

# 2015-국회직-컴퓨터일반-가형-해설-곽후근

1. 다음 프로그램의 실행 결과는?

```
# include <stdio.h>
int main()
{
    int arr[] = {8, 5, 3, 1, 2, 7, 9};
    int *p = arr+2, a = 0, b = 0;
    a = **p;
    b = (*p)++;
    printf("%d, %d\n", a, b);
    return 0;
}
```

- ① 3, 3
- ② 3, 1
- ③ 1, 1
- ④ 1, 2
- ⑤ 4, 1

해설)

정답 체크 :

(3) arr = &arr[0] = &8; // arr은 첫 번째 배열의 시작 주소를 의미하고 8을 가리킨다.  
p = arr+2 = &arr[2] = &3; // p는 세 번째 배열의 시작 주소를 의미하고 3을 가리킨다.  
a = \*\*p; // p의 주소를 하나 증가(&arr[3])하고 해당 주소의 값(1)을 a에 대입한다.  
b = (\*p)++; // 현재 p의 값(1)을 b에 대입하고, 그 값을 하나 증가(2)한다.

2. 정렬할 데이터 중 하나를 선택해 이를 기준으로 작은 값은 왼쪽에, 큰 값은 오른쪽에 오도록 주어진 데이터를 분할한다. 분할된 두 데이터 집합에 대하여 동일한 방법을 재귀적으로 적용하여 정렬하는 방법은?

- ① 퀵 정렬(quick sort)
- ② 선택 정렬(selection sort)
- ③ 거품 정렬(bubble sort)
- ④ 삽입 정렬(insertion sort)
- ⑤ 분할 정렬(division sort)

해설)

정답 체크 :

(1) 퀵 : 피벗을 이용한다.

오답 체크 :

- (2) 선택 : 숫자중의 가장 큰 수 또는 가장 작은 수를 선택하여 맨 왼쪽 또는 맨 오른쪽 수와 교환한다.
- (3) 거품 : 자신과 오른쪽을 비교하면서 정렬한다.

- (4) 삽입 : 자신을 자신이 있어야할 자리(오름차순 또는 내림차순 상)에 삽입한다.
- (5) 분할 : 분할 후 합병할 때 정렬한다.

3. 다음 중 C 언어 포인터 변수에 대한 설명으로 옳지 않은 것은?

- ① 다른 변수의 주소 값을 저장할 수 있다.
- ② 포인터 변수의 크기는 가리키고 있는 변수의 종류에 따라 달라진다.
- ③ 가리키고 있는 변수 값을 읽기 위해서는 \* 연산자를 사용한다.
- ④ 포인터 변수에 일반 변수의 주소를 대입하기 위해서는 & 연산자를 사용한다.
- ⑤ 포인터 변수를 가리키는 포인터 변수를 선언할 수 있다.

해설)

정답 체크 :

(2) 포인터 변수의 크기 : 포인터 변수의 크기는 고정된다. 32비트 컴퓨터에서는 32비트이고, 64비트 컴퓨터에서는 64비트이다.

오답 체크 :

- (1) 변수의 주소값 : 변수나 포인터의 주소값을 저장할 수 있다.
- (3) 변수값 : 포인터가 가리키는 값을 가져오는 간접 참조 연산자 \*를 사용한다.
- (4) 변수의 주소 : 변수의 주소를 계산하는 주소 연산자 &를 사용한다.
- (5) 포인터 변수 : 이중 포인터 혹은 포인터 배열을 선언할 수 있다. 3중 포인터, 4중 포인터, N중 포인터도 이론적으로는 가능하나 잘 사용되지 않는다.

4. 프로세스의 우선순위 값이 <보기>와 같은 규칙에 따라서 동적으로 변화하는 선점형 우선순위 기반 스케줄링(Preemptive Priority Scheduling) 알고리즘이 있다. 우선순위 값이 클수록 우선순위가 높다고 가정할 때,  $0 < \alpha < \beta$  인 경우 이 알고리즘과 가장 유사하게 동작하는 CPU 스케줄링 알고리즘은?

