AI 시대의 운영체제 개론

Chapter 01~13 연습문제 및 실전문제 정답 및 해설

도서출판 엠카잇

이 해설집의 사용 방법

이 해설집은『AI 시대의 운영체제 개론』의 객관식, 단답형·서술형과 실전문제에 일대일로 대응한다. 서술형 답은 핵심 요소를 갖춘 모범 답안이며, 같은 원리를 정확한 용어와 논리로 설명한 답도 인정할 수 있다. 실전문제는 하나의 고정 답보다 분석 과정과 선택 근거를 평가한다.

채점 원칙

객관식은 정답 선택과 개념 구분을 확인한다. 서술형은 핵심 개념의 정확성, 조건과 예외의 제시, 비교 기준의 일관성을 평가한다. 실전문제는 요구사항 충족, 흐름의 정확성, 정상·실패 경로 구분과 선택 근거를 중심으로 평가한다.

장별 구성

Chapter 01 운영체제와 컴퓨터 시스템

Chapter 02 컴퓨터 구조와 운영체제의 동작

Chapter 03 프로세스

Chapter 04 스레드와 프로세스 간 통신

Chapter 05 CPU 스케줄링

Chapter 06 임계구역과 동기화

Chapter 07 교착상태

Chapter 08 메모리 관리와 할당

Chapter 09 페이징과 세그먼테이션

Chapter 10 가상 메모리

Chapter 11 파일 시스템

Chapter 12 저장장치와 입출력 시스템

Chapter 13 가상화·보안과 AI 시대의 운영체제

Chapter 01

객관식 정답과 해설

번호	정답	해설
1	② 중앙처리장치	CPU 는 명령어를 인출·해석하고 산술·논리 연산과 제어를 수행한다. 입력·출력장치는 데이터를 주고받고 보조기억장치는 보관하므로 명령 실행의 중심 장치와 다르다.
2	② 처리 시간과 기억 공간을 필요한 작업에 배분한다.	운영체제는 CPU 시간, 메모리 공간과 장치 사용 순서를 작업에 배분한다. 문서 작성은 응용 프로그램의 일이며, 물리 용량 증가나 모든 작업에 같은 권한을 주는 것은 자원 관리의 정의가 아니다.
3	② 시스템 호출	시스템 호출은 사용자 프로그램이 보호된 커널 기능을 요청하는 공식 인터페이스다. 사용자 인터페이스는 사용자와 프로그램의 접점이고, 저장·출력장치는 커널 기능을 요청하는 인터페이스가 아니다.
4	② 다중 프로그래밍	다중 프로그래밍은 한 작업이 입출력을 기다릴 때 다른 작업을 실행해 CPU 유휴 시간을 줄인다. 일괄 처리는 작업 묶음의 처리 방식이고 단일 작업 처리는 여러 작업을 메모리에 두는 조건과 다르다. 순차 저장은 실행 전환 방식이 아니다.
5	③ 실시간 운영체제	실시간 운영체제는 평균 속도보다 정해진 마감 시간 안의 예측 가능한 응답을 중시한다. 데스크톱·개인용은 사용 편의, 서버는 서비스 제공을 중시하는 분류이며 마감 보장을 뜻하지 않는다.
6	② 디바이스 드라이버	디바이스 드라이버는 특정 하드웨어의 제어 방법을 처리하여 운영체제와 장치를 연결한다. 산술논리장치는 계산 회로이고 메모리와 입력장치는 하드웨어이므로 장치 제어 소프트웨어와 다르다.
7	④ SSD	SSD 는 파일을 오래 보관하는 보조기억장치이다. 일반적인 RAM 은 현재 작업의 내용을 보관한다. CPU 는 처리, 키보드는 입력을 담당하며 일반적인 RAM 은 전원이 끊기면 내용을 잃는다.

번호	정답	해설
8	① 시분할	시분할은 CPU 실행 시간을 짧은 시간 조각으로 나누어 사용자나 프로그램에 번갈아 배분한다. 일괄 처리나 한 작업의 독점은 짧은 시간 조각을 번갈아 배분하지 않으며 입력장치 교체도 시간 배분과 무관하다.
9	③ 임베디드 운영체제	임베디드 운영체제는 가전제품과 산업 장비 등 특정 목적의 장치에서 제한된 자원을 관리한다. 서버·데스크톱·모바일이라는 분류만으로 특정 가전의 제한된 자원에 맞춘 구성을 뜻하지는 않는다.
10	① 보안성	보안성은 허가되지 않은 사용과 접근으로부터 시스템과 데이터를 보호하는 목표이다. 처리량·입력 속도·화면 크기는 접근 허용 여부를 판단하는 보안 목표와 다르다.

단답형·서술형 모범 답안

1. 입력장치, 중앙처리장치, 주기억장치, 보조기억장치와 출력장치의 역할을 각각 설명하시오.

해설 입력장치는 외부 데이터를 받아들이고 CPU 는 명령을 해석·실행한다. 주기억장치는 실행 중인 명령과 데이터를 보관하고, 보조기억장치는 전원이 꺼져도 프로그램과 파일을 유지하며, 출력장치는 처리 결과를 외부에 전달한다.

2. 운영체제를 자원 관리자와 제어 프로그램의 관점에서 각각 설명하시오.

해설 운영체제는 자원 관리자로서 CPU 시간, 메모리와 장치를 배분한다. 제어 프로그램으로서는 프로그램의 실행과 입출력을 감독하고 잘못된 접근이 다른 작업에 영향을 주지 않도록 통제한다.

3. 명령줄 인터페이스와 그래픽 사용자 인터페이스의 차이를 설명하시오.

해설 CLI 는 명령을 문자로 입력하므로 반복·자동화에 유리하다. GUI 는 창, 아이콘과 메뉴를 사용해 상태를 시각적으로 보여 주므로 처음 사용하는 사람이 기능을 찾기 쉽다.

Chapter 01 단답형·서술형 모범 답안 계속

4. 클라이언트-서버 방식과 피투피 방식의 차이를 설명하시오.

해설 클라이언트-서버 방식은 서버가 자원과 서비스를 제공하고 클라이언트가 요청한다. 피투피 방식은 참여 노드가 상황에 따라 요청자와 제공자 역할을 함께 수행한다.

5. 데스크톱, 서버, 모바일과 실시간 운영체제에서 중요하게 보는 목표를 비교하시오.

해설 데스크톱은 상호작용성과 사용 편의, 서버는 처리량·가용성·격리, 모바일은 전력과 이동성, 실시간 운영체제는 마감 시간과 예측 가능성을 특히 중시한다.

실전문제 해설과 평가 기준

온라인 수업 중 자원 관리 분석

1. 키보드·네트워크는 입력, 화면·스피커는 출력, CPU 는 디코딩·편집·프로토콜 처리를 맡는다.

2. 메모리는 실행 코드와 버퍼를, 저장장치는 내려받은 파일과 임시 데이터를 보관한다.

3. 운영체제는 사용자의 요청을 받아 CPU·메모리·장치 사용을 배정하고, 작업이 끝나면 응용 프로그램에 결과를 돌려준다.

4. 여러 프로그램이 동시에 실행될 때 운영체제가 CPU 시간과 장치 사용 순서를 조정한다.
다운로드 종료 뒤 반응 속도가 회복되었다면 자원 경쟁의 감소를 한 가지 가설로 설명할 수 있다.
네트워크, 저장장치와 CPU 사용 추이를 함께 확인하며, 종료 시점만으로 원인을 확정하지 않는다.

권장 배점 개념 정확성 40%, 요구사항 충족 25%, 흐름·계산의 일관성 20%, 선택 근거와 실패 조건 15%

Chapter 02

객관식 정답과 해설

번호	정답	해설
1	④ 파일 버스	데이터·주소·제어 버스가 시스템 버스의 기본 분류이며 파일 버스는 해당하지 않는다. ①~③ 은 각각 데이터, 주소와 제어 신호를 전달하는 실제 기본 분류이다.
2	① 프로그램 카운터	프로그램 카운터는 다음에 실행할 명령어의 주소를 보관한다. 명령어 레지스터는 현재 명령을, 캐시 라인은 명령이나 데이터를 보관한다. 장치 제어기는 입출력장치를 제어한다.
3	② 지역성	시간 지역성과 공간 지역성은 최근 사용한 데이터나 인접 데이터를 다시 사용할 가능성을 뜻한다. 휘발성은 전원 차단 시 내용 보존 여부, 특권성은 실행 권한, 비동기성은 완료 모델에 관한 성질이다.
4	② 예외	0 으로 나누기는 현재 명령의 실행에서 발생한 동기적 예외다. 하드웨어 인터럽트는 외부 사건에 따른 통지이고 폴링은 상태 검사, 스풀링은 작업을 저장해 순서대로 처리하는 방식이다.
5	③ 직접 메모리 접근	DMA 는 CPU 가 매 바이트를 옮기지 않고 제어기가 장치와 메모리 사이의 블록 전송을 담당하게 한다. 파이프라이닝은 명령 단계 중첩, 캐싱은 재사용을 위한 보관, 바쁜 대기는 반복 검사이므로 블록 전송의 담당 주체를 바꾸는 DMA 와 다르다.
6	③ 산술논리장치	산술논리장치는 계산과 비교를 수행한다. 제어장치는 명령에 따라 동작을 지시하고 레지스터는 값을 보관한다. 제어장치는 실행을 지시하고 프로그램 카운터와 명령어 레지스터는 정보를 보관한다.
7	② 서로 다른 명령어의 처리 단계를 겹쳐 수행한다	파이프라이닝은 서로 다른 명령어의 단계를 겹쳐 처리한다. 데이터 의존이나 실행 흐름 변경으로 대기가 생길 수 있다. 단계를 한 번에 합치거나 인출을 생략하는 방식이 아니며 의존 관계가 있으면 대기할 수 있다.

번호	정답	해설
8	④ 공간적 지역성	공간적 지역성은 참조한 위치에 인접한 주소도 곧 참조할 가능성이 높은 성질이다. 인접 주소의 연속 참조는 공간적 지역성의 사례이므로 시간적 지역성만으로 설명할 수 없다. 명령 해석과 실행 모드는 이 분류의 기준이 아니다.
9	③ 예외를 발생시켜 운영체제가 처리한다	특권 명령어는 커널 모드에서만 허용되며 사용자 모드에서 실행하려 하면 예외가 발생한다. 무조건 실행하면 보호가 깨진다. 다른 명령으로 바꾸거나 모드 구분을 없애는 것도 기본 보호 동작이 아니다.
10	① 버퍼링	버퍼링은 송신자와 수신자의 속도나 처리 단위 차이를 임시 저장 공간으로 완화한다. 명령 해석·특권 명령·프로그램 카운터 갱신은 임시 저장 공간으로 장치 간 속도 차이를 완화하는 기법이 아니다.

