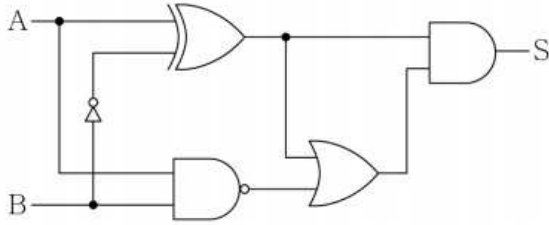


2016-서울시-컴퓨터일반-A형-해설-곽후근

1. 다음 논리회로에서 A=1010, B=0010일 때, S에 출력되는 값은?



- ① 1011
- ② 1101
- ③ 0111
- ④ 1110

정답 체크 :

(3)

첫 번째 비트에서 A가 1이고, B가 0일 때 S는 0이 나온다. 왜냐하면, S의 AND 입력 중에 하나(A XOR B' = 1 XOR 1 = 0)가 0이기 때문이다. 보기 중에 첫 번째 출력이 0인 것은 하나밖에 없기 때문에 두 번째 비트를 검사할 필요가 없다.

Tip! : 비슷한 문제가 나오면 시간을 절약하기 위해 최소한의 검사만을 수행한다.

2. 현재 실행 중인 프로세스에 할당된 CPU사용권을 다른 프로세스에게 할당하려면, 현재 실행 중인 프로세스의 실행 정보를 저장하고 다음으로 실행할 프로세스의 실행정보를 가져오는 과정이 필요하다. 이 과정을 무엇이라고 하는가?

- ① 컨텍스트 스위칭(Context Switching)
- ② 가상메모리(Virtual Memory)
- ③ 교체정책(Replacement Strategy)
- ④ 디스패치(Dispatch)

정답 체크 :

(1) 컨텍스트 스위칭 : 실행 중인 프로세스의 제어를 다른 프로세스에 넘겨 실행 상태가 되도록 하는 것이다. 프로세스 문맥 교환이 일어나면 프로세서의 레지스터에 있던 내용 저장한다.

오답 체크 :

(2) 가상메모리 : 사용자와 논리적 주소(보조기억장치)를 물리적(주기억장치)으로 분리하여 사용자가 주기억장치 용량을 초과한 프로세스에 주소를 지정해서 메모리를 제한 없이 사용할 수 있도록 하는 것이다. 프로그램 전체를 동시에 실행하지 않으므로, 요구한 메모리 전체가 아닌 일부만 적재해도 실행 가능한 원리를 이용한다.

(3) 교체정책 : 페이지 부재 발생 시 메인 메모리에 있으면서 사용하지 않는 페이지 없애 새로운 페이지로 바꾸는 것이다. 교체 알고리즘에는 선입선출(FIFO), 최적페이지(OPT), 최근최소사용(LRU, 카운터, 스택), 최근최소사용근접(참조비트, 시계-2차적기회, NUR, LFU, MFU) 등이 있다.

(4) 디스패치 : 준비(Ready) 상태의 프로세스가 CPU를 할당받아 실행(Running) 상태로 전이되는 과정을 의미한다.

3. 다음 중 유효한 SQL 문장이 아닌 것은?

- ① SELECT * FROM Lawyers WHERE firmName LIKE '% and %';
- ② SELECT firmLoc, COUNT(*) FROM Firms WHERE employees < 100;
- ③ SELECT COUNT(*) FROM Firms WHERE employees < 100;
- ④ SELECT firmLoc, SUM(employees) FROM Firms GROUP BY firmLoc WHERE SUM(employees) < 100;

정답 체크 :

(4) GROUP BY : 그룹에 대한 조건을 작성할 때는 WHERE가 아니라 HAVING을 사용해야 한다.