<p>&lt; 보 기 &gt;</p> <ul style="list-style-type: none"> <li>◦ 프로세스가 ready queue에 있을 동안 우선순위 값은 <math>\alpha</math>의 비율로 변한다.</li> <li>◦ 프로세스가 실행 상태에 있을 동안 우선순위 값은 <math>\beta</math>의 비율로 변한다.</li> <li>◦ 프로세스가 ready queue에 들어가는 순간 우선순위 값은 0이 된다.</li> </ul>
---

- ① FIFO(First In First Out)
- ② LIFO(Last In First Out)
- ③ SJF(Shortest Job First)
- ④ SRTF(Shortest Remaining Time First)
- ⑤ RR(Round-Robin)

해설)

정답 체크 :

(1) FIFO : 프로세스 처리 시간이 지날수록 우선순위가 높아진다.

오답 체크 :

- (2) LIFO : 프로세스 처리 시간이 지날수록 우선순위가 낮아진다.
- (3) SJF : 프로세스 처리 시간이랑 무관하게 각 작업의 실행 시간이 작은 것을 우선으로 한다.
- (4) SRTF : 프로세스 처리 시간이랑 무관하게 각 작업의 남아 있는 시간이 작은 것을 우선으로 한다.

(5) RR : 프로세스 처리 시간이랑 무관하게 각 작업에게 고정된 시간 할당량이 주어진다.

Tip! : 프로세스 처리 시간이 지날수록 우선순위가 높아지는 것을 찾으시면 된다.

5. 다음과 같이 주어진 후위표기방식의 수식을 중위표기방식으로 나타낸 것은?

ABC-/DEF+++

- ①  $A/(B-C)+F+E+D$
- ②  $A/(B-C)+D*(E+F)$
- ③  $A/(B-C)+D+E*F$
- ④  $A/(B-C)*D+E+F$
- ⑤  $(A-B)/C+D+E*F$

해설)

정답 체크 :

(2) 후위를 중위로 바꾸는 것은 스택을 이용하여 풀면 다음과 같다. 피연산자가 나오면 스택에 집어 넣고(push), 연산자가 나오면 스택에서 최상위 두 개를 꺼내(pop) 연산을 수행 후 다시 스택에 집어넣는다(push). 여기서 첫 번째 피연산자와 두 번째 피연산자의 순서에 주의한다.

							F			
		C				E	E	E+F		
	B	B	B-C		D	D	D	D	D*(E+F)	
A	A	A	A	A/(B-C)	A/(B-C)	A/(B-C)	A/(B-C)	A/(B-C)	A/(B-C)	A/(B-C)+D*(E+F)

Tip! : 전위(prefix), 중위(infix), 후위(postfix)를 서로 간에 교환하는 방식은 6가지가 존재한다. 4 가지 시험 문제에 나왔고, 2가지(전위 -> 후위, 중위 -> 전위)가 아직 시험 문제에 나오지 않았다. 자주 나오는 문제이므로 꼭 숙지를 해두어야 한다.

6. 대용량 데이터의 관리를 위해 사용되는 데이터베이스 관리 시스템(DBMS)에 대한 설명으로 옳지 않은 것은?

- ① 트랜잭션 처리 과정에서 데이터의 일관성과 무결성 유지를 위한 기능을 수행한다.
- ② 트랜잭션은 원자성(atomicity)을 가지도록 한다.
- ③ 데이터 무결성 유지를 위해 데이터의 중복을 허용하지 않는다.
- ④ 예상치 못한 시스템 중단으로 시스템이 재가동 될 때, 데이터 무결성이 유지되는 이전의 상태로 복구하는 기능을 수행한다.
- ⑤ 저장된 데이터에 대한 효과적인 접근을 위해 질의어를 지원한다.

해설)

정답 체크 :

(3) 데이터 중복 : 데이터 중복을 최소화하는 것은 무결성 유지를 위해서 하는 것이 아니라 동일한 데이터가 여러 위치에 중복 저장되는 현상을 방지하는 것이다. 데이터가 중복되면, 저장 공간이 낭비 되고 데이터의 일관성이 깨질 수 있다.

오답 체크 :

- (1) 일관성과 무결성 : 트랜잭션이 성공적으로 수행된 후에도 데이터베이스가 일관성 있는 상태를 유지해야 한다.
- (2) 원자성 : 트랜잭션의 연산들이 모두 정상적으로 실행되거나 하나도 실행되지 않아야 하는 all-or-nothing 방식이다.

(4) 이전 상태 복구 : DBMS에서는 롤백(Rollback) 기능을 제공한다.