단답형·서술형 모범 답안

1. 데이터 버스, 주소 버스와 제어 버스가 전달하는 정보를 각각 설명하시오.

해설 데이터 버스는 실제 데이터, 주소 버스는 접근할 메모리·장치의 위치, 제어 버스는 읽기·쓰기·인터럽트 같은 제어 신호를 전달한다.

2. 프로그램 카운터와 명령어 레지스터의 역할을 비교하시오.

해설 프로그램 카운터는 다음에 인출할 명령어 주소를 보관하고 명령어 레지스터는 현재 해석·실행 중인 명령어를 보관한다.

3. 캐시 적중과 캐시 실패가 발생했을 때 데이터의 이동 과정을 설명하시오.

해설 캐시 적중이면 캐시의 데이터가 CPU 에 전달된다. 실패이면 하위 기억장치에서 해당 블록을 가져와 캐시에 채운 뒤 CPU 에 전달하며 필요하면 기존 캐시 라인을 교체한다.

Chapter 02 단답형·서술형 모범 답안 계속

4. 인터럽트와 예외의 발생 원인 및 명령어와의 관계를 비교하시오.

해설 하드웨어 인터럽트는 장치 완료처럼 현재 명령의 실행과 직접 연결되지 않은 비동기적 사건에서 발생한다.예외는 0 으로 나누기, 페이지 부재처럼 현재 명령의 실행과 동기적으로 발생한다.

5. 사용자 모드에서 파일 읽기를 요청했을 때 커널 모드로 전환되는 이유를 설명하시오.

해설 파일 읽기는 장치 제어와 보호된 커널 자료 구조 접근이 필요하다. 사용자 프로그램은 시스템 호출로 커널 모드에 진입하고 운영체제가 권한 검사와 장치 처리를 수행한 뒤 사용자 모드로 돌아온다.

실전문제 해설과 평가 기준

파일 압축과 외장 저장장치 복사가 겹친 상황 분석

1. 압축은 주로 CPU 연산과 메모리 대역폭을 사용하고 복사는 저장장치·버스·DMA 를 사용한다.

2. 폴링은 CPU 가 장치 상태를 반복 확인하지만, 인터럽트 방식은 장치가 완료를 알리므로 CPU 가 그동안 다른 작업을 수행할 수 있다.

3. CPU 가 DMA 전송의 주소·크기·방향을 설정하면 제어기가 메모리와 장치 사이에서 데이터를 옮기고 완료 인터럽트를 보낸다.

4. 버퍼링은 속도 차이를 완화하고 캐싱은 재사용 데이터를 빠르게 제공하며, 동시 부하에서 버스·메모리·장치 큐 경쟁이 반응 속도에 미치는 영향을 설명한다.

권장 배점 개념 정확성 40%, 요구사항 충족 25%, 흐름·계산의 일관성 20%, 선택 근거와 실패 조건 15%

Chapter 03

객관식 정답과 해설

번호	정답	해설
1	③ 하나의 프로그램에서 여러 프로세스가 실행될 수 있다.	프로그램은 저장된 명령어 집합이고 프로세스는 실행 중인 인스턴스이므로 하나의 프로그램에서 여러 프로세스가 만들어질 수 있다. ① 과 ②는 프로그램과 프로세스의 정의를 뒤바꾸었고 ④는 저장된 코드와 실행 인스턴스의 차이를 없앴다.
2	② 준비 상태	준비 상태는 실행 조건을 갖췄지만 CPU 배정을 기다리는 상태다. 생성은 준비 이전, 대기는 사건 완료를 기다리는 상태, 종료는 실행이 끝난 상태이다.
3	① 프로세스 제어 블록	PCB 는 프로세스 상태, 프로그램 카운터, 레지스터와 자원 정보를 보관한다. 명령어 레지스터는 현재 명령, 파일 버퍼는 파일 데이터, 장치 제어기는 장치 상태를 관리한다.
4	② 현재 프로세스의 실행 상태를 저장한다.	문맥 교환 때 현재 실행 상태를 저장하고 다음 프로세스의 상태를 복원한다. 파일 삭제·장치 초기화·식별자 재발급은 문맥 교환마다 필요한 작업이 아니다.
5	③ 좀비 프로세스	좀비 프로세스는 실행을 마쳤지만 부모가 종료 상태를 아직 회수하지 않은 프로세스다. 고아는 부모가 먼저 끝난 경우이고 준비나 입출력 중심 여부는 종료 상태 회수와 다른 기준이다.
6	④ 스택	스택은 함수 호출 순서, 지역 변수와 복귀 위치를 관리한다. 힙은 실행 중 필요한 메모리를 요청해 사용하는 영역이다. 코드는 명령어, 데이터는 전역·정적 변수, 힙은 동적 할당을 위한 영역이다.
7	② fork 는 새 프로세스를 만들고 exec 는 새 프로그램 내용으로 주소 공간을 바꾼다	fork 계열 기능은 기존 프로세스를 바탕으로 새 프로세스를 만들고 exec 계열 기능은 새 프로그램으로 주소 공간을 바꾼다. exec 는 새 부모·자식 관계를 만들지 않으며 fork 는 CPU 사용률 측정이나 프로그램 내용 교체 기능이 아니다.

번호	정답	해설
8	① 준비	입출력 완료 후에는 준비 상태로 이동하여 CPU 배정을 기다린다. 완료 사실만으로 즉시 실행되는 것은 아니다. 입출력이 완료되었으므로 계속 대기할 이유는 없지만 CPU 배정 전 실행 상태로 확정할 수도 없다. 종료 조건도 아니다.
9	③ 자식은 고아 프로세스가 되고 정해진 관리 주체가 관계를 맡는다	고아 프로세스는 부모가 먼저 끝났으나 자식은 계속 실행하는 경우이며 운영체제가 관리 관계를 조정한다. 부모 종료만으로 자식이 종료되거나 좀비가 되는 것은 아니다. 좀비는 자식의 실행이 끝난 뒤 종료 상태가 회수되지 않은 경우이다.
10	④ 반복 측정한 추세와 지속 시간, 응답 상태를 함께 확인한다	일시적인 CPU 사용률 상승만으로 문제를 단정하지 않고 지속 시간과 추세, 응답 상태를 함께 확인한다. 한 번의 수치나 프로세스 이름만으로 원인을 판단하거나 모든 프로세스를 종료하면 정상 작업까지 중단할 수 있다.

단답형·서술형 모범 답안

1. 프로그램과 프로세스의 차이를 한 문장으로 설명하시오.

해설 프로그램은 저장된 명령어와 데이터의 정적 집합이고 프로세스는 그 프로그램을 실행하는 동적 활동이다.

2. 프로세스 주소 공간의 코드, 데이터, 힙과 스택 영역의 역할을 각각 설명하시오.

해설 코드에는 명령어, 데이터 영역에는 전역·정적 데이터, 힙에는 동적 할당 객체, 스택에는 호출 프레임·지역 변수·복귀 주소가 저장된다.

3. 실행 상태의 프로세스가 준비 상태와 대기 상태로 각각 이동하는 경우를 비교하시오.

해설 시간 할당량 종료나 더 높은 우선순위 작업의 등장으로 실행 상태에서 준비 상태로 갈 수 있다. 입출력이나 사건을 기다리면 대기 상태로 가며, 사건이 끝나면 준비 상태로 이동한다.

Chapter 03 단답형·서술형 모범 답안 계속

4. 프로세스 제어 블록과 문맥 교환이 필요한 이유를 설명하시오.

해설 PCB 는 중단한 프로세스를 나중에 재개하기 위한 실행 문맥과 관리 정보를 보관한다. 문맥 교환은 현재 실행 문맥을 해당 PCB 등의 저장 영역에 기록하고 다음 프로세스의 저장된 문맥을 복원하는 과정이다.

5. 유닉스 계열 운영체제에서 좀비 프로세스와 고아 프로세스가 생기는 원인을 비교하시오.

해설 좀비는 자식이 종료됐지만 부모가 종료 상태를 회수하지 않은 경우다. 고아는 부모가 먼저 종료한 살아 있는 자식으로, 유닉스 계열에서는 다른 관리 프로세스가 인계한다.

실전문제 해설과 평가 기준

온라인 수업 환경의 프로세스 상태 분석

1. 실행 중 시간 할당량이 끝나면 준비, 네트워크나 저장장치를 기다리면 대기 상태로 이동한다.
2. 입출력 완료 뒤에는 곧바로 실행이 아니라 준비 큐로 이동한다.
3. 스케줄러가 선택한 작업은 준비에서 실행으로 이동한다. 다른 프로세스에서 이 프로세스로 실행 주체가 바뀌면 문맥 교환이 일어난다. 같은 프로세스가 다시 선택되었다는 이유만으로 프로세스 간 문맥 교환이 반드시 발생하는 것은 아니다.
4. 세 프로세스의 CPU 구간과 입출력 구간을 별도 시간축으로 제시한다.

권장 배점 개념 정확성 40%, 요구사항 충족 25%, 흐름·계산의 일관성 20%, 선택 근거와 실패 조건 15%

Chapter 04

객관식 정답과 해설

번호	정답	해설
1	③ 힙 영역	같은 프로세스의 스레드는 코드·데이터·힙과 열린 자원을 공유하지만 레지스터와 스택은 각자 가진다. 프로그램 카운터와 레지스터는 스레드별 실행 상태이며 각 스레드는 별도의 스택을 가진다.
2	① 병행성	병행성은 단일 코어에서도 작업을 번갈아 진행할 수 있다는 성질이다. 병렬성은 같은 시점의 실행을 뜻한다. 교착상태와 단편화는 작업을 번갈아 진행하는 성질이 아니다.
3	② 스레드 풀	스레드 풀은 미리 만든 작업자가 큐의 요청을 반복 처리하므로 생성 비용을 줄인다. 메시지 큐는 전달 공간, 페이지 테이블은 주소 변환, 장치 큐는 입출력 대상을 관리한다.
4	② 공유 메모리	공유 메모리는 같은 영역을 매핑해 큰 데이터를 반복 복사하지 않고 교환할 수 있다. 시그널은 사건 알림이고 파이프는 바이트 전달 수단이다. 직접 통신은 상대 지정 방식으로 같은 메모리 매핑을 뜻하지 않는다.
5	① 소켓	소켓은 같은 컴퓨터뿐 아니라 네트워크로 연결된 프로세스 사이의 양방향 통신에도 사용된다. 스택·레지스터·프로그램 카운터는 실행 상태를 보관하며 네트워크 통신의 끝점이 아니다.
6	② 커널에서의 블로킹이 전체 사용자 스레드 실행을 막을 수 있다	다대일 모델은 하나의 커널 스레드가 블로킹되면 전체 실행이 막힐 수 있고 여러 코어 활용에도 제약이 있다. 커널 스레드가 하나이므로 사용자 스레드마다 독립 코어를 쓰거나 별도 커널 스레드가 생기는 구조가 아니다.
7	① 한 스레드의 블로킹에도 다른 스레드가 실행될 수 있지만 커널 관리 비용이 든다	일대일 모델은 사용자 스레드마다 커널 스레드를 연결하여 병렬 실행과 개별 블로킹에 유리하지만 관리 비용이 발생한다. 사용자 스레드마다 커널 스레드를 연결하므로 하나만 공유하는 모델과 다르며 레지스터가 합쳐지거나 생성 비용이 사라지지 않는다.

