

1. 응용프로그램을 작성할 때 표준 라이브러리 함수 (예:printf())를 호출하는 것이 시스템 호출 함수 (예 : write())를 호출하는 것보다 나은 방법인 이유를 설명하라.

표준 자격 자격( printf())가 시스템 호출 자격( write())보다 나은 이유

1. 성능 최적화 (버퍼링)

printf()는

반면에, write()호출될 때마다 시스템 호출을하이브리드 헤드가 증가 할 수 있음

2. 이식성 (Portability)

printf()는stdio.h)의 서론감정에 관계 없이 작동함.

write()는

3. 가독성과 안정성.

printf())\*\*포맷 지정 기능(%d,%s,%f등)\*\*을(를) 영상 코드

write()이를 사용하면 데이터를 직접

4. 추가 기능 지원

printf()는파일 출력( fprintf()), 문자열 저장( sprintf()) 등의 출력 방식),

write()는

5. 표준 출력 지원

printf()기본적으로 `stdoutstdout과 연결되어 있어 \*\*출력 방향방향 전환 및 이동.

write()는 파

구성

✓printf()는성능, 공유성, 가독성, 측면에서write()보printf()당신을 사용하는 것이 더 많다.

3. 컴퓨터 시스템과 운영체제의 목표는 CPU의 활용률을 극대화하는데 있다. 인터럽트는 이 목적을 달성하게 하는데 중요한 역할을 한다. 인터럽트가 어떤식으로 CPU 활용률을 높이는데 기여하는 설명하라.

인터럽트는 CPU 활용률을 높이는 데 중요한 역할을 합니다. 그 이유는 CPU가 작업을 기다리는 시간(예: I/O 작업 대기 시간)을 최소화하고, 유휴 시간을 최대한 줄여주는 방식으로 작동하기 때문입니다. 인터럽트가 CPU 활용도를 높이는 방법은 다음과 같습니다:

**비동기 처리:** 컴퓨터 시스템에서 CPU는 여러 작업을 처리하는데, 이때 대부분의 시간은 입출력(I/O) 장치와의 상호작용에서 대기하게 됩니다. 예를 들어, 디스크에서 데이터를 읽거나 네트워크에서 데이터를 받는 동안 CPU는 할 일이 없어서 대기 상태가 됩니다. 그러나 인터럽트는 이러한 대기 시간 동안 CPU가 다른 작업을 처리할 수 있도록 합니다. 즉, I/O 장치가 처리할 데이터를 준비하면 장치에서 발생한 인터럽트 신호가 CPU를 깨워서 다른 작업을 처리하도록 유도할 수 있습니다.

**효율적인 자원 분배:** CPU는 인터럽트를 통해 다른 작업을 즉시 처리할 수 있기 때문에 자원을 더 효율적으로 분배할 수 있습니다. 예를 들어, CPU는 우선순위가 높은 작업을 먼저 처리하고, 상대적으로 덜 중요한 작업을 나중에 처리할 수 있습니다. 이 과정에서 I/O 작업은 CPU가 처리할 때까지 기다리지 않고, 인터럽트로 CPU의 관심을 끌어 당겨 다른 작업들을 처리하도록 합니다.

**멀티태스킹:** 인터럽트는 시스템에서 멀티태스킹을 가능하게 합니다. 여러 작업들이 동시에 실행될 때, CPU는 한 작업을 처리하다가 인터럽트를 만나면 현재 작업을 일시적으로 중단하고, 다른 중요한 작업을 실행합니다. 이로 인해 CPU는 멈추지 않고 여러 작업을 동시에 처리하는 것처럼 보이게 됩니다. 이를 통해 CPU의 유휴 시간을 최소화하고, 시스템의 전체적인 효율성을 높일 수 있습니다.

**실시간 반응:** 인터럽트는 실시간으로 발생할 수 있기 때문에 시스템이 외부 사건에 즉각적으로 반응할 수 있게 합니다. 예를 들어, 사용자 입력이나 외부 장치의 요청이 있을 때 CPU는 즉시 인터럽트를 받아 해당 작업을 처리하고, 이전 작업을 나중에 처리하는 방식으로 효율성을 극대화합니다.

결론적으로, 인터럽트는 CPU가 유휴 상태에 있는 시간을 줄이고, 다른 작업을 처리할 수 있게 만들어 CPU 활용도를 극대화하는 중요한 기능을

수행합니다.