오답 체크 :

- (1) LIKE : LIKE 키워드는 %(0개 이상의 문자)와 _(한 개의 문자)와 같이 사용할 수 있다.
- (2) COUNT(*) : COUNT(속성)은 속성 값의 개수인데, 속성을 *로 지정하면 모든 속성 즉, NULL 값의 유무에 상관없이 대상 테이블에서 모든 행(튜플)의 수를 계산한다.
- (3) COUNT(*) : COUNT(속성)은 속성 값의 개수인데, 속성을 *로 지정하면 모든 속성 즉, NULL 값의 유무에 상관없이 대상 테이블에서 모든 행(튜플)의 수를 계산한다.

4. 다음 중 나머지 셋과 역할 기능이 다른 하나는?

- ① Array processor
- ② DMA
- ③ GPU
- ④ SIMD

정답 체크 :

(2) DMA : 대용량의 데이터를 이동시킬 때 효과적인 기술로, 기억장치와 입출력 모듈 간의 데이터 전송을 별도의 하드웨어인 DMA 제어가 처리하고, 중앙처리장치는 개입하지 않도록 하는 입출력 처리 기법이다.

오답 체크 :

- (1) Array processor : 벡터 프로세서(vector processor)와 배열 프로세서(array processor)가 대표적인 SIMD 분류에 속한다.
- (3) GPU : GPU도 하나의 명령어를 다수의 처리장치들에서 처리되기 때문에 SIMD 분류에 속한다.
- (4) SIMD : 하나의 명령어 스트림(IS)이 다수의 처리장치들에서 동시 처리되는 기술이다. 하나의 명령어는 각 처리요소가 각 기억장치에 저장된 독립된 데이터를 처리하도록 한다. 결과적으로 하나의 제어장치는 하나의 명령어를 인출하여 해독하고, 여러 개의 처리장치는 여러 데이터를 동시에 인출하여 명령어를 실행한다.

5. 다음은 IPv6에 대한 설명이다. 옳지 않은 것은?

- ① 기존의 IP 주소 공간이 빠른 속도로 고갈되어 왔기 때문에 고안되었다.
- ② IPv6는 IP 주소 크기를 기존의 4바이트에서 6바이트로 확장했다.
- ③ IPv6는 유니캐스트, 멀티캐스트 주소뿐만 아니라 새로운 주소 형태인 애니캐스트 주소가 도입되었다.
- ④ 네트워크 프로토콜을 바꾼다는 것은 매우 어렵기 때문에 IPv6로의 전환을 위해 여러 방법들이 고안되었다.

정답 체크 :

(2) 6바이트 : 16바이트로 확장되었다.

오답 체크 :

(1) 고갈 : IoT로 인해 더 많은 IP 주소가 필요하게 되었다.

(3) 애니캐스트 : 애니캐스트 주소는 여러 인터페이스를 식별한다. 애니캐스트 주소로 지정된 패킷은 적절한 멀티캐스트 라우팅 토폴로지를 통해 주소로 식별되는 가장 가까운 인터페이스인 단일 인터페이스로 배달된다.

(4) 전환 : IPv4/IPv6 듀얼스택 방식, 터널링(Tunneling) 방식, 주소 변환 방식 등이 존재한다.

6. 다음 정렬 알고리즘 중 최악의 경우에 시간복잡도가 가장 낮은 것은?

① 버블 정렬(Bubble sort)

② 삽입 정렬(Insertion sort)

③ 퀵 정렬(Quick sort)

④ 힙 정렬(Heap sort)

정답 체크 :

(4) 힙 : 최악의 경우에 시간 복잡도는 $O(n\log n)$ 이다.

오답 체크 :

(1) 버블 : 최악의 경우에 시간 복잡도는 $O(n^2)$ 이다.

(2) 삽입 : 최악의 경우에 시간 복잡도는 $O(n^2)$ 이다.

(3) 퀵 : 최악의 경우에 시간 복잡도는 $O(n^2)$ 이다.

7. 다음 C 프로그램의 실행 결과는?

```
#include <stdio.h>
int main()
{
    char* array1 [2]={"Good morning", "C language"};
    printf("%s \n", array1 [0]+5);
    printf("%c \n", *(array1 [1]+6));
    return 0;
}
```

①

Good morning

C-language

②

morning

a

③

morning

g

④

morning

u

정답 체크 :

(4)

array1은 두 개의 문자열을 가리키는 포인터 배열이다.