번호	정답	해설
8	④ 간접 통신	간접 통신은 채널이나 우편함을 중간에 두므로 송수신자가 상대의 프로세스 번호를 직접 알 필요가 없다. 직접 통신은 상대를 지정하고 공유 힙이나 스택 공유는 우편함을 통한 메시지 전달 모델이 아니다.
9	③ 시그널	시그널은 짧은 사건 알림에 적합하며 대량의 데이터를 전달하는 수단으로는 적절하지 않다. 공유 메모리·스트림 소켓·파이프는 데이터 전달에 사용할 수 있지만 이 문항의 짧은 사건 통지에 해당하는 수단은 시그널이다.
10	② 요청한 작업이 즉시 실행되며 별도 코어에서 병렬 실행된다는 점	비동기는 요청 후 다른 일을 할 수 있다는 뜻이며 즉시 실행이나 병렬 실행을 보장하지 않는다. 완료와 오류를 별도로 처리해야 한다. ①·③·④ 는 비동기 처리에서 고려할 사항이다. 비동기라는 완료 모델만으로 실제 실행 시점과 코어 배치를 알 수는 없다.

단답형·서술형 모범 답안

1. 프로세스와 스레드의 관계를 한 문장으로 설명하시오.

해설 프로세스는 자원 소유와 보호의 단위이고 스레드는 그 프로세스 안에서 CPU 를 사용하는 실행 흐름이다.

2. 같은 프로세스의 스레드가 공유하는 정보와 독립적으로 갖는 정보를 비교하시오.

해설 스레드는 코드, 데이터, 힙과 열린 파일 등을 공유한다. 프로그램 카운터, 레지스터 집합과 스택은 독립적으로 가진다.

3. 병행성과 병렬성의 차이를 단일 코어와 다중 코어의 관점에서 설명하시오.

해설 병행성은 작업이 겹쳐 진행되는 성질로 단일 코어에서도 번갈아 실행해 구현할 수 있다. 병렬성은 다중 코어에서 둘 이상의 작업이 실제로 동시에 실행되는 성질이다.

4. 공유 메모리와 메시지 전달의 장점과 주의점을 비교하시오.

해설 공유 메모리는 큰 데이터에 효율적이지만 별도 동기화가 필요하다. 메시지 전달은 격리와 분산 통신에 유리하지만 복사·직렬화와 커널 중개 비용이 생길 수 있다.

5. 동기 통신과 비동기 통신을 선택할 때 확인해야 할 조건을 설명하시오.

해설 완료 결과를 호출 흐름이 직접 확인해야 하는지, 완료 통지를 받을 수 있는지, 호출 스레드가 기다려도 되는지, 오류와 시간 초과를 어떻게 처리할지를 확인한다.

실전문제 해설과 평가 기준

온라인 강의 플랫폼의 스레드와 프로세스 통신 설계

1. 오류 격리와 권한 경계가 필요한 재생·다운로드는 프로세스로 나눈다.
2. 한 프로세스 안의 화면·입력·자동 저장은 공유가 많은 스레드로 구성할 수 있다.
3. 대용량 영상은 공유 메모리와 동기화, 제어 메시지는 메시지 큐나 소켓을 선택할 수 있다.
4. 완료 통지·오류·시간 초과·공유 영역 수명을 함께 설계한다.

권장 배점 개념 정확성 40%, 요구사항 충족 25%, 흐름·계산의 일관성 20%, 선택 근거와 실패 조건 15%

Chapter 05

객관식 정답과 해설

번호	정답	해설
1	① 3 과 10	응답=5−2=3, 반환=12−2=10 이다. 응답 시간은 도착부터 최초 실행까지, 반환 시간은 도착부터 완료까지이므로 시각 자체나 실행 시간과 혼동하면 안 된다.
2	② 호위 효과	긴 CPU 작업이 뒤의 짧은 작업과 입출력 중심 작업을 지연시키는 호위 효과이다. 스래싱은 과도한 페이지 부재, 외부 단편화는 흩어진 빈 메모리, 페이지 부재는 비상주 페이지 접근이다.
3	③ 문맥 교환 비용	더 자주 전환하므로 문맥 교환 비용의 비중이 커질 수 있다. 파일 크기·페이지 번호·디스크 패리티는 시간 할당량 축소로 직접 늘어나는 전환 비용이 아니다.
4	① 작업이 실행 행동에 따라 큐를 이동할 수 있다	실행 행동과 대기 시간 등에 따라 작업의 우선순위 큐를 조정한다. 큐 고정은 고정 다단계 큐의 특징이다. 피드백 큐는 선점을 금지하거나 미래 실행 시간을 정확히 아는 방식이 아니다.
5	④ 부하 균형과 캐시 재사용 비용	분산된 부하의 이익과 기존 캐시 상태를 잃는 비용을 함께 비교한다. 이동 시 달라지는 코어 부하와 캐시 상태를 확인해야 하며 파일 이름·암호키·표의 색은 이 비용의 기준이 아니다.
6	③ P2, P3, P1	SJF 는 실행 시간이 짧은 작업부터 선택하므로 2, 4, 6 인 P2, P3, P1 순서로 실행한다. 다른 순서는 실행 시간이 더 긴 작업을 짧은 작업보다 앞에 두므로 이 조건의 SJF 순서가 아니다.
7	① 현재 작업을 선점하고 새 작업을 실행한다	SRTF 는 남은 시간을 비교하므로 3 이 7 보다 짧은 새 작업을 실행한다. 끝까지 실행하는 것은 비선점 방식의 선택이며 종료나 입출력 대기로 보내는 조건은 주어지지 않았다.
8	④ 중기 스케줄러	중기 스케줄러는 메모리 체류를 조절한다. 단기 스케줄러는 곧 CPU 를 사용할 대상을 고른다. 디스패처는 선택된 작업의 실행 전환을 맡고 명령어 레지스터는 현재 명령을 보관한다.

번호	정답	해설
9	② 에이징	에이징은 대기 시간이 길어진 작업의 우선순위를 높여 기아를 완화한다. 호위 효과는 지연 현상이고 비선점 고정이나 할당량 제거만으로 오래 기다린 작업의 우선순위를 높이지 않는다.
10	③ 정해진 마감 시간 안에 작업을 완료할 수 있는가	실시간 스케줄링은 평균 속도보다 정해진 마감 시간의 준수를 중요하게 평가한다. 높은 이용률·같은 시간 배분·짧은 평균 응답만으로 각 작업의 마감 준수를 보장할 수 없다.

단답형·서술형 모범 답안

1. 같은 도착 시각과 실행 시간을 정하고 FCFS, SJF 와 RR 의 평균 대기, 반환과 응답 시간을 계산하시오.

해설 도착 시각 모두 0, CPU 실행 시간을 P1=6, P2=2, P3=4, RR 할당량을 2 로 정한다. 입출력과 문맥 교환 비용은 제외하며 동률이면 번호가 작은 작업을 먼저 선택한다. FCFS 는 P1 0~6, P2 6~8, P3 8~12 로 실행하므로 대기 시간은 0,6,8, 반환 시간은 6,8,12 이다. 평균 대기는 14/3, 평균 반환은 26/3, 평균 최초 응답은 14/3 이다. 비선점 SJF 는 P2 0~2, P3 2~6, P1 6~12 로 실행하므로 번호 순서의 대기는 6,0,2, 반환은 12,2,6 이며 평균은 각각 8/3,20/3,8/3 이다. RR 은 P1 0~2, P2 2~4, P3 4~6, P1 6~8, P3 8~10, P1 10~12 로 실행한다. 완료 시각은 12,4,10 이고 대기는 6,2,6 이므로 평균 대기는 14/3, 반환은 26/3 이다. 최초 실행 시각이 0,2,4 이므로 평균 응답은 2 이다. 반환=완료−도착, 대기=반환−CPU 실행 시간, 응답=최초 실행−도착을 사용한다.

2. 다단계 큐와 다단계 피드백 큐의 차이를 작업의 큐 이동과 기아 방지 정책으로 설명하시오.

해설 다단계 큐는 작업을 고정된 분류의 큐에 배치하며 큐 사이의 이동을 보통 허용하지 않는다. 다단계 피드백 큐는 실행 행동과 대기 이력에 따라 큐를 이동시킨다. 예를 들어 할당량을 모두 쓰면 낮은 우선순위로 이동하고, 오래 기다린 작업에는 에이징이나 주기적 우선순위 부스트를 적용해 기아를 줄인다. 큐 내부 정책과 큐 사이 정책을 구분해야 한다.

3. RR 시간 할당량을 줄일 때 응답 기회와 문맥 교환 비용이 어떻게 변하는지 설명하시오.

해설 짧은 시간 할당량은 작업이 CPU 를 다시 얻을 기회를 더 자주 제공하여 응답성을 높일 수 있다. 반면 문맥 교환 횟수와 스케줄링 비용이 늘며 캐시 재사용에도 영향을 줄 수 있다. 할당량이 지나치게 크면 FCFS 에 가까워진다. 작업의 특성과 실제 전환 비용을 함께 고려해 선택한다.

Chapter 05 단답형·서술형 모범 답안 계속

4. 멀티코어에서 부하 균형과 프로세서 친화성이 충돌할 수 있는 상황을 제시하시오.

해설 한 코어에 작업이 몰렸을 때 다른 코어로 옮기면 부하 균형은 좋아지지만 기존 코어의 캐시와 메모리 접근 이점을 잃을 수 있다. 예를 들어 캐시에 데이터가 많이 남은 작업을 이동하면 새 코어에서 캐시를 다시 채워야 한다. 보충 설명으로, CPU 와 메모리의 위치에 따라 메모리 접근 비

용이 달라지는 비균일 메모리 접근(Non-Uniform Memory Access, NUMA) 환경에서는 데이터가 위치한 메모리 노드도 고려한다. 이동으로 줄어드는 대기 비용과 친화성 손실을 비교해야 한다.

5. 실시간 스케줄링에서 평균 CPU 이용률만으로 마감 보장을 판단할 수 없는 이유를 설명하시오.

