

TI DSP, MCU 및 Xilinx Zynq FPGA 프로그래밍 전문가 과정

강사 : Innova Lee(이상훈)

gcccompil3r@gmail.com

학생 : 황수정

sue100012@naver.com

15일차 (2018. 03. 14)

11. 표준 상태에서 아래의 소스 코드가 컴파일이 되는지 안되는지 판정하시오.
만약 컴파일이 안된다면 어떻게 바꿔야 하는지 기술하시오.

```
#define inc_each(x, y) { x++; y++; }
```

```
int main(void) {  
    int x = 10, y = 5;  
    if(x > y)  
        inc_each(x, y);  
    else  
        x = y;  
    return 0; }
```

> if문 밑에 inc_each(x,y); 뒤의 ';'을 뺀다. define에서 바꾸어 준 식이 {x++; y++;}인데 ';'을 빼지 않으면 {}이런 식이 되기 때문에 문법 오류가 되어서 컴파일이 되지 않는 것이다.

15. 현재 디버깅중이며 메모리 주소 0x7ffffb70에 어떤 값이 저장되었다. 이 때 위 메모리에 저장된 값을 보고 싶다면 어떻게 해야 하는가 ?

'p'라는 명령어를 사용한다. print를 뜻한다. p 0x7ffffb70 를 하면 그 값이 볼 수 있을 것이다.

p [변수명] 변수의 값을 출력한다.

p [함수명] 함수의 주소값을 출력한다.

포인터 변수의 경우 위의 방법으로 하면 주소값만 출력된다. 포인터 변수의 값 또는 포인터 구조체 등의 값을 보기 위해서는 *을 붙여준다. GDB는 변수 뿐만 아니라 레지스터의 값도 확인할 수 있다.

(gdb) print \$[레지스터명]

print 명령어는 지역변수를 우선하여 보여주기 때문에 지역변수와 전역변수에서 동일한 이름을 사용할때 전역변수를 확인하기 위해서는 :: 을 이용한다.

(gdb) print 'main.c'::[변수명]

파일명은 '따옴표' 으로 감싸야한다. 특정 함수에 있는 변수를 확인하기 위해서는

(gdb) print [함수명]::[변수명]

print 명령어로 변수 또는 레지스터를 확인할 때는 기본적으로 10진수로 출력한다. 이를 다른 형식으로 보고 싶을 때는 다음과 같은 방법을 이용한다.

(gdb) print/t [변수명] 2진수로

(gdb) print/o [변수명] 8진수로

(gdb) print/d [변수명] 10진수로 (int)

(gdb) print/u [변수명] 부호없는 10진수로 (unsigned int)

(gdb) print/x [변수명] 16진수로

(gdb) print/c [변수명] 최초 1바이트 값을 문자형으로

(gdb) print/f [변수명] 부동소수점값

(gdb) print/a [변수명] 가장 가까운 심볼의 오프셋

print 명령어는 값을 보여줄 뿐 아니라 값을 설정하는 것도 가능하다.

(gdb) print [변수명] = [값]

