

임베디드 애플리케이션 분석

1.한달간 C 언어 등등 많은 것들을 학습했는데 이것들에 대해 자기 성찰을 수행 해보자!

내용

해당 과목에 대한 진심과 성의가 담긴 자기 성찰을 수행하도록 한다.

앞으로의 포부를 함께 기술하시오.

(배점 50 점)

처음 수업 때부터 안 어려운 날이 없었다.. 수업이 훌훌 지나갈 때마다 항상 개념이 잘 안 잡힌 상태에서

다음 수업을 들을 때 멘붕인 상태가 많았다

보통 수업 때는 못 따라가고 나중에 물어보면 그때 배운 게 이 내용이구나..(놓친내용에 대한 걱정도 있었음)

지금은 어려운 상태가 한 달간 지속되다 보니 익숙해진 것같기도 하지만

머리속에 남은 건 많지 않은것 같다.(복습이 많이 부족함->쉬는시간에 C언어를

공부해야겠다..코딩연습....;;)

사실 지금도 어떤 식으로 공부를 해야 할지 감이 잘 안 오는 경우가 많지만 일단은 이해를 바탕으로 하고 있다.

(이해를 제대로 했는지 의문이 들 때가 많음-막상 안다고 생각한 개념도 오개념일 때가 있었다.)

C언어 공부,자료구조,시스템 프로그래밍 그리고 현재 배우고 있는 네트워크 프로그래밍

복습도 해야 하는데

다 복습 하고 싶은 욕심 때문에 아무것도 못 한 적도 있지만.. 현재는 네트워크 프로그래밍 그 날 배운거 복습하는데에도 빠듯하다.

내용이 어려운 것도 있지만 머리속에 남은게 많지 않다고 느낀건 아마 지난 한 달간 내 공부를 하지 않아서 인 것 같기도 하고..

내 공부를 한다면보다 다른 친구들은 어떻게 하나를 더 많이 봤던 것 같다.(지금도 그렇지만 갈피를 잘 못 잡고 있어서 많이 보는 것 같다.

처음 보는 내용에 대한 접근 방법을 잘 모르겠음. 보통 수업 때 뭔가를 알려주시지만 단어들이 너무 생소해서 못 알아듣는게 많다.)

이제 첫 프로젝트를 할 팀도 있어서 실력을 많이 키워야겠다는 조바심도 난다

현재 너무나 막막한게 많고 당연하겠지만 더 열심히 해야겠다는 생각이 든다.(흠..어떤식으로 열심히 해야할지는 더 생각을 해봐야 할 것 같다.)

(진심을 담아서 암울한 얘기가 많다;;하지만 긍정적으로 해결해 보려함.^^;)

현재는 배우고 있는 내용을 다룰 수 있고 응용도 할 수 있을 정도로 만드는 것이 첫 목표이고 쿼드콥터에 대한 기술들을 알아보고 배운 다음 내가 다룰 수 있는 기술로 만드는 것이 두 번째 목표이다

2. 프로그램을 작성하시오.

내용

아무런 함수 3 개를 작성하여 프로그램이 동작하도록 만드시오.

(본인 만들고 싶은대로 만든다 - 함수 3 개 이상 쓰면 됨)

```
#include<stdio.h>
```

```
int func3(int num)
{
    if(num ==1 || num ==2)
        return 1;
    else
        return func3(num -1)+func3(num-2);
}
```

```
void func1(int c)
{
    int res1=0,res2=0,i;
    for(i = 0 ; i < c ;i++){
        if(func3(i+1)%2)
            res1 = res1 +func3(i+1);
        else
            res2 = res2 +func3(i+1);
    }
    func2(res1,res2);
}
```

```
void func2(int res1,int res2)
{
    printf("%d\n",res1-res2);
}
```

```
int main(void)
{
    int a,b,c;
    printf("30이하 입력한 항까지 피보나치 홀수와 짝수들의 합의 차 : ");
    scanf("%d",&c);

    func1(c);
    return 0;
}
```

3.C 언어 메모리 레이아웃

내용

C 언어 메모리 레이아웃을 기술하시오.