해설 평균 CPU 이용률은 특정 작업의 마감까지 발생하는 최악 실행 시간, 도착 패턴, 블로킹과 선점 비용을 보여 주지 않는다. 이용률이 낮아도 짧은 구간에 요청이 집중되거나 높은 우선순위 작업 때문에 마감을 놓칠 수 있다. 비율 단조 스케줄링(Rate Monotonic, RM)이나 최단 마감 우선 스케줄링(Earliest Deadline First, EDF)을 적용할 때는 주기, 마감, 최악 실행 시간과 정책의 가정을 명시하고 그 조건에 맞는 수용 검사를 수행한다.

실전문제 해설과 평가 기준

학사 서버의 스케줄링 정책 설계

1. 짧은 조회에는 라운드 로빈 또는 높은 초기 우선순위를, 긴 통계에는 에이징을 적용한다.
2. 입력은 P1, P2, P3 가 모두 시각 0 에 도착하며 CPU 실행 시간은 6,2,4 이다. RR 할당량은 2 이고 입출력과 문맥 교환 비용은 제외한다. 동률이면 번호가 작은 작업부터 선택한다.
3. FCFS, 비선점 SJF 와 RR 을 같은 입력으로 비교한다. 시간축과 평균 지표는 이 장의 서술형 1 번 풀이를 따른다. SJF 의 평균 대기가 작지만 RR 의 평균 최초 응답은 더 작다는 차이를 해석한다.
4. 응답성 향상과 문맥 교환 비용, 기아 방지의 상충 관계를 설명한다.

권장 배점 개념 정확성 40%, 요구사항 충족 25%, 흐름·계산의 일관성 20%, 선택 근거와 실패 조건 15%

Chapter 06

객관식 정답과 해설

번호	정답	해설
1	② 갱신 손실을 동반한 경쟁 상태	읽기, 증가와 저장 전체가 보호되지 않아 한 번의 증가가 소실되었다. 두 증가가 완료되었으므로 교착상태가 아니다. 페이지 부재나 저장장치 고장도 제시된 갱신 손실의 원인이 아니다.
2	① 메모리 연산의 재배치와 가시성에 관한 가정	증명의 메모리 순서 가정과 실제 언어, CPU 의 보장을 구분해야 한다. 일반 변수에 잠금이나 순서 보장이 자동으로 생기지 않으며 멀티코어 환경을 단일 코어로 가정할 수도 없다.
3	③ 소비자가 mutex 를 얻지 못해 진행이 막힘	소비자가 데이터를 제거해야 빈칸이 생기지만 생산자가 필요한 잠금을 보유한다. 빈칸은 소비자가 제거해야 생긴다. 생산자가 잠금을 놓지 않으면 소비자가 그 동작을 하지 못하며 자동 복제는 해법이 아니다.
4	④ 다른 작업이 상태를 바꾸었거나 허위 깨어남이 있을 수 있기 때문	깨어남 자체는 원하는 공유 상태가 유지된다는 보장이 아니다. 정상적인 깨어남도 가능하고 wait 는 복귀 전에 잠금을 재획득한다. 데이터 크기는 조건 재검사의 이유가 아니다.
5	① 우선순위 상속	잠금 보유자의 실행을 도와 우선순위 역전으로 인한 지연을 줄인다. 페이지 교체·압축은 메모리 기법이며 엄격한 교대는 잠금 보유자의 우선순위를 높이지 않는다.
6	① 진행	아무도 임계구역에 없는데 진입 의사가 없는 작업 때문에 다른 작업이 들어가지 못하면 진행 조건을 위반한다. 이 상황은 동시 진입을 허용한 것이 아니므로 상호 배제 위반과 다르다. 코드 공유와 문맥 저장도 진행 조건이 아니다.
7	④ 값 읽기와 새 값으로 변경을 원자적으로 수행한다	Test-and-Set 은 값을 읽고 새 값으로 바꾸는 동작을 하나의 원자적 연산으로 수행한다. 검사와 변경 사이에 개입을 허용하는 일반 연산이나 읽기 전용 연산이 아니다. 임계구역 전체의 계산을 한 연산으로 만드는 것도 아니다.

번호	정답	해설
8	② CPU 시간을 소비한다	스핀락의 반복 검사는 바쁜 대기이므로 CPU 시간을 소비한다. 예상 보유 시간과 코어 수를 고려해야 한다. 반복 검사는 저장장치 용량만 쓰는 것이 아니며 종료나 데이터 복사를 자동으로 일으키지 않는다.
9	③ 4	계수 세마포어의 초기값은 사용 가능한 자원 수이므로 연결 4 개를 나타내는 값은 4 이다. 0 은 가용 자원이 없음, 1 은 하나만 허용함을 뜻하며 10 은 실제 연결 수를 초과한다.
10	① 항상 계좌 번호가 작은 쪽부터 잠근다	모든 경로가 계좌 번호 순서처럼 같은 전역 순서를 따르면 서로 반대 순서로 잠금을 기다리는 상황을 막을 수 있다. 보내는 계좌는 작업마다 달라질 수 있다. 무작위나 반대 순서는 일관된 전역 순서를 보장하지 않는다.

단답형·서술형 모범 답안

1. 공유 카운터를 두 스레드가 동시에 증가시킬 때 경쟁 상태가 발생하는 과정을 명령 수준에서 설명하시오.

해설 두 스레드가 같은 값을 읽고 각각 증가시킨 뒤 저장하면 두 쓰기 중 하나가 다른 결과를 덮어쓸 수 있다. 읽기·수정·쓰기가 원자적이지 않기 때문에 증가 두 번이 한 번으로 사라진다.

2. 상호 배제, 진행과 한정 대기가 임계구역 해법에 필요한 이유를 각각 설명하시오.

해설 상호 배제는 동시에 둘이 들어가지 못하게 한다. 진행은 임계구역이 비었을 때 진입 결정이 무기한 미뤄지지 않게 하고, 한정 대기는 특정 스레드가 끝없이 양보하지 않도록 기다림의 상한을 요구한다.

3. 원자적 연산, 뮤텍스와 세마포어를 보호 범위·대기 방식·적용 대상의 관점에서 비교하시오.

해설 원자적 연산은 한 변수나 짧은 상태 변경을 중간에 나뉘지 않는 하나의 연산으로 수행하므로 작은 단위의 갱신에 적합하다. 뮤텍스는 소유권을 가진 하나의 스레드가 임계구역을 보호하며, 대기 시간이 길면 보통 스레드를 잠재우는 블로킹 방식을 사용한다. 세마포어는 정수 카운터로 여러 자원의 수나 실행 순서를 표현하며 소유권 없이 신호를 주고받을 수 있다. 매우 짧은 커널 임계구역처럼 대기가 짧을 때는 스핀 방식이 적합할 수 있다.

Chapter 06 단답형·서술형 모범 답안 계속

4. 조건 변수의 wait 연산이 뮤텍스를 원자적으로 해제해야 하는 이유를 설명하시오.

해설 조건 검사는 뮤텍스를 보유한 상태에서 수행한다. 조건이 충족되지 않으면 wait 는 뮤텍스 해제와 대기 등록을 원자적으로 수행하여 해제와 잠들기 사이의 신호 누락을 막는다. 깨어난 뒤 뮤텍스를 다시 획득하고 while 로 조건을 재검사한다.

5. 스핀락과 블로킹 잠금을 대기 시간과 문맥 교환 비용의 관점에서 선택하는 기준을 설명하시오.

해설 예상 대기가 문맥 교환 비용보다 짧고 선점이 제한된 구간이면 스핀락이 유리할 수 있다. 대기가 길거나 CPU 를 다른 작업에 양보해야 하면 블로킹 잠금이 적합하다.

실전문제 해설과 평가 기준

좌석 예매의 동시성 오류 해결

1. 읽기-잔여석 확인-감소-저장을 하나의 임계구역으로 묶어 보호한다.
2. 두 흐름의 읽기와 쓰기가 교차될 때 마지막 좌석이 중복 판매되는 순서를 그린다.
3. 뮤텍스·세마포어·원자적 연산 중 공유 범위와 자원 수에 맞는 방법을 선택한다.
4. 예외 시 잠금 해제, 시간 초과와 재시도 정책을 포함한다.

권장 배점 개념 정확성 40%, 요구사항 충족 25%, 흐름·계산의 일관성 20%, 선택 근거와 실패 조건 15%

Chapter 07

객관식 정답과 해설

번호	정답	해설
1	③ 순환 대기	증가하는 전체 순서로 획득하면 자원 대기의 닫힌 고리를 만들 수 없다. 획득 순서를 정해도 상호 배제와 비선점은 남을 수 있다. 자원의 존재 자체는 교착상태의 필요조건 이름이 아니다.
2	② 추가 자원 수량과 완료 가능성을 검사해야 한다	고리 밖 인스턴스가 반납되면 순환을 가진 작업도 진행할 수 있다. 다중 인스턴스에서는 순환만으로 교착이나 안전을 확정할 수 없으며 순환 자체가 불가능한 것도 아니다.
3	④ (2,2,1)	Need=Max−Allocation 을 각 성분에서 계산한다. 합을 구하거나 일부 성분의 차를 빠뜨리면 남은 요구량이 아니다. (3−1,2−0,2−1) 을 계산해야 한다.
4	③ 최대 요구를 고려한 완료 순서를 보장할 수 없음	안전 순서가 없다는 뜻이며 실제 교착상태와 동일하지 않다. 불안전은 향후 최대 요구를 보장할 순서가 없다는 뜻이지 현재 모두 교착되거나 자원이 고장났다는 뜻이 아니다.
5	① 이전 희생 횟수를 고려한다	이전 피해와 복구 비용을 고려해야 반복 희생을 완화할 수 있다. 같은 작업만 종료하거나 수행량·복구 가능성을 무시하면 반복 희생을 막기 어렵다.
6	④ 3	P1 의 Need 가 Work 이하이므로 완료 가능하다고 가정하고 보유량 Allocation 1 을 돌려주면 Work 는 2+1=3 이다. Work 에는 Need 가 아니라 완료한 작업의 기존 Allocation 을 더한다. 추가로 준 자원도 완료 시 반환되므로 순증가는 1 이다.
7	② 최대 요구량을 넘는 요청이므로 허용 조건을 만족하지 못한다	요청량은 Need 이하이고 Available 이하이어야 한다. Need 를 초과하면 선언한 최대 요구량을 넘으므로 허용 조건을 충족하지 못한다. Available 이 충분해도 선언한 최대 요구를 넘는 요청은 허용 조건을 어긴다. 다른 작업의 Need 를 바꾸어 해결하지 않는다.