(5) 질의어 : DBMS에서는 SQL을 지원한다.

7. 다음 트리를 후위 순회(Post-order Traversal)할 경우의 탐색 순서로 옳게 나열된 것은?

① D-E-B-F-C-A

② A-B-D-E-C-F

③ F-C-E-D-B-A

④ D-E-F-B-C-A

⑤ F-C-A-E-B-D

해설)

정답 체크 :

(1) D - E - B - F - C - A : 후위 순회

오답 체크 :

(2) A - B - D - E - C - F : 전위 순회

8. 네트워크를 통한 데이터 전송 시 데이터의 전송 경로를 파악하기 위해 사용하는 도구들(예 : UNIX 계열 운영체제의 traceroute, Windows 운영체제의 tracert 등)은 공통적으로 다음 중 어느 프로토콜에 기반하여 동작하는가?

① TCP

② UDP

③ ICMP

④ ARP

⑤ RTP

해설)

정답 체크 :

(3) ICMP : 인터넷 제어 메시지 프로토콜은 RFC 792에서 정의한 인터넷 프로토콜 모음 중의 하나이다. ICMP 메시지들은 일반적으로 IP 동작에서 진단이나 제어로 사용되거나 오류에 대한 응답으로 만들어진다. 예를 들어, 핑(ping) 유틸리티는 ICMP "에코 요청(Echo request)"과 "에코 응답(Echo reply)" 메시지를 사용해 구현할 수 있다.

오답 체크 :

(1) TCP : 근거리 통신망이나 인트라넷, 인터넷에 연결된 컴퓨터에서 실행되는 프로그램 간에 일련의 옥텟을 안정적으로, 순서대로, 에러없이 교환할 수 있게 한다. 연결 설정을 수행하고, 흐름 제어와 혼잡 제어를 수행한다. TCP는 웹 브라우저들이 월드 와이드 웹에서 서버에 연결할 때 사용되며, 이메일 전송이나 파일 전송에도 사용된다.

(2) UDP : 연결을 설정하지 않고 수신자가 데이터를 받을 준비를 확인하는 단계를 거치지 않고 단방향으로 정보를 전송한다. UDP를 사용하는 애플리케이션에는 도메인 이름 서비스(DNS), IPTV, 음성 인터넷 프로토콜(VoIP), TFTP, IP 터널, 그리고 많은 온라인 게임 등이 있다.

(4) ARP : IP 주소(논리 주소)를 MAC 주소(물리 주소)로 바꿔준다.

(5) RTP : 실시간으로 음성이나 동화상을 송수신하기 위한 전송 계층 통신 규약. RFC 1889에 RTCP (RTP control protocol)와 함께 규정되어 있다. 자원 예약 프로토콜(RSVP)과는 달리 라우터 등의 통신망 기기에 의지하지 않고 단말 간에 실행되는 것이 특징이다. RTP는 보통 사용자 데이

터그램 프로토콜(UDP)의 상위 통신 규약으로 이용된다.

9. 어떤 프로그램에서 부동소수점 곱셈(floating-point multiplication) 연산이 전체 수행시간의 70%를 차지한다고 하자. 해당 프로그램의 성능을 2배 향상시키려면(즉, 전체 수행시간을 1/2로 단축시키려면), 부동 소수점 곱셈 연산의 성능이 몇 배 향상되어야 하는가?

- ① 1. 3배
- ② 2배
- ③ 3배
- ④ 3.5배
- ⑤ 5배

해설)

정답 체크 :

(4) 암달의 법칙은 컴퓨터 시스템의 일부를 개선할 때 전체적으로 얼마만큼의 최대 성능 향상이 있는지 계산하는 데 사용된다. 암달의 법칙에 따르면, 어떤 시스템을 개선하여 전체 작업 중 P%의 부분에서 S배의 성능이 향상되었을 때 전체 시스템에서 최대 성능 향상은 다음과 같다.

$$\frac{1}{(1-P) + \frac{P}{S}}$$

문제의 주어진 조건에서, P는 0.7이고 전체 시스템의 성능 향상은 2가 된다. 이를 수식에 대입하면 다음과 같다.

