1단원. 운영체제 개요와 구조

> 대상: 전공자 · 정보처리기사 취득자 | 목적: 인프라 직무(클라우드·시스템·네트워크·보안) 기술 면접 대비 중요도: ★★★ 최빈출 · ★★ 자주 출제 · ★ 기본 개념

Q1. 운영체제란 무엇이며, 핵심 역할 3가지를 설명하세요. ★★★

답안 운영체제는 하드웨어와 응용 프로그램 사이에서 자원을 관리하는 시스템 소프트웨어입니다. 핵심 역할은 첫째 **자원 관리자**로서 CPU, 메모리, 디스크, I/O 장치를 여러 프로세스에 효율적·공정하게 배분합니다. 둘째 **추상화 제공자**로서 파일, 프로세스, 가상 메모리 같은 추상화를 통해 응용 프로그램이 하드웨어의 복잡함을 몰라도 되게 합니다. 셋째 **보호와 격리**로서 프로세스 간 메모리 침범을 막고, 사용자 모드/커널 모드를 구분해 시스템을 보호합니다.

관련 개념 자원 관리(CPU·메모리·I/O), 추상화(파일, 프로세스, 소켓), 커널, 시스템 콜

Q2. 커널 모드와 사용자 모드를 나누는 이유는 무엇인가요? ★★★

답안 시스템 보호를 위해서입니다. 하드웨어 접근, 메모리 관리 같은 특권 명령을 아무 프로그램이나 실행하면 시스템 전체가 손상될 수 있으므로, CPU의 모드 비트로 권한을 구분합니다. 사용자 프로그램은 사용자 모드에서 실행되다가 특권 작업이 필요하면 **시스템 콜**을 통해 커널 모드로 전환(트랩)하여 커널이 대신 수행합니다. 이중 모드 덕분에 악의적이거나 버그 있는 프로그램이 다른 프로세스나 커널을 직접 훼손하는 것을 막을 수 있습니다.

관련 개념 이중 모드(dual mode), 특권 명령, 모드 비트, 트랩(trap), 시스템 콜

Q3. 시스템 콜(System Call)이란 무엇이고, 호출 과정은 어떻게 되나요? ★★★

답안 시스템 콜은 사용자 프로세스가 커널의 서비스를 요청하는 공식 인터페이스입니다. 과정은 ① 응용 프로그램이 라이브러리 함수(예: glibc의 read())를 호출 → ② 시스템 콜 번호와 인자를 레지스터에 담고 트랩 명령(x86의 syscall) 실행 → ③ CPU가 커널 모드로 전환되어 시스템 콜 핸들러가 번호에 해당하는 커널 함수 수행 → ④ 결과를 반환하며 사용자 모드로 복귀합니다. 모드 전환과 컨텍스트 저장 비용이 있어 일반 함수 호출보다 훨씬 비쌉니다.

관련 개념 트랩, 시스템 콜 테이블, 모드 전환 오버헤드, 대표 시스템 콜(fork, exec, read, write, open)

Q4. 인터럽트란 무엇이며, 폴링과 비교해 설명하세요. ★★★

답안 인터럽트는 I/O 완료, 타이머 만료 등 이벤트 발생 시 하드웨어가 CPU에게 비동기적으로 알리는 메커니즘입니다. CPU는 현재 작업을 중단하고 인터럽트 벡터 테이블을 참조해 해당 핸들러(ISR)를 실행한 뒤 원래 작업으로 복귀합니다. 반면 **폴링**은 CPU가 주기적으로 장치 상태를 검사하는 방식으로, 이벤트가 드물면 CPU를 낭비합니다. 인터럽트는 CPU 낭비가 없지만 컨텍스트 저장/복원 오버헤드가 있어, 고속 네트워크 카드처럼 이벤트가 매우 잦은 환경에서는 오히려 폴링을 혼용(리눅스 NAPI)하기도 합니다.

관련 개념 하드웨어/소프트웨어 인터럽트, 인터럽트 벡터, ISR, 폴링, NAPI

Q5. 커널 구조에서 모놀리식 커널과 마이크로커널의 차이를 설명하세요. ★★

답안 모놀리식 커널은 파일시스템, 드라이버, 네트워크 스택 등 대부분의 OS 기능이 하나의 커널 주소 공간에서 동작합니다. 구성 요소 간 함수 호출로 통신해 성능이 좋지만, 한 모듈의 결함이 커널 전체를 다운시킬 수 있습니다(리눅스, 전통 유닉스). **마이크로커널**은 IPC, 스케줄링, 최소한의 메모리 관리만 커널에 두고 나머지는 사용자 공간 서버로 분리합니다. 안정성과 보안이 좋지만 IPC로 인한 성능 저하가 있습니다(QNX, Minix, L4). 리눅스는 모놀리식이지만 로더블 커널 모듈(LKM)로 확장성을 보완합니다.

