2. 왜 인터럽트 핸들러의 일부가 어셈블리 언어로 작성되는가?

속도와 효율성

인터럽트 응답 시간은 매우 짧아야 하므로, 고수준 언어보다 어셈블리 언어가 더욱 빠르고 효율적인 처리가 가능하다.

직접 메모리 접근

하드웨어와 직접 상호작용할 때 필요한 메모리 접근 및 특수 명령어의 활용이 가능해진다.

3. 4 개의 프로그램이 동시에 메인 메모리에 상주할 때 낭비되는 CPU 시간 비율

- 각 프로그램은 실행 시간의 50%를 I/O 대기로 보내므로, 동시에 모두 I/O 대기 상태일 확률은
  $(0.5)4=0.0625$ → 약 6.25%
- 즉, CPU 시간의 약 6.25%가 낭비된다

4. 2GB RAM, 256MB OS, 각 프로세스 128MB 일 때 목표 CPU 이용률 90%를 위한 프로세스의 최대 I/O 대기 시간

1. 사용 가능한 메모리:
   전체 RAM − OS 메모리 = 2GB − 256MB
   = 2GB − 0.25GB = 1.75GB = 1792MB
2. 동시에 상주 가능한 프로세스 수:
   N = 1792MB ÷ 128MB = 14
3. CPU 이용률과 I/O 대기 비율 관계:
   각 프로세스가 I/O 대기 상태인 비율을 α ($0 \le \alpha \le 1$)라고 하면,
   전체 CPU 이용률 U 는 다음과 같이 표현된다.
   $U = 1 - (\alpha)^N$
   목표 CPU 이용률이 90%이므로,
   $1 - \alpha^{14} = 0.90$
   $\alpha^{14} = 0.10$
   따라서,
   $\alpha = 0.10^{(1/14)}$
4. 계산:
   $\ln \alpha = (\ln 0.10) / 14$
   $= (-2.302585) / 14 \approx -0.16447$
   $\alpha = e^{(-0.16447)} \approx 0.8485$

즉, 각 프로세스는 전체 실행 시간의 약 **84.85%**까지 I/O 대기할 수 있어야 목표 CPU 이용률 90%를 달성할 수 있다.

➔ 각 프로세스는 전체 실행 시간의 약 84.85%까지 I/O 대기할 수 있다.