$$\frac{1}{(1-0.7) + \frac{0.7}{S}} = 2, S = 3.5$$

10. 자식 프로세스를 만들어서 'ls' 프로그램을 수행하도록 하는 아래 C 프로그램에 대한 설명 중 옳지 않은 것은?

```
# include <stdio.h>
void main(int argc, char * argv [])
{
    int pid;
    pid = fork();
    if ( ⓐ ) {
        fprintf(stderr, "Fork Failed");
        exit(-1);
    }
    else if ( ⓑ ) execlp("/bin/ls", "ls", NULL);
    else {
        wait(NULL);
        printf("Child Complete");
        exit(0);
    }
}
```

}

- ① ㉠에 들어갈 조건은  $pid < 0$ 이다.
- ② ㉡에 들어갈 조건은  $pid == 0$ 이다.
- ③ 부모 프로세스는 자식 프로세스가 종료한 후에만 종료한다.
- ④ fork가 정상적으로 수행 된다고 가정할 때, 부모 프로세스는 위 프로그램을 수행하는 동안 총 세 번의 시스템 콜을 호출하게 된다.
- ⑤ 자식 프로세스를 fork한 직후, 부모 프로세스와 자식 프로세스는 위 프로그램의 같은 위치에서부터 동작하게 된다.

해설)

오답 체크 :

(4) 시스템 콜 : 부모 프로세스는 2번의 시스템 호출(wait, exit)을 수행한다.

정답 체크 :

(5)  $pid = fork()$  : 자식 프로세스를 생성한다. 부모와 자식 프로세스의 분기가 일어난다. 즉, 2개의 프로세스가 존재한다. // 부모 프로세스와 자식 프로세스가 프로그램의 같은 위치에서부터 동작하게 된다.

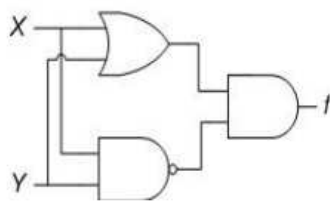
(1), (2)  $pid < 0$ 이면 자식 프로세스 생성에 실패한 것이고,  $pid == 0$ 은 자식 프로세스를 의미하고,  $pid > 0$ 은 부모 프로세스를 의미한다. // 조건문에서 첫 번째 조건, 두 번째 조건, 세 번째 조건에 해당한다.

$execvp()$  : /bin/lis의 lis를 수행한다.

(3)  $wait()$  : 자식 프로세스가 끝날 때까지 기다린다. // 부모 프로세스는 자식 프로세스가 종료한 후에 종료한다.

Tip! : 원래 fork도 부모 프로세스에 포함되는 시스템 호출인데, 문제의 의도는 하나의 프로세스에서 fork가 발생하여 부모와 자식으로 분리된 것으로 보고 있다. 그러므로 fork 이후에 부모가 수행한 시스템 호출은 2개이다.

11. 다음의 논리 회로와 동일한 연산을 수행하는 것은?



- ①  $f = X \text{ NAND } Y$
- ②  $f = X \text{ NOR } Y$
- ③  $f = X \text{ XOR } Y$
- ④  $f = X \text{ OR } Y'$
- ⑤  $f = X' \text{ AND } Y$

해설)

정답 체크 :

(3) 논리 회로를 부울 대수로 표현하면 다음과 같다.

$$(x+y)(xy)' = (x+y)(x'+y') = x'y+yx' = x \text{ xor } y$$

12. 다음 중 RISC(Reduced Instruction Set Computer)의 특징에 해당하지 않는 것은?

- ① 명령어별로 소요되는 Clock Cycle이 기본적으로 같다.
- ② 한 개의 명령어로 여러 작업을 수행할 수 있다.
- ③ 명령어 길이가 명령어 종류에 관계없이 일정하다.
- ④ 주소지정 방식이 단순하다.
- ⑤ CISC(Complex Instruction Set Computer)에 비해 Pipelining 구현이 용이하다.

해설)

정답 체크 :

(2) 여러 작업 : 해당 설명은 CISC이고, RISC는 한 개의 명령어로 한 개의 작업을 수행할 수 있다.

오답 체크 :

(1) Clock Cycle : 명령어가 단순하므로 사이클당 한 명령어 실행(one instruction per cycle)이 가능하다. 즉, 기계 사이클당 하나의 기계어가 실행된다.

(3) 명령어 길이 : 고정 길이의 명령어를 사용하여 더욱 빠르게 해석할 수 있다. 명령어의 길이를 고정하면 파이프라인 처리의 고속화를 꾀할 수 있지만, 컴파일러의 최적화 과정이 복잡해지기 쉽다.

