

Quiz

- 프로세스가 자기 할 일을 전부 끝마쳤다면 큐 이동 없이 메모리에서 프로세스를 제거한다. 만약, 인터럽트가 올 때까지 못 끝냈으면 중간 과정을 별도로 메모리에 저장하고 대기 큐로 진입한다.
- 하나의 OS에서도 어떠한 일을 하느냐에 따라 스케줄링 알고리즘을 여러 개 쓸 수 있다. 모두 핵심적인 일처리를 담당하는 코어가 존재하면 FIFO로 처리할 수 있을 것이고 우선순위가 다양한 사용자 프로그램들을 처리하는 경우에는 다단계 스케줄링을 쓸 수 있을 것이다.
- 메모리에 적재되는 프로세스가 모두 용량이 다르면서 생기는 문제다. 그렇기에 페이지 형태로 적재하여 외부 단편화 현상을 극복할 필요가 있다.
- 10번째
- .은 현재 디렉터리, ..은 상위 디렉터리
- 하드링크. 하드링크는 원본 파일 데이터를 가리키는 아이노드 번호를 가지고 있다. 그렇기에 원본 파일 데이터가 살아만 있으면 접근이 가능하다.

Test

- CPU에 전력이 들어오면 바이오스 프로그램을 통해 커널이 기록된 위치를 찾는다. 바이오스는 POST를 수행하여 하드웨어 문제점을 찾고 보조기억장치의 MBR을 읽어 부트스트랩을 실행한다. 부트스트랩이 커널의 위치를 찾아 메모리에 적재한다.
- 먼저 파일 시스템을 구성하기 위한 영역을 만들기 위해 파티션을 형성한다. 그리고 포매팅을 통해 파일 시스템 기반을 구축하고 마운트하여 파일 시스템을 구성을 갖춘다. 이후에는 터미널을 통해 텍스트 파일을 만들고 저장한다.
- 파일 이름에 해당하는 디렉터리 엔트리를 찾는다. 그리고 엔트리에서 파일의 아이노드 번호를 얻는다. 아이노드 테이블에서 아이노드 번호를 통해 해당하는 아이노드를 찾는다. 마지막으로 아이노드를 통해 실제 파일이 저장된 위치를 찾아가면 끝이다.
- 메모리에 적재된 페이지를 스와핑을 통해 보조기억장치로 옮긴다. 만약, 프로세스가 너무 커서 보조기억장치로 옮긴 페이지가 많으면 I/O 작업이 빈번해지므로 급격히 컴퓨터의 효율성이 감소한다.
- 모든 페이지가 메모리에 적재되어 있으면 메모리 접근 횟수가 늘어나고 메모리 용량이 부족할 수 있다. 그렇기에 계층적 페이지 테이블을 구성하여 사용성이 많은 페이지는 캐싱하고 사용성이 적은 페이지는 보조기억장치로 보내는 등의 방법을 강구해야 한다.
- 가상머신은 공통적으로 OS를 전부 가상화한다. 커널 영역도 가상화하기에 기반 환경을 설정하는 게 좀 더 자유롭지만 용량을 많이 차지한다. 컨테이너는 커널 영역도 호스트 OS에 의존하고 그 외의 OS 부분을 가상화한다. 그렇기에 용량이 적고 가벼운 장점이 있으나 OS 선택에 제약이 따른다.