16. 다음 어셈블리어를 보고 함수의 main함수부터 my_function 함수까지 stack의 동작 방식을 그림과 함께 기술하라.

```
int my_function(int num1, int num2, int num3) { return num2 * num3 - num1; }

int main(void)    {
    int result = 0;

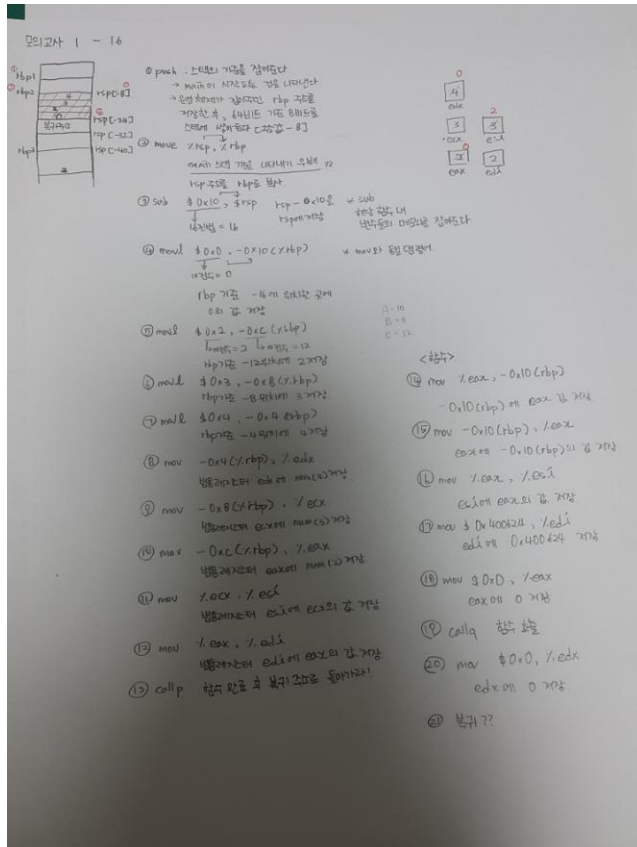
    int num1 = 2, num2 = 3, num3 = 4;

    result = my_function(num1, num2, num3);

    printf("result = %d\n", result);

    return 0; }
```

```
Dump of assembler code for function main:
0x000000000040053f <+0>:    push    %rbp
0x0000000000400540 <+1>:    mov     %rsp,%rbp
0x0000000000400543 <+4>:    sub     $0x10,%rsp
=> 0x0000000000400547 <+8>:    movl    $0x0,-0x10(%rbp)
0x000000000040054e <+15>:   movl    $0x2,-0xc(%rbp)
0x0000000000400555 <+22>:   movl    $0x3,-0x8(%rbp)
0x000000000040055c <+29>:   movl    $0x4,-0x4(%rbp)
0x0000000000400563 <+36>:   mov     -0x4(%rbp),%edx
0x0000000000400566 <+39>:   mov     -0x8(%rbp),%ecx
0x0000000000400569 <+42>:   mov     -0xc(%rbp),%eax
0x000000000040056c <+45>:   mov     %ecx,%esi
0x000000000040056e <+47>:   mov     %eax,%edi
0x0000000000400570 <+49>:   callq   0x400526 <my_function>
0x0000000000400575 <+54>:   mov     %eax,-0x10(%rbp)
0x0000000000400578 <+57>:   mov     -0x10(%rbp),%eax
0x000000000040057b <+60>:   mov     %eax,%esi
0x000000000040057d <+62>:   mov     $0x400624,%edi
0x0000000000400582 <+67>:   mov     $0x0,%eax
0x0000000000400587 <+72>:   callq   0x400400 <printf@plt>
0x000000000040058c <+77>:   mov     $0x0,%eax
0x0000000000400591 <+82>:   leaveq
0x0000000000400592 <+83>:   retq
---Type <return> to continue, or q <return> to quit---
```



26. 행렬의 곱셈을 프로그램으로 작성하라.

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

```
int main (void) {
int arr_1[2][2] = {{3,4}, {1,2}};
int arr_2[2][2] = {{3,2}, {4,2}};
int arr_3[2][2] = {0};
int i, j, k;
int sum = 0;

for(i=0;i<2;i++)
{
for(j=0;j<2;j++)
{
for(k=0; k<2; k++)
{
sum += arr_1[i][k]*arr_2[k][j];
}
arr_3[i][j] = sum;
sum = 0;
}
}
}
```

```

for(i=0; i<2; i++)
{
    for(j=0; j<2; j++)
        {   printf("%d\t", arr_3[i][j]);   }
    printf("\n");
}

return 0;    }

```

28. 정수형 변수 2개를 선언하고 서로 다른 값으로 초기화한다. 이후에 포인터를 사용하여 2개의 값을 서로 교환해보라.

```

#include< stdio.h>
int main (void)  {
    int a = 4;
    int b = 2;

    printf("a = %d\n b = %d\n", a, b);

    int *pa = &a;
    int *pb = &b;
    int tmp = *pa;
    *pa = *pb;
    *pb = tmp;

    printf("변환 후 a = %d\n b = %d\n", *pa, *pb);
    return 0;    }

```

30. 아래 코드를 보고 화면에 출력될 결과를 예측하시오.

```

#include< stdio.h>
int main(void)  {
int shortcut1 = 0, shortcut2 = 3, num = 9;

```

```

if(shortcut1&& num++)                >shortcut1이 0이므로 거짓이라 if문 pass
    shortcut1++;
    printf("%d, %d\n", shortcut1, num) >shortcut = 0 , num = 9로 출력

```

```

if(shortcut1 || shortcut2)           >둘 중 하나 값이 있어 참이 되어 if문 실행

```

```

    shortcut1++; >shortcut = 1
    printf("%d\n", shortcut1);                >shortcut1 = 1 출력

```

```

if(shortcut1&& ++num)                >shortcut1 이 1이 되었으므로 참으로 if문 실행
shortcut2--; >shortcut2 = 2
printf("%d, %d, %d\n", shortcut1, num, shortcut2);    >1, 10, 2 를 출력한다.

