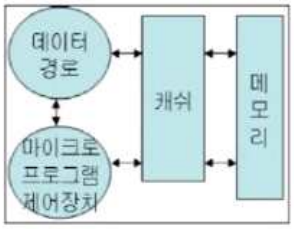
A1 정답 : ④

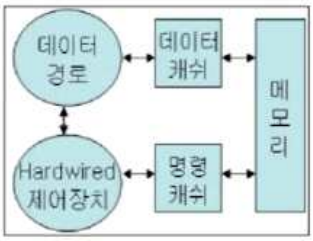
1) CISC (Complex Instruction Set Computer)

- 명령어 하나를 처리하는 구조가 복잡
- 연산 처리시 복잡한 여러 개의 명령어들을 가지고 있는 복합 명령형 프로세서
- 마이크로 프로그램 제어방식을 채택
- 가변 길이 명령어 사용
- 주로 Intel 계열 CPU 구조
- 폰 노이만 아키텍처
- 파이프라인 사용이 어려움
- 단일캐시 공유



2) RISC (Reduced Instruction Set Computer)

- 고정된 길이의 명령어 사용
- 처리속도를 위해 자주 사용되는 몇 가지 명령어 위주로 단순화한 명령축소형 프로세서
- Hardwired 제어 방식 사용
- 데이터 캐시와 명령 캐시의 분리
- 하바드 아키텍처
- 슈퍼 파이프라인, 슈퍼 스칼라 기법 사용
- 주로 모바일, 임베디드 계열 (ARM)



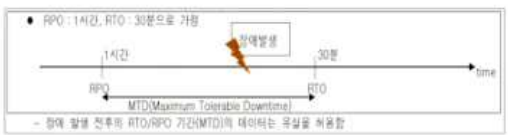
A2 정답 : ②

RPO (Recovery Point Objectives) : 복구 목표 지점

정보시스템 장애 시, 비즈니스 연속을 위해 어느 시점으로 정보시스템을 되돌릴지를 결정하는 지표

재해 발생 이전의 알려진 상태로의 "롤백" 또는 동기화하는 목표 상태를 의미

감내가능한 데이터 유실량 (금융권 RPO 목표는 0)



RTO (Recovery Time Objectives) : 복구 목표 시간

정보시스템 장애 시 시스템을 원상태로 복원하는데 소요되는 시간을 의미

정보시스템의 최대 허용 다운타임과 연관됨

A3

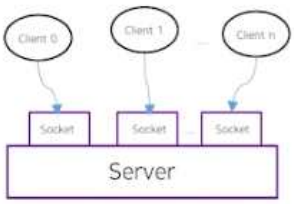
정답 : ② connect() 요청을 받아 들이는 함수는 wait()가 아니라 listen()이다.

가. 서버 소켓(Server Socket)은 처리 과정이 조금 복잡함.

1) 소켓(Socket)을 생성(create)
2) 생성한 소켓에서 사용할 IP 주소와 포트 번호를 결합(bind)
3) 그후 클라이언트로부터 연결 요청이 수신되는지 주시(listen)
4) 요청이 수신되면 요청을 받아들여(accept) 데이터 통신을 위한 소켓을 연결함.
5) 새로운 소켓을 통해 연결이 수립(ESTABLISHED)되면 송수신(send/recv)할 수 있음.
6) 데이터 송수신이 완료되면, 소켓(Socket)을 닫습니다(close).

나. 클라이언트 소켓(Client Socket) 처리과정

1) 소켓(Socket)을 생성(create)
2) 서버 측에 연결(connect)을 요청
3) 서버 소켓에서 연결이 받아들여지면 데이터를 송수신(send/recv)
4) 모든 처리가 완료되면 소켓(Socket)을 닫습니다(close).



서버 소켓은 핸드폰 통화하는 방식으로 얘기할 수 있고 클라이언트는 공중전화해서 휴대폰으로 전화거는 방식으로 생각할 수 있다.

4. 시스템 아키텍처 이해와 활용 단답형 해설 Q4

A4

정답 : 2 또는 2개

1. 코드 설명

코드내에서는 omp_set_num_threads(4);와 #pragma omp parallel num_threads(2)가 쓰레드 갯수를 조절하며, 지정하지 않으면 컴파일러에서 알아서 지정.

루프문 안에서 printf()의 출력결과는 omp_get_thread_num()은 쓰레드 번호를 출력과 index를 출력하고 있음

따라서 omp_get_thread_num()에서 출력되는 0번과 1번이므로 스레드 수는 2개임

5. 시스템 아키텍처 이해와 활용 서술형 해설 Q5

A5

1. 프로세서의 명칭 : SIMD

   배열 프로세서(Array Processor)

2. 제어장치 역할 :

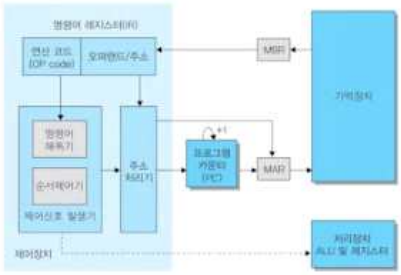
제어장치는 기억장치에서 명령을 읽어 해독

다음에 실행될 명령어 주소를 계산

연산의 수행 순서를 결정

명령어 실행에 필요한 제어 신호를 발생

(정답 키워드 : 해독, 순서결정, 제어신호, 명령주소 계산)

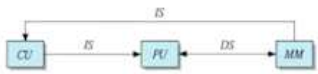


컴퓨터구조(김형근 손진곤 한국방송통신대학교출판문화원)

A5

Flynn의 분류(Flynn's classification) : 구조적 특징에 따른 분류 방식

구분	명령어 흐름	데이터 흐름	사례
SISD	1	1	펜티엄,폰노이만구조
SIMD	1	다중	Array, 슈퍼 컴퓨터
MISD	다중	1	적용어려움
MIMD	다중	다중	SMP, MPP, NUMA

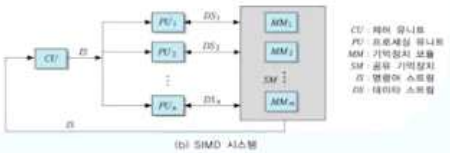


1. 단일 명령어 스트림 - 단일 데이터 스트림 (SISD, Single Instruction Stream - Single Data Stream)

   . SISD의 병렬처리 구현의 예 : 파이프라이닝 구조

2. 단일 명령어 스트림 - 복수 데이터 스트림 (SIMD, Single Instruction Stream, Multiple Data Stream)

- n개의 프로세싱 유닛으로 구성되나, 동작은 하나의 제어 장치(CU)에 의해 통제됨
- 모든 처리장치들은 다 동일 연산을 수행하지만, 각각 서로 다른 데이터들을 처리하는, 데이터 관점의 병렬성(data-level parallelism) 구현



(b) SIMD 시스템