{커널영역 - 스택(로컬영역) - 힙(동적할당) - 데이터(전역변수,정적변수) }
텍스트(코드,기계어) } 가상메모리

Memory구조

프로그램을 실행시키면 운영체제는 우리가 실행시킨 프로그램을 위해 메모리 공간을 할당해준다.

할당되는 메모리 공간은 크게 스택,힙,데이터,텍스트영역으로 나뉜다

스택 영역 - 함수 호출 시 생성되는 지역변수와 매개변수가 저장되는 영역, 함수 호출이 완료되면 사라짐

힙 영역 - 필요에 의해 동적으로 메모리를 할당 할 때 사용

데이터 영역 - 전역 변수와 Static 변수가 할당되는 영역 프로그램의 시작과 동시에 할당되고

프로그램이 종료되어야 메모리에서 소멸됨

스택영역에 할당될 메모리의 크기는 컴파일 타임(컴파일하는 동안)에 결정

사용자의 요구에 맞게 메모리를 할당해 주기 위해서는(런타임에 메모리 크기를 결정하고 싶을 때)

메모리 동적 할당을 통해 힙 영역에 메모리를 할당해야 한다.

4.함수 포인터는 왜 쓰는가 ?

내용

알고 있는대로 기술하시오.

객체 지향 언어와 같은 효과를 누릴 수 있다

즉, 함수의 추상화를 통해 사용자에게는 필요한 인터페이스(사물과 사물 사이 또는 사물과 인간 사이의 경계에서, 상호 간의 소통을 위해 만들어진 물리적 매개체나 프로토콜을 말한다)만을 제공함으로써 보안 강화 및 유지/보수가 용이해진다

여기서,

절차지향언어와 객체지향언어 차이.

절차지향언어란 어떠한 절차에 따라 처리가 중요시되며 프로그램 전체가 유기적으로 연결되는 프로그래밍 방법

대표적으로 C언어 - 1.나는 목이 마르다 2. 슈퍼로 간다 3.콜라를 산다 4.콜라를 사서 마신다
이처럼 동작과 행동을 나열하여 순차적으로 실행하는 것을 의미
이렇게 순차적으로 실행되는 방식은 컴퓨터의 처리 방식과 비슷하여 객체지향보다 빠르게 처리되지만 유지/보수가 힘들다.

객체지향언어

객체지향언어란 동작이나 절차를 적어놓고 한 곳에 묶어놓은 것을 의미한다.

ex) 나/슈퍼/콜라/마신다 를 기술해놓고

내가 필요한 언어를, 필요할 때 꺼내어 쓰는 것을 의미.

때문에 절차지향언어에 비해 코드가 짧아서 동작을 수행하는데 더 효율적이라고 할 수 있다.

내가 아니라 철수,영희 등 몇 천명의 데이터를 처리할 때,
절차지향언어 같은 경우는 일일이 입력해야 하지만 객체지향언어는 목이 말라서 콜라를 마신다를 한 객체로 묶어놓을 수 있기 때문에
코드의 길이도 짧아지고 효율적으로 처리할 수 있다
대표적으로 C++,C#,JAVA

5.함수를 여러개 만드는 이유는 무엇인가 ?

내용

서술형이니 기술하시오.

코드를 유지/보수하는데 편하다

함수의 이름을 보고 어떤 기능인지 쉽게 알 수 있다. 이름이 없더라도 어떤 기능인지 눈에 더 잘 보일 수 있다.

반대로 통메인/통함수의 단점

코드 읽기가 매우 힘들고 재사용이 불가능.

유지/보수가 불가능하다.

디버깅이 힘들고 메모리를 많이 잡는다.

6.포인터가 없는 언어들과의 차이점

내용

포인터 때문에 할 수 있는 기능이 무엇인지 기술하시오.

함수를 호출하면 메모리에서 스택영역을 새로 할당한다

포인터가 없다면 어떠한 영역의 변수의 값을 바꾸는 일을 다른 함수에서 할 수 없다

그러한 이유로 포인터를 사용하면 프로그램을 하기 편한다

함수 호출시 인자를 보면 다른 사람이 코딩한 프로그램을 쉽게 파악할 수 있다

함수에서 한개 이상의 값을 리턴이 가능하다.