번호	정답	해설
8	① 체크포인트는 복구 지점이고 롤백은 그 지점으로 상태를 되돌리는 작업이다	체크포인트는 재시작에 필요한 상태를 저장한 복구 지점이며 롤백은 그 지점으로 되돌리는 동작이다. 복구 지점 저장은 무조건 할당 정책이 아니며 롤백은 전체 동시 종료나 CPU 측정을 뜻하지 않는다.
9	③ 라이브락	라이브락에서는 상태 변화와 반응이 계속되지만 유용한 진행이나 완료가 이루어지지 않는다. 상태가 계속 변하므로 단순 정지 대기와 다르지만 완료하지 못하므로 정상 종료나 안전 순서의 완료가 아니다.
10	④ 우선순위, 수행 시간, 보유 자원과 재시작 비용	희생자 선택에는 우선순위와 수행한 시간, 보유 자원, 남은 시간 및 재시작 비용 등을 비교한다. 이름 길이·생성 시점·화면 순서만으로는 복구 비용이나 피해를 충분히 비교할 수 없다.

단답형·서술형 모범 답안

1. 교착상태와 단순 대기를 자원 반납 가능성과 대기 고리의 관점에서 구분하시오.

해설 단순 대기는 다른 작업이 자원을 반납하면 진행된다. 교착상태는 각 작업이 상대가 가진 자원을 기다리는 닫힌 고리를 이루어 외부 조치 없이는 누구도 반납 단계에 도달하지 못한다.

2. 교착상태의 네 필요조건을 설명하고 각 조건을 차단하는 예방 방법을 한 가지씩 제시하시오.

해설 필요조건은 상호 배제, 점유와 대기, 비선점, 순환 대기다. 공유 가능한 자원으로 바꾸어 상호 배제를 줄이고, 필요한 자원을 한꺼번에 요청하게 해 점유와 대기를 막으며, 새 자원을 얻지 못하면 보유 자원을 반납하게 해 비선점을 깨고, 전역 자원 순서를 정해 순환 대기를 막을 수 있다.

3. 자원 할당 그래프에서 단일 인스턴스와 다중 인스턴스의 순환 해석이 다른 이유를 설명하시오.

해설 자원 인스턴스가 하나라면 순환은 교착상태를 뜻한다. 여러 개이면 순환이 있어도 다른 인스턴스를 얻어 완료할 수 있으므로 현재 할당량과 남은 요청량을 추가로 계산해야 한다.

Chapter 07 단답형·서술형 모범 답안 계속

4. 교착상태 예방, 회피, 탐지·회복의 적용 시점과 비용을 비교하시오.

해설 예방은 할당 규칙으로 필요조건을 차단하고, 회피는 각 요청 때 안전 상태를 검사한다. 탐지·회복은 일단 허용한 뒤 주기적으로 찾아 종료·선점·롤백으로 복구한다.

5. 은행원 알고리즘에서 안전 상태와 불안전 상태를 구분하는 절차를 설명하시오.

해설 Work 를 Available 로 놓고 Need 가 Work 이하인 미완료 프로세스를 찾는다. 완료를 가정해 Allocation 을 Work 에 더하는 과정을 반복했을 때 모든 프로세스를 나열할 수 있으면 안전 상태다. 모든 프로세스를 완료하기 전에 더 이상 선택할 프로세스가 없으면 불안전 상태다.

실전문제 해설과 평가 기준

파일 변환 서비스의 교착상태 대응

1. 프로세스와 파일 잠금·폴더 잠금·장치를 노드로 놓고 요청·할당 간선을 그린다.
2. 상호 배제, 점유와 대기, 비선점, 순환 대기를 각각 확인한다.
3. 전역 잠금 순서나 일괄 요청을 예방안으로 제시한다.
4. 탐지 후 희생자는 보유 자원·진행률·재실행 비용과 기아 이력을 고려해 선택한다.

권장 배점 개념 정확성 40%, 요구사항 충족 25%, 흐름·계산의 일관성 20%, 선택 근거와 실패 조건 15%

Chapter 08

객관식 정답과 해설

번호	정답	해설
1	④ 11,250	범위 검사 후 기준+변위=11,250 이다. 1,250 은 변위, 2,000 은 한계, 10,000 은 기준 주소이므로 각각을 그대로 물리 주소로 사용할 수 없다.
2	② 보호 위반 처리	허용 범위는 0≤a<2,000 이다. 변위가 한계와 같으면 범위 밖이다. 기준을 더하거나 바꾸거나 0 으로 보내는 것은 보호 검사를 대신하지 못한다.
3	③ 200 KB	충분한 구간 중 가장 작은 200 KB 를 고른다. 100 KB 는 부족하고 500·300 KB 는 충분하지만 200 KB 보다 크다.
4	① 외부 단편화	빈 공간이 할당 영역 사이에 흩어져 연속 요청을 만족하지 못한다. 내부 단편화는 할당 구간 안의 남는 공간이며 TLB 적중과 쓰기 시 복사는 빈 공간이 흩어진 현상의 이름이 아니다.
5	④ 폐기한 뒤 필요할 때 파일에서 다시 읽는다	파일에 동일한 내용을 다시 구할 수 있으므로 별도 기록 없이 회수할 수 있다. 원본 파일에서 복원할 수 있으므로 반드시 스왑할 필요는 없다. 파일 삭제나 전체 종료는 페이지 회수 방법이 아니다.
6	② 컴파일, 적재, 실행 시점	주소 바인딩은 기호 주소와 실제 메모리 주소 등을 연결하며 컴파일, 적재 또는 실행 시점에 이루어질 수 있다. 입출력·키보드·종료만으로 시점을 제한하면 컴파일과 적재 및 실행 시점의 주소 연결을 설명하지 못한다.
7	③ 240 KiB	최초 적합은 검색 중 처음 만나는 충분한 공간을 고른다. 90 은 부족하고 다음 240 은 충분하므로 240 KiB 를 선택한다. 90 은 부족하다. 180 과 400 도 충분하지만 검색 순서상 240 을 먼저 만나므로 최초 적합의 선택이 아니다.
8	① 내부 단편화	할당된 구간 안의 남는 24−18=6 KiB 는 내부 단편화이다. 6 KiB 는 할당 구간 내부에 남았으므로 구간 사이의 외부 단편화나 실제 사용 중인 코드가 아니다.

번호	정답	해설
9	④ 같은 크기의 짝 블록이 모두 비어 있다	버디 시스템은 같은 크기의 짝 블록이 모두 비면 병합하여 더 큰 블록으로 되돌린다. 같은 크기라도 짝 관계가 아니면 병합할 수 없다. 사용 중인 블록이나 임의의 크기를 합치는 것도 버디 병합 조건이 아니다.
10	② 저장장치 입출력 비용	전체 주소 공간을 저장장치로 내보내고 다시 읽으면 이동량이 커져 저장장치 입출력 비용이 증가한다. 스와핑은 메모리 내용을 저장장치와 옮기는 작업이며 CPU 명령 수가 0 이 되거나 이름·화면 비용만 늘어나는 것은 아니다.

단답형·서술형 모범 답안

1. 논리 주소와 물리 주소를 구분하고 주소 변환이 필요한 이유를 설명하시오.

해설 논리 주소는 프로그램과 CPU 가 사용하는 주소이고 물리 주소는 메모리 장치에 전달되는 실제 위치다. 재배치, 보호와 각 프로세스의 독립된 주소 공간을 위해 변환이 필요하다.

2. 컴파일 시점, 적재 시점과 실행 시점 주소 바인딩의 차이를 설명하시오.

해설 컴파일 시 바인딩은 위치가 고정돼야 하고, 적재 시 바인딩은 로더가 실제 적재 위치를 반영한다. 실행 시 바인딩은 MMU 가 접근 때마다 변환하므로 실행 중 이동과 가상 메모리를 지원한다.

3. 기준 레지스터와 한계 레지스터가 재배치와 보호를 수행하는 과정을 예로 설명하시오.

해설 논리 주소 d 가 한계 레지스터 값보다 작은지 검사한 뒤 물리 주소(기준 레지스터 값+d)를 만든다. 범위를 벗어나면 예외를 발생시켜 다른 프로세스나 커널 영역 접근을 막는다.

Chapter 08 단답형·서술형 모범 답안 계속

4. 고정 분할과 가변 분할에서 각각 나타나는 단편화를 비교하시오.

해설 고정 분할은 할당된 분할 안의 남는 공간인 내부 단편화가 주로 생긴다. 가변 분할은 빈 공간이 흩어지는 외부 단편화가 생길 수 있다.

5. 최초 적합, 최적 적합과 최악 적합을 탐색 비용과 남는 빈 공간의 관점에서 비교하시오.

해설 최초 적합은 탐색이 빠르고, 최적 적합은 가장 작은 충분 공간을 찾지만 작은 조각을 만들 수 있다. 최악 적합은 가장 큰 공간을 사용해 큰 나머지를 남기지만 탐색과 결과가 항상 유리한 것은 아니다.

실전문제 해설과 평가 기준

사진 편집 프로그램의 메모리 배치

1. 빈 공간과 요청 크기를 수치로 정하고 최초 적합과 최적 적합의 선택 위치를 표시한다.
2. 할당 뒤 남은 구멍을 지도에 표시해 외부 단편화를 비교한다.
3. 내부 단편화는 할당 단위 안의 미사용 공간이라는 점을 구분한다.
4. 빈 공간을 회수·병합하는 비용과 압축 시 프로세스 이동·주소 재배치 비용을 설명한다.

권장 배점 개념 정확성 40%, 요구사항 충족 25%, 흐름·계산의 일관성 20%, 선택 근거와 실패 조건 15%

Chapter 09

객관식 정답과 해설

번호	정답	해설
1	① 3 과 712	13,000=3×4,096+712 이다. 번호와 변위를 뒤바꾸거나 몫·나머지를 잘못 구하면 주소 13,000 을 재구성할 수 없다.
2	③ 37,576	9×4,096+712=37,576 이다. 36,864 는 프레임 시작 주소로 변위가 빠졌고 13,000 은 가상 주소이다. 9,712 는 프레임 번호를 자릿수처럼 붙인 값이다.
3	② 페이지가 물리 메모리에 없다	TLB 는 변환 캐시이므로 실패와 비상주는 다르다. ① 은 실패의 의미이고 ③·④ 는 가능한 후속 동작이다. 페이지가 상주해도 TLB 에 변환이 없을 수 있다.
4	④ 4 MiB	2^20 개 항목에 4 바이트씩 필요하다. 페이지 수는 2^(32−12)=2^20 이므로 다른 보기는 항목 수나 항목당 4 바이트를 잘못 적용한 크기이다.
5	① 경계 밖 접근이다	유효한 변위는 길이 미만이다. 마지막 바이트는 길이−1 이다. 경계 밖 변위를 0 으로 바꾸거나 페이지 크기를 늘려 허용하지 않는다.
6	③ 115 ns	적중 경로는 5+100=105 ns, 실패 경로는 5+100+100=205 ns 이므로 평균은 0.9×105+0.1×205=115 ns 이다. 105 와 205 는 각각 한 경로의 시간이다. 195 는 주어진 적중률을 반대로 적용한 평균이므로 조건과 다르다.
7	① valid 는 유효 주소 공간 여부, present 는 물리 메모리 상주 여부이다	valid 는 해당 페이지가 유효 주소 공간에 속하는지, present 는 현재 물리 메모리에 있는지를 나타낸다. 변경 여부와 실행 권한은 다른 필드이다. 유효 주소라도 비상주일 수 있으므로 valid=1 이 present=1 을 보장하지 않는다.
8	④ 실제로 사용하는 범위의 하위 테이블만 만들 수 있기 때문	다단계 구조는 사용하지 않는 주소 범위의 하위 테이블을 생략하고 필요한 테이블만 만들 수 있다. 주소를 물리 주소와 같게 하거나 변위를 없애는 방식이 아니다. 페이지마다 추가 프레임을 주는 것은 절약의 이유가 아니다.