첫 번째 포인터인 array1[0]은 첫 번째 문자열("Good morning")의 시작 주소를 가리킨다.

두 번째 포인터인 array1[1]은 두 번째 문자열("C language")의 시작 주소를 가리킨다.

array1[0]+5는 첫 번째 문자열에서 6번째 문자인 m의 주소를 나타내고, 이를 %s인 문자열로 출력하면 "morning"이 된다.

array1[1]+6은 두 번째 문자열에서 7번째 문자인 u의 주소를 나타내고 해당 값(*)을 문자인 %c로 출력하면 'u'가 된다.

8. 메모리 크기가 200KB인 시스템에서 요구 페이징(demand paging)으로 가상 메모리(virtual memory)를 구현한다고 하자. 페이지 크기가 2KB이고 페이지 테이블(page table)의 각 항목이 3바이트라고 하면, 25KB 크기의 프로세스를 위한 최소 페이지 테이블의 크기는 어떻게 되는가?

- ① 25바이트
- ② 39바이트
- ③ 60바이트
- ④ 75바이트

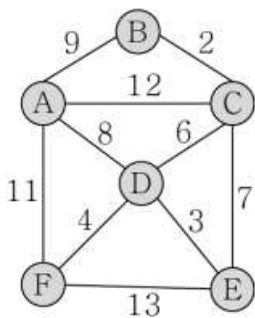
정답 체크 :

(2)

25KB 크기의 프로세서는 13개의 페이지가 필요하다(= 프로세스 크기 / 페이지 크기 = 25KB / 2KB = 13).

각 페이지에 테이블 항목으로 3바이트가 필요하므로 최소 페이지 테이블의 크기는 39바이트이다(= 페이지 개수 x 테이블 항목 크기).

9. 다음 그래프에서 최소 비용의 신장 트리 값은 얼마인가?



- ① 16
- ② 20
- ③ 23
- ④ 26

정답 체크 :

(3)

문제의 조건에 MST 알고리즘이 주어지지 않았으므로 가중치가 가장 작은 간선을 묶어 신장 트리를

만들면 된다. 가중치가 가장 작은 간선을 묶으면 다음과 같다.

$$8 + 6 + 4 + 3 + 2 = 23$$

10. 라우팅 알고리즘은 라우터에 패킷이 도착했을 때 포워딩 테이블을 검색하고 패킷이 전달될 인터페이스를 결정하는 알고리즘이다. 다음 중 라우팅 알고리즘이 아닌 것은?

- ① RIP(Routing Information Protocol)
- ② OSPF(Open Shortest Path First)
- ③ CDMA(Code Division Multiple Access)
- ④ BGP(Border Gateway Protocol)

정답 체크 :

(3) CDMA : 2세대 이동통신 기술로서 현재까지 사용되고 있다. 아날로그 형태인 음성을 디지털 신호로 전환한 후 여기에 난수를 추가하여 여러 개의 디지털 코드로 변환해 통신을 하는 것으로 휴대 폰이 통화자의 채널에 고유하게 부여된 코드만을 인식한다.

오답 체크 :

(1) RIP : AS(autonomous system) 내부(Intra-AS)에서 경로배정을 위해 사용된다. 모든 경로를 업데이트하기 때문에 traffic load가 심하다.

(2) OSPF : AS(autonomous system) 내부(Intra-AS)에서 경로배정을 위해 사용된다. 변경된 경로만 업데이트하기 때문에 traffic load를 감소시킨다.

(4) BGP : AS 경계(Inter-AS)에서 경로배정을 위해 사용된다.

Tip! : IDRP는 AS 사이(Inter-AS)에서 경로배정을 위해 사용된다.