관련 개념 모놀리식/마이크로커널/하이브리드(Windows NT, macOS XNU), LKM, IPC 오버헤드

Q6. 멀티프로그래밍, 멀티태스킹(시분할), 멀티프로세싱의 차이는 무엇인가요? ★★

답안 멀티프로그래밍은 메모리에 여러 프로그램을 올려두고, 한 프로그램이 I/O 대기 중일 때 다른 프로그램에 CPU를 주어 이용률을 높이는 개념입니다. **멀티태스킹(시분할)**은 여기에 타이머 인터럽트 기반의 짧은 타임 슬라이스를 더해 여러 작업이 동시에 실행되는 것처럼 보이게 하는 방식으로, 응답성이 목적입니다. **멀티프로세싱**은 물리적으로 CPU(코어)가 여러 개인 시스템에서 실제로 병렬 실행하는 것을 말합니다. 정리하면 각각 'CPU 이용률', '응답성', '물리적 병렬성'에 초점이 있습니다.

관련 개념 CPU 이용률, 타임 슬라이스, 동시성(concurrency) vs 병렬성(parallelism), SMP

Q7. 부팅 과정을 전원 인가부터 로그인까지 설명하세요. ★★

답안 ① 전원 인가 시 CPU가 펌웨어(BIOS/UEFI)를 실행하고 POST로 하드웨어를 점검합니다. ② 펌웨어가 부트 디바이스에서 부트로더(GRUB 등)를 메모리에 적재합니다. ③ 부트로더가 커널 이미지(vmlinuz)와 initramfs를 메모리에 올리고 커널에 제어를 넘깁니다. ④ 커널이 하드웨어 초기화, 드라이버 적재, 루트 파일시스템 마운트를 수행합니다. ⑤ 최초 사용자 프로세스인 init(현대 리눅스는 systemd, PID 1)이 실행되어 서비스들을 기동하고 로그인 프롬프트를 띄웁니다. 인프라 직무에서는 부팅 실패 시 어느 단계 문제인지 구분하는 능력이 중요합니다.

관련 개념 BIOS vs UEFI, POST, GRUB, initramfs, systemd(PID 1), 런레벨/타깃

Q8. 라이브러리 함수 호출과 시스템 콜의 차이는 무엇인가요? ★

답안 라이브러리 함수는 사용자 공간에서 실행되는 코드로 모드 전환이 없어 빠릅니다. 시스템 콜은 커널 모드로의 전환이 필요한 커널 서비스 요청입니다. 다만 둘은 배타적이지 않습니다. 예를 들어 printf()는 라이브러리 함수지만 내부적으로 write() 시스템 콜을 사용하며, 라이브러리가 버퍼링으로 시스템 콜 횟수를 줄여 성능을 높입니다. strcpy()처럼 커널 도움 없이 끝나는 함수는 시스템 콜을 전혀 쓰지 않습니다.

관련 개념 glibc, 사용자 공간 버퍼링, strace(시스템 콜 추적), 모드 전환 비용

Q9. 가상화와 컨테이너에서 OS는 어떤 역할을 하나요? (인프라 연계) ★★★

답안 가상 머신은 하이퍼바이저가 하드웨어를 가상화해 게스트마다 독립된 커널을 실행합니다. Type 1(베어메탈: ESXi, Xen, KVM)과 Type 2(호스트형: VirtualBox)로 나뉩니다. **컨테이너**는 커널을 새로 띄우지 않고 호스트 커널의 기능을 이용해 프로세스를 격리합니다. 리눅스의 **네임스페이스**(PID, 네트워크, 마운트 등 가시성 격리)와 **cgroups**(CPU·메모리 등 자원 사용량 제한)가 핵심입니다. 그래서 컨테이너는 가볍고 빠르지만 호스트와 커널을 공유하므로 격리 수준은 VM보다 낮습니다.

관련 개념 하이퍼바이저 Type 1/2, KVM, 네임스페이스, cgroups, Docker, 커널 공유의 보안 함의