번호	정답	해설
9	② 각 페이지 테이블 항목이 같은 물리 프레임을 가리킨다	공유 페이지는 여러 프로세스의 페이지 테이블 항목을 같은 물리 프레임으로 연결하여 구현한다. 서로 다른 프레임은 이 매핑 방식의 공유가 아니다. 테이블 삭제나 이름 저장으로 같은 물리 페이지를 연결할 수도 없다.
10	③ 8120	변위 120 은 길이 500 보다 작아 범위 안에 있으며 기준 주소와 더한 8000+120=8120 이 된다. 120 은 변위, 8000 은 기준이고 8500 은 범위 끝이다. 유효 변위를 기준에 더해야 한다.

단답형·서술형 모범 답안

1. 페이지와 프레임의 관계를 설명하고 페이지와 프레임의 크기가 같아야 하는 이유를 쓰시오.

해설 페이지는 논리 주소 공간, 프레임은 물리 메모리의 고정 크기 블록이다. 크기가 같아야 한 페이지를 어느 프레임에든 넣고 변위 값을 그대로 유지할 수 있다.

2. 페이지 크기가 4 KiB 일 때 논리 주소를 페이지 번호와 변위로 나누는 방법을 설명하시오.

해설 4 KiB 는 2 의 12 제곱 바이트이므로 하위 12 비트가 변위이고 나머지 상위 비트가 페이지 번호다.

3. TLB 적중과 실패일 때 물리 주소를 얻는 과정을 각각 설명하시오.

해설 TLB 적중이면 TLB 에서 프레임 번호를 얻어 변위와 결합한다. 실패이면 페이지 테이블을 조회해 변환을 얻고 TLB 를 갱신한 뒤 물리 주소를 만든다.

Chapter 09 단답형·서술형 모범 답안 계속

4. valid, present, 보호 비트와 변경 비트가 나타내는 상태를 구분하시오.

해설 valid 는 주소 공간에 속하는 매핑인지, present 는 현재 물리 메모리에 상주하는지 나타낸다. 보호 비트는 읽기·쓰기·실행 권한을, 변경 비트는 적재 후 수정 여부를 나타낸다.

5. 다단계 페이지 테이블이 단일 선형 페이지 테이블보다 메모리를 절약할 수 있는 이유를 설명하시오.

해설 다단계 구조는 사용하지 않는 큰 주소 구간에 하위 페이지 테이블을 만들지 않는다. 주소 공간이 희소할수록 단일 선형 표보다 표 자체의 메모리를 줄일 수 있다.

실전문제 해설과 평가 기준

주소 변환 추적표 작성

1. 입력은 페이지 크기 4,096 바이트, 페이지 0→프레임 2(읽기/쓰기), 페이지 1→프레임 5(읽기 전용), 페이지 2 는 유효하지만 비상주, 페이지 3→프레임 9(읽기/쓰기), 페이지 4 는 주소 공간 밖이다. 초기 TLB 에는 페이지 0 과 3 이 있다. 각 접근은 독립적으로 검사한다.

2. 1,024 읽기는 페이지 0, 변위 1,024 이며 TLB 적중으로 물리 주소 2×4,096+1,024=9,216 이다. 5,000 읽기는 페이지 1, 변위 904 이며 TLB 실패 후 페이지 테이블을 조회하여 물리 주소 5×4,096+904=21,384 를 얻는다.

3. 5,000 쓰기는 페이지 1 의 읽기 전용 권한 위반이다. 8,300 읽기는 페이지 2, 변위 108 이며 유효한 주소지만 비상주이므로 페이지 부재이다. 적재 뒤 프레임 번호가 주어지지 않았으므로 최종 물리 주소는 정할 수 없다.

4. 13,000 읽기는 페이지 3, 변위 712 이며 TLB 적중으로 물리 주소 9×4,096+712=37,576 이다. 16,500 읽기는 페이지 4, 변위 116 으로 주소 공간 밖 접근이다. 보호 위반과 비상주 부재를 구분하고, 허용되지 않는 접근을 단순 적재로 해결하지 않는다.

권장 배점 개념 정확성 40%, 요구사항 충족 25%, 흐름·계산의 일관성 20%, 선택 근거와 실패 조건 15%

Chapter 10

객관식 정답과 해설

번호	정답	해설
1	② 시간 지역성	시간 지역성은 최근 사용한 위치를 다시 참조하는 경향이다. 공간 지역성은 인접 위치의 참조이고 단편화·스래싱은 같은 위치를 곧 다시 읽는 성질이 아니다.
2	③ 5	초기 1,2,3 과 이후 4,5 에서 부재가 생긴다. 반복된 1 과 2 는 당시 상주하여 적중한다. 최초 세 부재만 세거나 모든 참조를 부재로 세면 실제 FIFO 흐름과 다르다.
3	④ LRU 가 모든 참조열에서 FIFO 보다 우수한 것은 아님	교체 정책의 우열은 입력에 따라 달라질 수 있다. LRU 는 과거 사용을 이용하며 미래를 알지 못한다. 한 참조열의 결과로 계산 오류나 FIFO 의 항상 최적임을 단정할 수 없다.
4	① 동시에 활성화할 작업 수를 줄인다	작업 집합을 수용할 수 있도록 메모리 압력을 낮춘다. 작업 추가는 메모리 압력을 키울 수 있다. 부재를 무시하거나 캐시를 지워도 부족한 작업 집합 수용 공간이 확보되는 것은 아니다.
5	② 새 프레임에 복사 후 쓰기 허용	허용된 사적 쓰기이면 복제와 매핑 갱신으로 처리할 수 있다. 허용된 사적 쓰기는 정상 복제로 처리할 수 있으므로 항상 종료할 필요가 없다. 파일 삭제나 권한 검사 생략도 필요하지 않다.
6	② Belady 이상	FIFO 는 프레임 수 증가에도 페이지 부재가 늘어나는 Belady 이상을 보일 수 있다. 모든 참조열에서 발생하는 것은 아니다. 프레임 증가에 따른 부재 증가이지 인접 참조나 사본 생성이 아니다. 모든 참조열에서 일어나는 규칙도 아니다.
7	④ A	LRU 는 가장 오랫동안 사용하지 않은 페이지를 고르므로 제시된 순서에서 A 를 교체한다. C 와 B 는 A 보다 최근에 쓰였고 D 는 새로 들어올 페이지이므로 기존 희생 대상이 아니다.

번호	정답	해설
8	① {2, 3, 5}	작업 집합은 관찰 구간에서 참조한 서로 다른 페이지의 집합이므로 {2, 3, 5}이다. 작업 집합은 중복을 제거한 집합이다. 5 나 다른 참조 페이지를 빼면 관찰 구간을 빠짐없이 반영하지 못한다.
9	③ 현재 프로세스에 할당된 프레임	지역 교체는 현재 프로세스에 할당된 프레임 안에서만 희생 페이지를 고른다. 다른 프로세스까지 선택하는 것은 전역 교체이다. 파일이나 미할당 프레임은 현재 프로세스의 희생 페이지가 아니다.
10	② RAM 안에 압축 보관해 반출을 늦출 수 있지만 압축과 해제에 CPU 시간이 든다	압축 메모리는 덜 쓰는 페이지를 RAM 안에 압축하여 저장장치 반출을 늦추지만 압축과 해제 비용이 든다. 압축과 해제에는 CPU 작업이 필요하고 모든 페이지를 즉시 반출하는 방식이 아니다. 압축 내용은 해제하여 다시 접근한다.

단답형·서술형 모범 답안

1. 요구 페이징이 프로그램 시작 시간과 메모리 사용량에 주는 장점과 비용을 설명하시오.

해설 필요한 페이지만 마련하므로 시작 시간과 초기 물리 메모리 사용량을 줄일 수 있다. 첫 참조에서는 페이지 부재 처리 비용이 발생할 수 있다. 파일 내용을 읽어야 하면 저장장치 지연이 들지만, 0 초기화나 쓰기 시 복사는 RAM 안에서 처리될 수 있으므로 모든 부재를 저장장치 읽기로 보아서는 안 된다.

2. 페이지 부재가 발생한 뒤 중단된 명령을 재시도하기까지의 과정을 순서대로 설명하시오.

해설 메모리 접근 예외 뒤 커널이 주소와 권한을 검사한다. 허용되지 않는 접근이면 오류로 처리한다. 허용되는 접근이면 프레임을 확보하고 필요에 따라 저장장치 적재, 0 초기화 또는 쓰기 시 복사를 수행한다. 수정된 희생 페이지를 내보낼 때는 필요한 기록을 먼저 처리한다. 페이지 테이블과 관련 TLB 정보를 갱신하고 중단된 명령을 재시도한다.

3. FIFO, LRU 와 최적 페이지 교체를 미래 참조 정보와 구현 가능성의 관점에서 비교하시오.

해설 최적 알고리즘은 가장 늦게 다시 사용할 페이지를 골라 이론적 하한을 주지만 미래를 알아야 한다. FIFO 는 구현이 단순하지만 Belady 이상이 가능하고, LRU 는 최근 과거를 이용해 지역성을 근사하나 정확한 구현 비용이 크다.

Chapter 10 단답형·서술형 모범 답안 계속

4. 스래싱이 발생하는 원인을 작업 집합과 페이지 부재 빈도를 이용해 설명하시오.

해설 실행 중인 작업의 작업 집합 합이 물리 프레임보다 커지면 페이지가 재사용되기 전에 교체되어 부재가 급증한다. 운영체제는 부재 빈도나 작업 집합을 관찰해 프레임을 늘리거나 일부 작업을 줄일 수 있다.

5. 대용량 페이지와 압축 메모리가 해결하려는 문제와 각각의 비용을 비교하시오.

해설 대용량 페이지는 TLB 와 페이지 테이블 부담을 줄이지만 내부 단편화와 연속 메모리 확보 비용이 있다. 압축 메모리는 저장장치 반출을 늦추지만 압축·해제에 CPU 시간을 사용한다.

실전문제 해설과 평가 기준

페이지 교체 시뮬레이션

1. 프레임은 처음에 비어 있으며 3 개이다. 참조열은 1,2,3,1,4,2,5 이고 최초 적재도 부재로 센다. 각 참조 뒤의 상태와 적중 또는 부재를 기록한다.