11. 암달(Amdahl)의 법칙은 컴퓨터 시스템의 일부를 개선할 때 전체적으로 얼마만큼의 최대 성능 향상을 기대할 수 있는지를 예측하는 데 사용된다. 만약 특정 응용프로그램의 75%가 멀티코어(Multicore)를 이용한 병렬 수행이 가능 하고 나머지 25%는 코어의 수가 증가해도 순차 실행만 가능하다는 전제 하에, 컴퓨팅 코어(Core)의 수를 4개로 늘릴 때 기대할 수 있는 최대 성능 향상은 약 몇 배인가?

- ① 약 1.28배
- ② 약 2.28배
- ③ 약 3.28배
- ④ 약 4.28배

정답 체크 :

(2)

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 0.75이고 S는 4가 된다. 이를 수식에 대입하면 다음과 같다.

$$\frac{1}{(1 - 0.75) + \frac{0.75}{4}} = 2.28$$

12. <보기> 중 우리가 흔히 인터넷을 통해 비용을 지불하거나 혹은 무료로 사용하는, 클라우드 저장 서버에 대한 분류로 옳은 것을 모두 고르면?

- <보기>
- ㄱ. Public cloud
 - ㄴ. Private cloud
 - ㄷ. Software as a service(SaaS)
 - ㄹ. Platform as a service(PaaS)
 - ㅁ. Infrastructure as a service(IaaS)

- ① ㄱ, ㄷ
- ② ㄱ, ㅁ
- ③ ㄴ, ㄷ
- ④ ㄴ, ㄹ

정답 체크 :

(2)

(ㄱ) Public cloud : 일반적인 공적 업무에 이용하기 위하여 외부 서비스 제공자가 관리하는 클라우드이다.

(ㅁ) IaaS : 인프라스트럭처를 제공하는 서비스로, 대상자는 네트워크 아키텍처이다. 대표적인 예는 AWS Amazon Web Service 등이 있다.

오답 체크 :

(ㄴ) Private cloud : 회사 내부의 이용자가 공유할 수 있도록 만든 클라우드이다.

(ㄷ) SaaS : 웹 브라우저를 통하여 소프트웨어를 제공하는 서비스로, 대상자는 일반사용자이다. 대표적인 예는 Gmail 등이 있다.

(ㄹ) PaaS : 표준화된 플랫폼을 제공하는 서비스로, 대상자는 애플리케이션 개발자이다. 대표적인 예는 Google AppEngine, Microsoft Azure 등이 있다.

Tip! : 이외에도 클라우드 컴퓨팅의 서비스 유형에는 BaaS, NaaS, SECaaS 등이 있다.

13. 16진수로 표현된 B9E₍₁₆₎를 2진수로 표현하면 다음 중 무엇인가?

- ① 1100 0101 1101₍₂₎
- ② 0101 0101 1001₍₂₎
- ③ 1011 1001 1110₍₂₎
- ④ 1110 0101 1101₍₂₎

정답 체크 :

(3) 16진수 B9E은 2진수 1011 1001 1110로 바꿀 수 있다(16진수는 4비트의 2진수로 대응되고, 10부터 A로 할당함).

14. 다음 프로그램의 구성 요소들 중 프로세스 내에서 생성한 스레드들 사이에 공유되지 않는 것을 모두 고르면?

- ㄱ. 레지스터(Register)
- ㄴ. 힙(Heap) 메모리

ㄷ. 전역 변수(Global variables)

ㄹ. 스택(Stack) 메모리

- ① ㄱ, ㄴ
- ② ㄱ, ㄹ
- ③ ㄴ, ㄷ
- ④ ㄷ, ㄹ

정답 체크 :


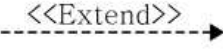
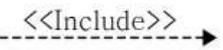

(2) 스레드는 스레드별로 스택, 지역 데이터, 스레드 실행 환경 정보(레지스터)를 가진다.

오답 체크 :

(1), (3), (4)

스레드는 그림에서 보는 바와 같이 코드(실행 파일), 전역 데이터(전역 변수), 힙(동적 할당)을 공유한다.

15. 다음 중 Use case diagram에서 사용하는 기본 요소가 아닌 것은?