(4) 주소 지정 방식 : 단순하게 레지스터 주소 지정 방식을 사용하므로 적은 수의 간단한 주소지정 방식을 사용 할 수 있다.

(5) Pipelining : 명령어 수행 단계별 시간이 일정하고, 연산 코드의 해독과 레지스터 오퍼랜드의 액세스가 동시에 일어나는게 가능하다.

13. 다음은 검색 및 정렬 알고리즘에 관한 특징을 나열한 것이다. 잘못 기술한 것은?

- ① 순차검색(Sequential Search) : 모든 레코드를 처음부터 탐색한다.
- ② 해쉬(Hash) : 레코드 양과 관계없이 검색 시간이 일정하다.
- ③ 트리검색(Tree Search) : 이진트리를 구성하여 실행한다.
- ④ 이분검색(Binary Search) : 자료가 정렬(Sort)되어 있어야 가능하다.
- ⑤ 삽입정렬(Insertion Sort) : 최악의 경우  $O(n^2)$  시간을 필요로 한다.

해설)

정답 체크 :

(2) 해시 : 레코드 양이 적으면 검색 시간이 일정하지만, 레코드 양이 많으면 충돌이 발생하고 충돌을 해결하기 위해 검색 시간이 증가하게 된다.

오답 체크 :

(1) 순차 : 리스트에서 찾고자 하는 값을 맨 앞에서부터 끝까지 차례대로 찾아 나가는 것이다. 검색할 리스트의 길이가 길면 비효율적이지만, 검색 방법 중 가장 단순하여 구현이 쉽고 정렬되지 않은 리스트에서도 사용할 수 있다는 장점이 있다.

(3) 트리 : 이진탐색트리는 탐색작업을 효율적으로 하기 위한 자료구조이다.

(4) 이분 : 이분 검색은 검색할 자료를 반씩 나누어 나머지 반만 검색하는 방식을 반복하여 자료를 찾는 검색 방법이다. 이 이분 검색을 이용하여 자료를 찾는다면 빠른 속도로 원하는 자료를 찾을 수 있다. 단, 이분 검색은 정렬되어있는 데이터에 사용할 수 있다.

(5) 삽입 : 최선은  $O(n)$ 이고, 평균과 최악은  $O(n^2)$ 이다.

14. 부동소수점 계산(floating-point calculation)에서 정규화(normalization)를 하는 이유는?

- ① 가수의 값을 크게하기 위하여
- ② 가수부의 비트를 줄이기 위하여
- ③ 연산 속도를 빠르게하기 위하여
- ④ 유효 숫자를 늘리기 위하여
- ⑤ 지수부를 최대화하기 위하여

해설)

정답 체크 :

(4) 정규화된 표현은 부동소수점 수에 대한 표현을 통일하기 위한 방법이다. 즉, 정규화된 표현을 사용하여 되면 표현할 수 있는 유효 숫자를 늘리게 된다.

15. 다음 중 TCP(Transmission Control Protocol)에 대한 설명으로 옳지 않은 것은?

- ① OSI 7-계층 모델에서 트랜스포트(transport) 계층에 해당한다.
- ② 다수의 기기에 대한 브로드 캐스팅(broad casting)을 지원한다.
- ③ Connection-oriented 프로토콜이다.
- ④ 패킷이 전송 도중 손상되거나 손실될 경우 재전송을 수행한다.
- ⑤ 흐름 제어(flow control) 기능이 지원된다.

해설)

정답 체크 :

(2) 브로드캐스팅 : 해당 설명은 UDP에 해당하고, TCP는 브로드캐스팅을 지원하지 않는다.

오답 체크 :

- (1) 트랜스포트 : 4계층에 해당한다.
- (3) Connection-oriented : 연결을 맺고 패킷을 전송한다.
- (4) 재전송 : 패킷에 문제가 발생했을 때 재전송(retransmission) 기능을 제공한다.
- (5) 흐름 제어 : 흐름 제어와 혼잡 제어 기능을 지원한다.

16. C와 같은 고급 언어로 작성된 프로그램은 컴파일 과정을 거쳐 CPU에 의해 실행 가능한 바이너리 형태의 실행 파일(executable file)로 변환된다. 다음 중 생성되는 실행 파일의 크기를 결정하는 요소가 아닌 것은?

