5

3 6

이거나

하나 끝나고 하는 경우에는 상관없게 써있으니, 잘 확인할 것

9. fork()와 exec()를 수도코드로 작성하라.

// 이도 마찬가지로 수도 코드로 작성하는게 제일 좋고, 정 안되면 한국어로라도 달달 외워서 가기. fork exec exit 다 외우면 좋지만, 다 못 외울시 가장 많이 나왔던 것이 fork이니 아무래도 이거라도?

10. semaphore의 정의와 deadlock이 일어나는 경우를 설명하시오.

// semaphore의 P(S), V(S)를 그대로 쓸 줄 알아야 함. 물론 하드웨어적 solution이나 mutual exclusion 정의 3가지도 외워주는 것이 좋다고 생각.