- ① 
- ② 
- ③ 
- ④ 

정답 체크 :

(4) 해당 그림(직사각형)은 데이터베이스에서 객체를 표현할 때 사용하고, Use case diagram에서는 사용하지 않는다.

오답 체크 :

(1) Use case diagram에서 Actor(시스템과 상호작용하는 외부 엔티티)를 나타낸다.

(2) Use case diagram에서 확장 관계(정상적인 이벤트와 예외적인 이벤트를 분리)를 나타낸다. (포함 관계와 반대 방향임에 유의)

(3) Use case diagram에서 포함 관계(사용 사례 사이의 중복을 제거)를 나타낸다.

16. 가상메모리(Virtual Memory)를 효과적으로 제공하기 위해 Core i7과 같은 프로세서 내부에 있는 장치는 무엇인가?

- ① TLB(Translation Lookaside Buffer)
- ② 캐시(Cache)
- ③ 페이지 테이블(Page Table)
- ④ 스왑 스페이스(Swap Space)

정답 체크 :

(1) TLB : 가상 메모리 주소를 물리적인 주소로 변환하는 속도를 높이기 위해 사용되는 캐시이다. TLB는 최근에 일어난 가상 메모리 주소와 물리 주소의 변환 테이블을 저장하기 때문에 일종의 주소

변환 캐시라고 할 수 있다.

오답 체크 :

(2) 캐시 : CPU와 주기억장치의 속도차를 개선하기 위해 CPU와 주기억장치 사이에 놓인 고속의 메모리를 캐시라고 한다.

(3) 페이지 테이블 : 논리 주소를 물리 주소로 변환한다. 일반적으로 주기억장치에 상주한다.

(4) 스왑 스페이스 : 컴퓨터의 실제 메모리, 즉 주기억장치의 가상 메모리 확장으로 사용되는 하드디스크 상의 한 공간이다. 주기억장치에 있는 것 중 가장 오래 전에 사용되었던 파일들은, 새로운 파일이 주기억장치에 들어올 수 있는 공간을 내주기 위해, 다시 필요해질 때까지 하드디스크로 스왑된다.

17. 다음 중 C 프로그래밍 언어의 식별자로 사용할 수 없는 것은?

- ① 3id
- ② My_ID
- ③ __yes
- ④ K

정답 체크 :

(1)

C언어에서 식별자를 만드는 규칙은 다음과 같다.

- 알파벳 문자와 숫자, 밑줄 문자 _로 구성한다.
- 첫 번째 문자는 반드시 알파벳 또는 밑줄 문자 _로 시작해야 한다. (보기의 3id는 숫자로 시작하기 때문에 식별자로 사용할 수 없다)
- 대문자와 소문자를 구별한다.
- C 언어의 키워드와 똑같은 이름은 허용되지 않는다.

18. 3개의 page를 수용할 수 있는 메모리가 있으며, 현재 완전히 비어 있다. 어느 프로그램이 <보기>와 같이 page 번호를 요청했을 때, LRU(Least-Recently-Used)를 사용할 경우 몇 번의 page-fault가 발생하는가?

<보기> 요청하는 번호순서 : 2 3 2 1 5 2 4 5

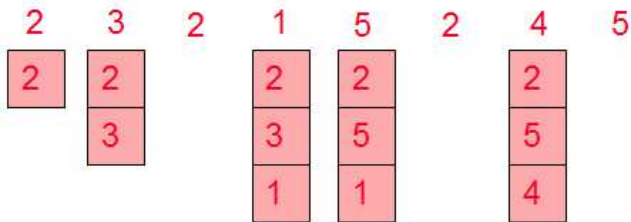
- ① 6번
- ② 5번
- ③ 4번
- ④ 3번

정답 체크 :

(2)