- ① CPU의 동작 클럭 주파 수
- ② CPU의 설계 방식(RISC 혹은 CIS C)
- ③ 사용된 컴파일러의 최적화 옵션
- ④ CPU의 비트수(8 비트/16 비트/32 비트/64 비트)
- ⑤ 동적 링킹(dynamic linking) 사용 여부

해설)

정답 체크 :

(1) CPU의 동작 클럭 주파수 : 크기가 아니라 실행 속도와 연관된다.

오답 체크 :

- (2) CPU의 설계 방식 : CISC는 명령어의 길이가 짧고, 이로 인해 크기가 작아진다.
- (3) 최적화 옵션 : 최적화가 될수록 크기가 작아진다.
- (4) 비트수 : 비트수가 클수록 크기가 커진다.
- (5) 동적 링킹 : DLL을 사용하면 크기가 작아진다.



17. CPU와 DRAM 사이에 캐시(cache)가 있는 구조에서, CPU가 캐시와 DRAM을 접근하는데 각각 1 사이클과 100 사이클이 소요된다고 가정하자. 캐시 적중률(hit ratio)이 90%라고 할 때 평균 메모리 접근 시간은?

- ① 1.1 사이클
- ② 1.9 사이클
- ③ 10.1 사이클
- ④ 10.9 사이클
- ⑤ 11 사이클

해설)

정답 체크 :

(5) 주기억장치와 캐시기억장치에서 데이터를 인출하는데 소요되는 평균기억장치 접근시간(Taverage)은 다음과 같다.  $T_{average} = H_{hit\_ratio} \times T_{cache} + (1 - H_{hit\_ratio}) \times T_{main}$  여기서,  $H_{hit\_ratio}$ 는 적중률이고,  $T_{cache}$ 는 캐시기억장치 접근시간,  $T_{main}$ 은 주기억장치 접근시간이다. 주의할 것은  $T_{main}$ 은 주기억장치로 바로 접근하는 시간이 아니고 캐시로 가서 실패한 후에 주기억장치로 가는 시간을 의미한다. 이를 기반으로 문제를 풀면 다음과 같다.

$0.9 \times 1$  : 90%는 캐시(1 사이클)에서 가져간다.

$0.1 \times 101$  : 10%는 DRAM에서 가져간다. DRAM(100 사이클)에 바로 가는 것이 아니라 캐시(1 사이클)에서 miss가 발생했을 때 DRAM으로 간다. 그러므로 총 101 사이클이 소요된다.

$0.9 \times 1 + 0.1 \times 101 = 11$  사이클 : 이를 계산하면 총 11 사이클이 나온다.

오답 체크 :

(4)  $0.9 \times 1 + 0.1 \times 100 = 10.9$  : 해당 사이클로 계산하지 않도록 주의한다. 해당 사이클로 계산될 수 있는 경우 DRAM에 접근하는 사이클에 캐시에서 miss가 난 사이클이 포함될 때이다.

18. 최근 NAND 플래시 메모리를 이용한 저장장치가 모바일 기기를 중심으로 확산되고 있다. 다음 중 NAND 플래시 메모리의 특징이 아닌 것은?

- ① NAND 플래시 메모리는 페이지(page) 단위로 읽기/쓰기가 행해지며, 페이지의 크기는 보통 섹터 크기의 배수로 정해져 있다.
- ② 데이터를 많이 쓸수록 셀의 수명이 단축된다.
- ③ 한번 쓴 페이지에 새로운 데이터를 쓰기 위해서는 이전의 데이터를 먼저 지운 후에만 가능하다.
- ④ DRAM과 같이 데이터의 내용을 보존하기 위해 주기적인 리프레시(refresh)가 필요하다.
- ⑤ 읽기/쓰기 연산을 하지 않을 때에는 거의 전력을 소모하지 않는다.

해설)

정답 체크 :

(4) 리프레시 : 캐패시터를 사용하지 않기 때문에 리프레시가 필요 없다. 그리고 메모리 칩 안에 정보를 유지시키는 데에 전력이 필요 없는 비휘발성 메모리이다.

오답 체크 :

(1) 읽기/쓰기 : 페이지 단위로 읽기/쓰기 동작이 가능하다.

(2) 수명 단축 : 매체의 소재 자체의 한계로 인해 기록 가능 횟수에 한계가 있다.