포인터를 인자로 준다면, 함수는 그변수의 값을 바꾸는 기능을 할 가능성이 매우 높다.

단순히 값을 인자로 준다면, 함수는 그 변수의 값을 이용만 할 것이라 추측이 가능하다.

크기가 큰 구조체를 주소값만 저장하여 포인터의 크기로 다룰 수 있다.

자료구조

1.이번 시험 결과와 관련하여 자료구조 과목에 대한 자기 성찰을 해보자!

내용

자기 성찰과 앞으로의 포부를 기술하시오.

(자기 성찰을 자기 성찰 답게 대충적지 말고 진심과 성심을 다해 적도록 한다)

큐,스택할 때는 많이 물어봐서 숨은 실 정도가 되었다

자료구조에서 어느정도 따라갈 수 있었던 정도는 Tree-insert까지 였다.(갑자기 숨이 안 쉬어짐)

삭제부분을 못 따라간건 트리의 인설트의 원리가 복잡한데 이걸 이해하느라 많이 힘들어서 인것 같다

그래서 그 이후도 어찌어찌 했건 안보고 삭제부분까지 코드를 짰었는데 지금은 건들지도 못하겠다;;
그때는 그렇게 계속 코드를 짜다보니 나중에는 외우는 식으로 자동적으로 쓰는 경우도 많았던 것 같았다.
그리고 지금 큐나 스택을 다시 보니
원리 이해가 제대로 되지 않았던지 오류가 나는 경우가 많았다.(자료구조가 끝나고 건튼적이 없다..)
다시 그림을 통해 차근차근 만들어 볼 필요가 있다고 생각한다.
다시 tree까지 해보고 avl까지 욕심을 내볼 생각이다.
된다면 재귀도 풀어보고 rb도 구경해 볼 생각이다

2. 연결리스트에 대한 문제

내용

선입선출을 따르는 구조를 무엇이라 하는가 ?

큐

3. 연결리스트 문제다.

내용

후입선출 기능을 가진 자료구조를 무엇이라 하는가 ?

스택

4.

트리에 관련한 개념 문제

내용

트리 자료구조는 연결리스트에 비해 어떠한 이점을 가질 수 있는가?

스택형태로 데이터가 정렬되어 있다면 10억번째에 데이터가 있을 경우 10억번을 검색해야 한다

하지만 이진트리를 사용하면 10억은 약 2의 31승 즉 31레벨까지만 검색하면 되므로

31번의 검색을 수행하면 된다.

즉, 경로설정을 빨리 해주기 위해 트리개념이 도입되었다.

5.AVL 트리 개념

내용

AVL 트리와 일반 트리의 차이점이 무엇인가 ?

기본 트리의 경우 완벽하게 트리구성이 안되는 경우가 많다. 그래서 avl트리(균형 이진 트리)가 만들어졌다.

1.이진 탐색 트리의 단점을 해결한 avl트리

이진 트리에서 모든 노드의 왼쪽과 오른쪽 트리의 높이차를 1이하로 만든 트리이다

6.레드 블랙 트리는 왜 사용하는가 ?

내용

기존 트리들과 레드 블랙 트리의 차이점은 무엇인가 ?

이진 탐색 트리를 베이스로 하기 때문에 이진 탐색 트리의 특성을 모두 갖는다.

rb트리는 최악의 경우에 대해 삽입,삭제 시간에 있어서 가장 나은 성능을 제공한다.

avl트리는 rb트리보다 균형이 잡혀 있으며 이로 인하여 최악의 경우에 삽입,삭제 동작 시 rb트리보다 더 많은 회전을 요구하게 된다.

일반적인 경우에는 보다 타이트한 균형을 가지는 avl트리가 평균적인 탐색 시간은 더 빠르다
삽입/삭제의 동작은 rb트리가 우의이고 최악의 경우에 avl트리보다 탐색이 빠르다고 한다면
avl트리보다 rb트리가 전체적으로 우위에 있다고 할 수 있다.