2. FIFO 는 처음 1,2,3 을 적재하고 1 의 재참조로 적재 순서를 바꾸지 않는다. 4 에서 1 을 내보내고 2 는 적중한다. 5 에서 2 를 내보내므로 부재 5 회, 적중 2 회이다.

3. LRU 는 1 의 재참조 뒤 2 가 가장 오래 미사용된 페이지가 된다. 4 에서 2, 다음 2 에서 3, 5 에서 1 을 내보내므로 부재 6 회, 적중 1 회이다. OPT 는 4 를 참조할 때 이후 다시 쓰일 2 를 남기고, 동률 처리 규칙에 따라 1 번 슬롯의 1 을 내보낸다. 2 는 적중하며, 마지막 5 를 참조할 때도 1 번 슬롯의 4 를 교체한다. 따라서 최종 슬롯 상태는 왼쪽부터 5, 2, 3 이고, 페이지 부재는 5 회, 적중은 2 회이다.

4. 이 참조열의 결과만으로 벨레이디 이상을 확인했다고 결론 내릴 수 없다. 이상을 확인하려면 같은 참조열에서 프레임 수를 늘린 결과와 비교해야 한다. LRU 가 모든 입력에서 FIFO 보다 적은 부재를 보장하지 않는다는 점도 설명한다.

권장 배점 개념 정확성 40%, 요구사항 충족 25%, 흐름·계산의 일관성 20%, 선택 근거와 실패 조건 15%

Chapter 11

객관식 정답과 해설

번호	정답	해설
1	② 다른 이름으로 파일에 접근 가능	다른 디렉터리 항목이 같은 파일을 계속 가리킨다. 남은 하드 링크는 같은 파일을 가리킨다. 링크 하나 삭제만으로 전체 데이터 삭제·링크 종류 변경·오프셋 초기화가 일어나지 않는다.
2	① 디스크립터는 열린 파일 참조를 나타낸다	프로그램은 열린 파일의 상태를 디스크립터나 핸들로 참조한다. 디스크립터는 확장자나 영속 식별자가 아니다. 이름 제거와 열린 참조 해제는 서로 다른 동작이다.
3	③ 앞선 연결을 따라가는 비용	연결을 순서대로 추적해야 할 수 있다. 연결 할당은 블록 연결을 따라가므로 한 번의 주소 계산이나 RAM 색인 존재를 보장하지 않는다. 물리 연속 배치도 전제가 아니다.
4	④ 최신 사용자 데이터 전체의 내구성	사용자 데이터 보존은 기록 방식과 동기화 정책도 필요하다. ①~③ 은 메타데이터 저널이 지원하는 기능이다. 사용자 데이터 전체의 최신 보존은 별도의 기록 정책이 필요하다.
5	② 내구성이 필요하면 동기화와 오류 확인을 검토한다	쓰기의 수락과 영속 기록 완료를 구분한다. write 나 close 의 성공만으로 모든 장치의 전원 장애 후 보존을 보장할 수 없고 메모리 캐시도 일반적으로 영속 저장장치와 다르다.
6	③ 대상 파일의 경로	심볼릭 링크는 대상 경로를 내용으로 저장한 별도 파일이며 대상이 이동하거나 삭제되면 끊어질 수 있다. 심볼릭 링크는 대상 내용 전체의 복제본이 아니다. CPU 사용률과 프로그램 카운터도 경로 정보가 아니다.
7	① 파일 이름을 제외한 메타데이터와 블록 포인터를 담는다	아이노드는 파일 이름을 제외한 속성과 데이터 블록 포인터를 담는 메타데이터 구조이다. 파일 이름은 디렉터리 항목에서 연결한다. 아이노드는 다음 명령 주소나 파일 내용 전체를 보관하는 구조가 아니다.

번호	정답	해설
8	④ 색인 할당	색인 할당은 데이터 블록 주소들을 색인 블록에 모아 원하는 블록을 직접 찾도록 한다. 연결 할당은 다음 블록 연결, 연속 할당은 시작과 길이를 기록한다. 최초 적합은 메모리 빈 공간 선택 정책이다.
9	② 같은 위치에서 연속성을 유지하며 크기 증가	연속 할당은 이어진 블록을 필요로 하므로 뒤 공간을 사용할 수 없으면 같은 위치에서 파일을 확장하기 어렵다. 뒤 블록의 점유는 이름·속성 조회나 이미 확보한 첫 블록 읽기를 직접 막지는 않는다.
10	③ 현재 작업 디렉터리	상대 경로는 현재 작업 디렉터리를 기준으로 해석하므로 실행 위치가 바뀌면 가리키는 대상도 달라질 수 있다. 글꼴·CPU 할당량·페이지 크기는 상대 경로 해석의 기준 디렉터리를 결정하지 않는다.

단답형·서술형 모범 답안

1. 파일 디스크립터, 시스템 열린 파일 항목과 아이노드의 연결 관계를 설명하시오.

해설 프로세스의 파일 디스크립터는 시스템 열린 파일 항목을 가리키고, 열린 파일 항목은 오프셋과 상태를 보관하며 아이노드 또는 파일 제어 블록의 메타데이터에 연결된다. 같은 열린 파일 항목을 공유하면 파일 오프셋도 공유된다.

2. 순차 접근과 직접 접근에 적합한 데이터 사용 사례를 하나씩 제시하시오.

해설 로그처럼 앞에서부터 처리하는 데이터에는 순차 접근이 알맞다. 데이터베이스 파일이나 영상의 특정 위치를 바로 읽는 작업에는 직접 접근이 유리하다.

3. 하드 링크와 심볼릭 링크가 대상 파일 삭제 뒤 서로 다르게 동작하는 이유를 설명하시오.

해설 하드 링크는 같은 파일 객체를 가리키는 또 하나의 디렉터리 항목이므로 원래 이름을 지워도 남는다. 심볼릭 링크는 대상 경로를 내용으로 가진 별도 파일이므로 대상이 없어지면 끊어진다.

Chapter 11 단답형·서술형 모범 답안 계속

4. 연속 할당, 연결 할당과 색인 할당의 직접 접근 성능과 단편화 특성을 비교하시오.

해설 연속 할당은 직접 접근이 빠르지만 외부 단편화와 성장 문제가 있다. 연결 할당은 성장과 순차 접근에 유리하지만 직접 접근이 느리고, 색인 할당은 직접 접근을 지원하지만 색인 블록 비용이 든다.

5. 저널링과 백업이 해결하는 장애 범위가 다른 이유를 설명하시오.

해설 저널링은 파일 시스템의 구조적 일관성을 빠르게 회복하는 기법이다. 백업은 삭제, 장치 고장, 손상과 과거 버전 복구를 위해 별도의 시점과 장소에 데이터를 보존한다.

실전문제 해설과 평가 기준

공유 자료실의 파일 시스템 설계

1. 작은 문서·큰 영상·빈번한 수정 파일의 접근 패턴을 구분한다.
2. 익스텐트 또는 색인 할당과 비트맵 같은 빈 공간 관리 방식을 선택하고 이유를 쓴다.
3. 경로·소유자·권한·공유 정책을 제시한다.
4. 저널링은 일관성 회복, 백업은 삭제·장치 고장 복구라는 차이를 포함한다.

권장 배점 개념 정확성 40%, 요구사항 충족 25%, 흐름·계산의 일관성 20%, 선택 근거와 실패 조건 15%

Chapter 12

객관식 정답과 해설

번호	정답	해설
1	③ HDD	HDD 는 자기 원판과 기계적 헤드를 사용한다. SSD·플래시 USB 는 플래시 메모리를 사용하고 DRAM 은 반도체 메모리이므로 자기 원판의 기계적 접근이 없다.
2	① 100	50→60→80→20→10 으로 합은 100 이다. 138 은 장치 끝까지 이동하는 다른 조건의 값이다. 이 LOOK 경로의 각 이동량 10+20+60+10 을 모두 더해야 한다.
3	② 2 TB	동일 데이터를 복제하므로 한 디스크 분량을 사용할 수 있다. 미러는 두 디스크의 합을 서로 다른 데이터에 쓰지 않으므로 4·8 TB 가 아니다. 용량을 다시 절반으로 줄인 1 TB 도 아니다.
4	④ CPU 가 전송을 설정하고 장치 또는 제어기가 데이터를 옮긴다	설정과 완료 처리에는 CPU 가 관여할 수 있다. DMA 에서도 설정·권한 확인·완료 처리가 필요할 수 있다. 매 바이트를 CPU 가 직접 옮기는 것은 프로그램 기반 전송에 해당한다.
5	① 미러는 삭제도 복제할 수 있기 때문	RAID 는 잘못된 삭제와 악성 변경으로부터 과거 상태를 독립적으로 보존하지 않는다. RAID 도 디스크를 사용하고 RAID 6 은 읽기를 지원한다. 패리티는 복구용 중복 정보이지 암호 키가 아니다.
6	① 9 ms	제시된 모형의 접근 시간은 탐색, 회전 지연, 전송 시간의 합이므로 4+3+2=9 ms 이다. 4 와 7 은 일부 비용을 빠뜨린 값이고 24 는 시간을 곱한 값이다. 이 모형의 세 비용은 더한다.
7	④ SCAN 은 장치 끝까지, LOOK 은 그 방향의 마지막 요청까지만 간다	SCAN 은 장치 끝 위치까지 이동하고 LOOK 은 현재 방향의 마지막 요청 위치에서 되돌아가므로 이동 거리가 달라질 수 있다. LOOK 이 장치 끝까지 가는 것이 아니며 디스크의 방향 전환을 CPU 시간 할당량으로 결정하지 않는다.

번호	정답	해설
8	② 4 TB, 1 개	RAID 5 의 공칭 용량은 (N−1)×S 이므로 4 TB 이고 한 디스크 고장을 견딜 수 있다. ① 은 중복 없는 RAID 0, ③ 은 이 디스크 수의 RAID 6, ④ 는 다섯 디스크 모두를 미러링한 용량과 고장 수에 해당한다.
9	③ 장치 드라이버	장치 드라이버는 운영체제 요청을 장치 제어기 명령으로 변환하는 소프트웨어이다. 원판은 물리 매체, 산술논리장치는 계산 회로, 파일 이름은 식별 정보이므로 요청 변환 소프트웨어가 아니다.
10	① 페이지 단위로 읽고 쓰며 더 큰 블록 단위로 지운다	SSD 플래시는 페이지 단위로 읽고 기록하지만 기존 데이터는 더 큰 블록 단위로 지우는 특성이 있다. 플래시에는 자기 원판이 없다. 한 바이트만 독립적으로 소거하는 모형이나 CPU 명령어와 단위를 같게 보는 모형도 부정확하다.