(3) 지운 후 쓰기 : 덮어 쓸 수 없으므로, 모든 블록을 지우기 전까지는 해당 자료를 변경할 수 없다.

(5) 전력 소모 : 연산을 수행하지 않을 때의 대기 중 전력 소모는 낮다.

19. 익스트림 프로그래밍(XP: eXtreme Programming) 방법에서 채택한 것으로 일련의 차례와 계획을 기반으로 하여 개발을 진행시키지 않고, 일정한 주기를 가지고 끊임없이 프로토타입을 만들어 내며 그때 그때 필요한 요구를 더하고 수정하여 하나의 커다란 소프트웨어를 만들어내는 소프트웨어 개발방법론은?

- ① Waterfall development
- ② Spiral development
- ③ Agile development
- ④ Rapid application development
- ⑤ Plan-driven development

해설)

정답 체크 :

(3) Agile : 폭포수 프로세스의 단점을 해결하여 절차와 도구보다, 개인과 소통을 중요시 한다. 잘 쓴 문서보다는 실행되는 소프트웨어에 더 가치를 두고, 계약 절충보다는 고객 협력을 더 중요하게 여긴다. 계획을 따라 하는 것보다, 변경에 잘 대응하는 것을 중요하게 여긴다

오답 체크 :

(1) Waterfall : 1970년대 소개되었고, 각 단계가 다음 단계 시작 전에 끝나야 한다. 단순하거나 응용 분야를 잘 알고 있는 경우 적합하고, 결과물 정의가 중요하다.

(2) Spiral : 소프트웨어의 기능을 나누어 점증적으로 개발한다. 점증적으로 개발하게 되면 실패의 위험을 줄이고 테스트 등이 용이하게 된다.

(4) Rapid application : 프로토타이핑(원형) 모델 기준 사용자 요구사항, 분석, 설계, 개발을 신속한 시스템으로 개발한다(원형 모델을 개선). 제한된 범위의 단독 시스템을 CASE (Computer Aided Software Engineering, 시스템 개발 방법론들의 자동화를 지원하는 소프트웨어 도구를 제공해 개발자의 반복적인 작업량을 줄이도록 하는 것)와 같은 다양한 도구를 활용하여 신속히 개발한다.

(5) Plan-driven : 소프트웨어를 개발하는 과정에서 계획을 세우고 그 계획을 실천하는데에 많은 시간과 노력을 할애하는 개발 방법이다. 장점으로는 소프트웨어 개발이 조금더 예측가능해지고, 효율적이게 된다는데에 있다. 하지만, 단점도 존재하는데 가장 큰 단점이자 가장 많은 지적을 받는 점으로는 너무 계획에 치중을 하다보니 개발 방법 자체가 너무 형식에만 신경을 쓰고 얽매이게 된다는 것이다.

20. 페이징(paging)을 기반으로 한 가상 메모리 시스템과 관련한 다음의 설명 중 옳지 않은 것은?

- ① 프로세스에서 사용하는 가상 주소(virtual address)가 페이지 테이블을 통해 물리 주소(physical address)로 변환된다.
- ② 동일한 물리 주소를 서로 다른 프로세스에서 서로 다른 가상 주소를 이용해 접근 하는 것이 가능하다.
- ③ 가상 주소의 변경 없이 해당 가상 주소가 가리키는 데이터의 물리적인 주소를 변경시킬 수 있다.
- ④ Intel이나 ARM CPU에서는 하나 이상의 페이지 크기를 지원한다.
- ⑤ 페이지 크기가 커질수록 외부 단편화(external fragmentation) 문제가 심각해진다.

해설)

정답 체크 :

(5) 외부 단편화 : 페이지 크기가 커질수록 내부 단편화 문제가 심각해진다.

오답 체크 :

(1) 페이지 테이블 : 논리 주소를 물리 주소로 변환한다.

(2) 동일한 물리 주소 : 여러 프로세스가 하나의 페이지를 공유한다.

(3) 물리주소 변경 : 페이지 교체, 프로그램 재배치(relocation) 등으로 물리주소가 변경될 수 있다.

(4) Intel이나 ARM : Intel은 4KB, 2MB/4MB, 1GB의 페이지 크기를 제공하며, ARM은 4KB, 16KB, 64KB의 페이지 크기를 제공한다.