LRU는 프로세스가 가장 최근의 페이지에 액세스했다는 것은 멀지 않아 다시 액세스할 가능성이 있다는 의미이다. 과거 오랫동안 사용하지 않은 페이지로 대체하는 효과로 생각할 수 있다. 페이지를 대체할 때 오랫동안 사용하지 않은 페이지를 선택하므로 시간적으로 거꾸로 찾는 최적 페이지 대체 알고리즘이라고 할 수 있다. 주어진 조건으로 LRU를 적용하면 다음과 같다. 5번째 5가 들어왔을 때 페이지 프레임 내에 5가 없기 때문에 교체를 해야 한다. 이때, 1이 가장 최근에 사용되었고, 그 다음에 2가 사용되었기 때문에 3을 교체한다. 페이지가 사용되었다는 의미는 참조열이 프레임 내의 빈 공간에 새로 들어왔거나 참조열과 동일한 프레임이 존재하는 것을 의미한다. 이러한 방식으로 참조열

을 처리하면 총 5회의 페이지 부재(page-fault)가 발생한다.



19. 소프트웨어 프로젝트 관리가 어려운 이유로 옳지 않은 것은?

- ① 소프트웨어는 형태가 없어 프로젝트 관리자는 프로젝트 진척사항을 분석하는 데 어려움이 있다.
- ② 소프트웨어 개발 프로세스는 조직에 따라 가변적이므로 관리에 어려움이 있다.
- ③ 컴퓨터와 통신에서의 빠른 기술적 변화로 인해 관리자의 경험이 새로운 프로젝트에 전달되지 않을 수 있다.
- ④ 대규모 소프트웨어 프로젝트는 일회성(one-off) 프로젝트가 전혀 없어서, 경험이 충분한 관리자가 문제를 예측할 수 없다.

정답 체크 :

(4) 일회성 : 대규모 소프트웨어 프로젝트는 종종 일회성(one-off) 프로젝트이다.

오답 체크 :

- (1) 진척사항 : 소프트웨어 제품은 형태가 없다. 개발자가 Program을 개발하고 있지만 PL, PM, 발주자는 Project의 진척 상황을 알 수 없다. 왜냐하면 개발자가 개발한 것이 실질적으로 운용 될수 있는것을 개발 한것인지 아니면 껍데기만을 개발한것인지 개발을 안하였으면서 개발 한것 처럼 허위 진척 보고를 하고 있는것 인지 실질적으로 확인 할 수 있는 방법이 없다.
- (2) 가변적 : 표준화된 프로세스가 없다. 즉 조직에 따라 매우 가변적이다. 조직의 문화 또는 프로젝트의 규모, 프로젝트팀의 특성, 심지어는 프로젝트 팀원의 성향에 따라 변경 될수도 있다.
- (3) 기술적 변화 : 컴퓨터와 통신에서는 빠른 기술적 변화로 인해 관리자의 과거 경험이 새로운 프로젝트에 의미가 없을 수도 있다.

20. 다음 C 프로그램의 실행 결과는?

```
#include <stdio.h>
struct student {
    char name[20]; // 이름
    int money; // 돈
    struct student* link; // 자기 참조 구조체 포인터 변수
};
int main(void)
{
    struct student stu1 = {"Kim", 90, NULL};
    struct student stu2 = {"Lee", 80, NULL};
    struct student stu3 = {"Goo", 60, NULL};
    stu1.link = &stu2;
```

```
stu2.link = &stu3;
printf("%s %d\n", stu1.link->link->name, stu1.link->money);
return 0;
}
```

- ① Goo 80
- ② Lee 60
- ③ Goo 60
- ④ Lee 80

정답 체크 :

(1)

stu1.link = &stu2; // stu1.link는 stu2의 주소를 가리킨다.
 stu2.link = &stu3; // stu2.link는 stu3의 주소를 가리킨다.
 stu1.link->link->name // stu1.link는 stu2이므로 stu1.link->link(=stu2->link)는 stu3을 가리킨다. 그러므로 stu1.link->link->name(=stu3->name)은 "Goo"가 된다.
 stu1.link->money // stu1.link는 stu2이므로 stu1.link->money(=stu2->money)는 80이 된다.