단답형·서술형 모범 답안

1. HDD 접근 시간을 탐색 시간, 회전 지연과 전송 시간으로 나누어 설명하시오.

해설 탐색 시간은 헤드를 목표 트랙으로 옮기는 시간, 회전 지연은 원하는 섹터가 헤드 아래로 올 때까지 기다리는 시간, 전송 시간은 실제 비트를 읽고 쓰는 시간이다.

2. SSD 의 쓰기와 삭제 단위가 다른 이유와 가비지 컬렉션의 역할을 설명하시오.

해설 플래시는 빈 페이지에는 쓸 수 있지만 기존 데이터를 제자리에서 덮어쓰려면 더 큰 블록을 지워야 한다. 가비지 컬렉션은 유효 데이터를 옮기고 무효 페이지가 많은 블록을 지워 빈 공간을 만든다.

3. FCFS, SSTF 와 SCAN 을 공정성·평균 이동 거리·기아 가능성의 관점에서 비교하시오.

해설 FCFS 는 공정하고 단순하지만 이동 거리가 클 수 있다. SSTF 는 평균 이동을 줄일 수 있으나 먼 요청을 굶길 수 있고, SCAN 은 한 방향 순서로 서비스해 공정성과 이동 거리를 절충한다.

Chapter 12 단답형·서술형 모범 답안 계속

4. RAID 0, 1, 5, 6 과 10 을 가용 용량·쓰기 비용·고장 허용성의 관점에서 비교하시오.

해설 같은 공칭 용량 S 의 디스크를 사용하고 관리 정보의 공간은 제외한다. RAID 0 은 N×S 의 용량을 사용하고 별도의 중복 기록 부담이 없지만 디스크 고장을 견디지 못한다. RAID 1 에서 N 개 디스크가 모두 같은 데이터를 미러링하면 가용 용량은 S 이고, 정상 사본이 하나 남는 범위에서 N−1 개의 고장을 견딘다. 흔한 두 디스크 미러에서는 총 공칭 용량의 절반인 S 를 사용하며 한 디스크 고장을 견딘다. 두 디스크씩 미러 쌍을 만든 전형적인 RAID 10 은 짝수 N 에 대해 (N/2)×S 의 용량을 사용하며, 같은 미러 쌍의 두 디스크가 모두 고장 나지 않으면 여러 쌍에서 발생한 고장을 견딜 수

있다. RAID 5 는 (N−1)×S 와 한 디스크 고장 허용, RAID 6 은 (N−2)×S 와 두 디스크 고장 허용을 제공한다. 미러는 복제 쓰기, RAID 5 와 6 은 패리티 계산과 갱신 비용이 들며, 특히 작은 쓰기에 추가 비용이 발생할 수 있다.

5. CPU, DMA 제어기, 입출력장치와 주기억장치 사이의 DMA 전송 과정을 설명하시오.

해설 CPU 가 원본·목적지·크기·방향을 DMA 제어기에 설정한다. DMA 제어기는 장치와 주기억장치 사이에서 블록을 전송하고 완료 또는 오류 때 CPU 에 인터럽트를 보낸다.

실전문제 해설과 평가 기준

저장 서버의 장치 구성

1. 순차 영상과 임의 문서 접근 비율, 허용 가능한 장치 고장 수를 먼저 정한다.
2. RAID 수준의 가용 용량·쓰기 비용·재구축 위험을 비교하고 별도 백업을 설계한다.
3. CPU 설정→DMA→장치·메모리 전송→완료 인터럽트의 흐름을 그린다.
4. 캐시 기록 완료와 장치의 실제 기록 완료를 구분하고 안전한 제거 절차를 포함한다.

권장 배점 개념 정확성 40%, 요구사항 충족 25%, 흐름·계산의 일관성 20%, 선택 근거와 실패 조건 15%

Chapter 13

객관식 정답과 해설

번호	정답	해설
1	③ 컨테이너는 호스트 커널 공유, VM 은 게스트 커널 실행	실행 환경과 커널 공유의 경계가 다르다. 일반적인 VM 에는 게스트 커널이 있고 컨테이너마다 항상 독립 커널이 있는 것은 아니다. VM 도 프로세스를 실행한다.
2	① 신원 확인, 권한 결정, 요청에 결정 적용	주체 확인과 허용 판단, 실제 집행을 구분한다. 다른 보기는 파일 처리·암호화·자원 실행 동작을 섞은 것으로 신원 확인과 권한 판단 및 집행의 역할이 아니다.
3	② 상대의 실제 완료 여부가 불확실할 수 있다	부분 장애와 응답 손실 때문에 재시도에 중복 방지가 필요하다. 시간 초과는 미수행이나 전체 노드 고장을 확정하지 않는다. 이미 완료된 요청을 재시도하면 중복될 수 있다.
4	④ 약 14 GB	70 억×2=140 억 바이트이며 추가 작업 공간은 별도이다. 70 억 개에 2 바이트를 곱해야 하므로 개수를 작게 해석하거나 원소당 크기를 빼면 7 MB·7 GB 가 된다. 2 TB 도 계산과 맞지 않는다.
5	① 전자는 노드 배치, 후자는 해당 노드의 CPU 실행을 관리한다	작업 배치와 CPU 실행 선택은 계층과 관리 대상이 다르다. 두 계층의 관리 대상이 다르므로 항상 같은 프로그램이 아니며 파일 이름만 관리하지 않는다. GPU 가 있어도 노드 운영체제가 필요하다.
6	④ 제 1 형은 하드웨어 위에서 직접, 제 2 형은 호스트 운영체제 위에서 실행된다	제 1 형은 하드웨어 위에서 직접 가상 머신을 관리하고 제 2 형은 호스트 운영체제 위의 응용 프로그램처럼 실행된다. 제 1 형을 호스트의 응용 프로그램으로 보는 것은 제 2 형과 혼동한 것이다. 하이퍼바이저 유형은 컨테이너의 커널 공유 분류가 아니다.
7	② 제어 그룹	제어 그룹은 CPU, 메모리와 입출력 사용량을 계측하고 제한한다. 네임스페이스는 자원이 보이는 관점을 분리한다. 프로그램 카운터는 실행 주소, 심볼릭 링크는 경로 참조, 확장자는 이름의 일부로 자원 계측·제한 기능이 아니다.

번호	정답	해설
8	③ 최소 권한 원칙	최소 권한 원칙은 작업에 필요한 권한만 제공하여 침해나 오용의 범위를 제한한다. 관리자 권한을 모두에게 주거나 검사·키 보호를 없애면 필요한 범위로 권한을 제한하지 못한다.
9	① 측정 서비스	측정 서비스는 자원 사용량을 측정하고 관리하며 그 정보를 이용자와 제공자에게 제공하는 성질이다. 주소 바인딩은 주소 연결, 시간 할당량은 CPU 배분 단위, 심볼릭 링크는 경로 참조이므로 클라우드 사용량 계측의 특성이 아니다.
10	④ 메모리 용량과 대역폭, 저장장치 및 네트워크 입출력	AI 작업은 가중치와 데이터, 중간 활성값의 이동이 많아 메모리 용량과 대역폭, 저장장치와 네트워크도 병목이 될 수 있다. 이름 길이·글꼴·확장자만으로 연산 장치에 데이터를 공급하는 속도와 작업 메모리 부족을 판단할 수 없다.

단답형·서술형 모범 답안

1. 제 1 형과 제 2 형 하이퍼바이저를 실행 위치, 신뢰 경계와 관리 편의성의 관점에서 비교하시오.

해설 제 1 형은 하드웨어 위의 특권 계층에서 VM 을 관리하고, 제 2 형은 호스트 운영체제 위에서 실행되며 호스트의 드라이버와 관리 도구를 활용한다. 제 2 형은 호스트 계층의 영향을 함께 받지만, 유형만으로 신뢰 경계의 크기나 보안 수준을 단정할 수는 없다. 실제 경계는 하이퍼바이저, 드라이버와 관리 구성 요소의 구현에 따라 달라진다.

2. 가상 머신과 컨테이너의 격리 경계, 시작 비용과 커널 공유 여부를 비교하시오.

해설 VM 은 독립 게스트 커널을 사용해 서로 다른 운영체제를 실행하며, 컨테이너는 보통 호스트 커널을 공유한다. 일반적으로 컨테이너는 시작과 메모리 부담이 작지만, 실제 비용과 격리 강도는 이미지, 실행 환경과 보안 설정에 따라 달라진다. 컨테이너는 공유 커널의 취약점 영향을 함께 받을 수 있으며, VM 도 가상화 계층의 취약점에서 자유로운 것은 아니다.

3. 클라우드 컴퓨팅의 다섯 가지 필수 특성을 설명하시오.

해설 주문형 셀프서비스는 이용자가 제공자의 직원에게 일일이 요청하지 않고 필요한 컴퓨팅 자원을 직접 마련하는 성질이다. 광범위한 네트워크 접근은 표준적인 네트워크 방식으로 다양한 기기에서 서비스를 이용할 수 있다는 뜻이다. 자원 풀링은 여러 이용자의 수요에 맞춰 공용 자원을 배정하고 다시 배정하는 방식이다. 신속한 탄력성은 수요 변화에 따라 자원을 빠르게 늘리거나 줄이는 성질이다. 측정 서비스는 자원 사용량을 측정, 관리하고 그 정보를 이용자와 제공자에게 보여 주는 기능이다.

Chapter 13 단답형·서술형 모범 답안 계속

4. 인증, 인가와 접근 제어가 요청 처리 과정에서 어떤 순서와 역할을 갖는지 설명하시오.

해설 인증은 주체의 신원을 확인하고 인가는 허용할 작업을 결정한다. 접근 제어는 객체 접근 시 그 결정을 실제로 집행하고 기록한다.

5. CPU·GPU·NPU 작업을 운영체제, 장치 드라이버, 가속기 런타임과 클러스터 스케줄러가 어떻게 나누어 관리하는지 설명하시오.

해설 운영체제와 드라이버는 노드의 프로세스·메모리·장치를 관리한다. 가속기 런타임과 프레임워크는 연산 배치와 데이터 이동을 조정하고, 클러스터 스케줄러는 여러 노드의 작업 배치와 재시작을 담당한다.

실전문제 해설과 평가 기준

AI 추론 서비스의 운영체제 설계

1. 가상 머신과 컨테이너의 격리 경계 및 커널 공유 여부를 비교한다.
2. 인증→인가→자원 할당→GPU 추론→결과 반환의 흐름을 작성한다.
3. CPU·GPU·메모리·네트워크의 한도와 스케줄링 정책을 제시한다.
4. 최소 권한, 개인정보 보호, 감사 기록, 실패한 작업의 재시작과 노드 장애 대응을 포함한다.

권장 배점 개념 정확성 40%, 요구사항 충족 25%, 흐름·계산의 일관성 20%, 선택 근거와 실패 조건 15%