9 ~ 10 semina

**Q. typedef문으로 함수 사용이 가능합니까? 또한 define문으로 typedef문의 기능을 사용할 수 있습니까?**

A. typedef문으로 함수를 사용하는 것은 불가능합니다. typedef문은 데이터형에만 기호명을 줄 수 있기 때문입니다. 그리고 define으로 typedef문의 기능이 사용 가능하며 둘의 차이점은 define은 프리프로세서이고 함수를 사용할 수 있지만, typedef는 컴파일러에 의해 처리되어지며 함수 사용이 불가능합니다.

**Q. 구조체를 사용하는 목적은 무엇입니까?**

A. 학생정보를 모을 때 배열로 학번, 이름, 전공 등을 따로따로 저장하지 않고 한꺼번에 저장을 하고 싶을때, 구조체를 사용하여 저장을 할 수 있습니다. 즉 관련이 있는 자료형을 자신이 원하는대로 한곳에 묶을 때 사용한다고 볼 수 있습니다.

**(추가) 구조체 정의 후 세미콜론을 붙히는 이유는 무엇입니까?**

A. 세미콜론을 붙였을 때 구조체 정의를 마치기 위해서 입니다.

**Q. 연결리스트를 사용하는 이유는 무엇입니까?**

A. 배열과 비교를 하여 도서관리나 학생정보를 관리할 때 배열은 데이터의 삽입 및 삭제 시 데이터들의 저장위치를 다 옮겨주어야 하지만 연결리스트는 노드가 가리키는 주소만 수정해주면 되기 때문에 배열에 비해 수정이 간편하고 메모리 상에서도 쓸데없는 공간을 낭비할 필요가 없습니다.

**Q. 도트연산자와 에로우연산자의 차이점은 무엇입니까?**

A. 도트연선자는 구조체 맴버에 접근하는데 사용이 되며 에로우 연산자는 포인터를 사용하여 구조체에 접근할 때 사용됩니다. 그렇기 때문에 st->data 를 (\*st).data로 표현이 가능합니다.

**Q. malloc 함수의 리턴값이 왜 void\*입니까?**

A. 일단 void\*란 어떠한 변수나 함수의 주소값을 담을 수 있는 변수입니다. 하지만 값의 변경이나 참조가 불가능하기 때문에 앞에 포인터의 형을 따로 캐스팅을 해주어야 합니다. 그래서 void\*는 일단 주소값에만 의미를 두고 포인터형은 나중에 결정할 때 사용합니다. 그리고 동적할당은 정적할당 변수처럼 변수 자체가 이름을 갖는 것이 아닌 포인터변수(참조변수)를 이용하여 간접접근을 합니다. 이떄 참조변수는 동적변수의 주소를 갖게 되기 때문에 void\*를 씁니다.

**Q. free함수 이외에 동적할당을 해제하는 함수가 있습니까?**

A. 4.3BSD 유닉스 운영체제에서 cfree함수를 가지며 free함수와 똑같은 동작을 합니다.

**Q. 연결리스트에는 어떤 것들이 있는지 아십니까?**

A. 연결리스트에는 ‘단일 연결리스트’, ‘이중 연결리스트’, ‘원형 연결리스트’ 가 있습니다. 이중 연결리스트는 prev(previous)가 맴버로 하나 더 추가되며 앞의 노드를 연결하게 됩니다. 원형 연결리스트는 이중 연결리스트에서 tali이 head를 가리키게 됩니다.

**(추가)follow없이 노드를 연결하는 방법을 말해주세요.**

A.

순서가 중요합니다.

current->next->prev = newnode;

newnode->next = current->next;

newnode->prev = current;

current->next = newnode;

**Q. malloc함수에서 sizeof연산자를 쓰지 않고 사용자가 크기를 임의로 넣었을 때 어떻게 됩니까?**

A. malloc 뒤에 오는 unsigned 형 정수들은 메모리의 최솟값을 의미하게 됩니다. 그래서 메모리를 넘은 값을 넣어도 문제는 일어나지 않지만 주변에 변수가 생성이 될 경우 메모리 침범 가능성이 있습니다.

**Q. 정적할당과 동적할당의 차이와 장단점에 대해 말해주십시오.**

A.

정적할당은 프로그램 실행 될 때 이미 메모리의 크기가 결정되어 있는 것 입니다.

동적할당은 프로그래밍 실행 시간 동안 사용할 메모리 공간을 할당하는 것으로 사용이 끝나면 운영체제가 쓸 수 있도록 반납하고 다음에 요구가 오면 재 할당을 받을 수 있습니다.

정적할당의 장점은 실행 도중에 해제되지 않고, 프로그램이 종료할 때 알아서 운영체제가 회수하기 때문에 동적할당과 달리 해제하지 않아서 메모리 누수가 일어나는 문제는 없습니다. 단점은 메모리 크기가 하드 코딩(시스템적으로 사용하는 변수 없이 값을 직접 소스코드에 넣어 사용하는 방식)으로 이루어져 메모리 크기가 고정되어 있습니다(변경이 불가능하다). 또한(배열) 데이터의 삽입과 삭제가 자유롭지 못합니다

동적할당의 장점은 자신이 원하는 크기만큼 메모리를 할당해 경제적이며, 이미 할당된 메모리라도 언제든지 크기를 조절할 수 있습니다(ralloc). 단점은 더 이상 사용하지 않을 때 명시적으로 해제해주어야만 합니다.

**Q. 구조체는 캐스트 연산이 가능합니까?**

A. 구조체 맴버는 가능하지만 구조체는 불가능합니다.

**Q. 주소의 0번지부터 free함수로 무한반복하여 메모리를 해제하면 어떻게 됩니까?**

A. 모든 메모리가 해제되기 때문에 문제가 됩니다.

**Q. 구조체와 malloc()를 안쓰고 연결리스트를 만들 수 있습니까?**

A. 배열과 포인터를 사용하여 만들수는 있겠지만 그럴 경우 삽입과 삭제 등 여러가지 귀찮은 일들이 발생할 수 있습니다.

**Q. 메모리 관리 기능을 선언하는 헤더는 무엇이 있습니까?**

A. stdlib.h와 alloc.h가 있습니다.