```

```

return 0; }

```

31. 아래 코드를 보고 화면에 출력될 결과를 예측하시오.

```

#include< stdio.h>
int main(void) {
    int i, j;
    int num1, num2, num3;
    int arr[2][6];
    int *arr_ptr[3] = {&num1, &num2, &num3};    >주소를 저장
    int (*p)[6] = arr;                            >포인터 배열에 배열 입력 이 포인터가
                                                    arr[]하나를 가르키게 된다.

    for(i = 0; i < 3; i++)
    { *arr_ptr[i] = i;                            >배열 값에 0, 1, 2 입력 num1,2,3도
      printf("%d\n", *arr_ptr[i]);                >0, 1, 2 이 출력 된다.
    }

    for(i = 0; i < 2; i++)
    for(j = 0; j < 6; j++)
    arr[i][j] = (i + 1) * (j + 1) * 10;            >arr[2][6]에 10,20,30,40,50,60
                                                    20,40,60,80,100,120

    for(i = 0; i < 2; i++)
    printf("%d\n", *p[i]);                        >p[0], p[1]을 출력하는데 10, 20 출력 된다

    return 0; }

```

32. 아래 코드를 보고 해당 프로그램이 어떻게 동작하는지 기술하시오.

```

#include< stdio.h>

```

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

```
int main(void)
{
    int result, final_val = 6;
    result = fib(final_val);
    printf("%d번째 항의 수는 = %d\n", final_val, result);
    return 0;
}
```

33. 임의의 값 x 가 있는데, 이를 4096 단위로 정렬하고 싶다면 어떻게 해야 할까? (힌트 : $4096 = 2^{12}$)
34. 구조체를 사용하여 Stack을 구현해보시오.

모의고사 2

1. 아래 Code를 작성하고 이 Code의 기계어에 대한 그림을 그리고 분석하시오.

```
void swap(int *a, int *b)
{
    int tmp;

    tmp = *a;
    *a = *b;
    *b = tmp;
}

int main(void)
{
```

```
int num1 = 3, num2 = 7;
swap(&num1,& num2);
return 0;
}
```

배점 : 20점

2. RB Tree에 3, 7, 10, 12, 14, 15, 16, 17, 19, 20을 넣을 때의 그림을 그리시오.

배점 : 3점

5. 임의의 값 x가 있는데, 이를 8192 단위로 정렬하고 싶다면 어떻게 해야 할까? (힌트 : $8192 = 2^{13}$)

배점 : 2점

7. int[4][4]배열이 메모리 공간에 나란히 존재함을 증명하라!

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

```
int main (void) {
int arr[4][4]={0}; //배열을 모두 0으로 초기화시키는 선언
int i, j;

for(i = 0; i<4; i++)
{
for(j = 0; j<4; j++)
printf("arr[%d][%d]의 주소는 %p\n", i, j, &arr[i][j]); }

return 0;
}
```

배점 : 3점 / 모의고사 1의 35번과 동일

9. char *str = "Pointer Is Very Important"라는 문자열이 있다. 여기에 대문자가 총 몇 개 사용되었는지 세어보자(프로그래밍으로)

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

```
int main (void) {
```



```

char *str = "Pointer Is Very Important";
int i;
int len = strlen(str); // 길이를 확인해서 조건식을 걸기 위함
int count = 0;
for(i = 0; i <= len; i++)
{ if(str[i] >= 65 && str[i] <=90)
{ count ++; // 후위 연산자로 조건에 맞을 경우 카운트가 된다.
printf("%c\\n", str[i]); //사용된 대문자의 갯수가 맞게 출력되었나 확인하기
위해 사용함.
}
}
printf("대문자는 총 %d개 사용되었다.\\n", count);
return 0; }

```

배점 : 3점

10. int arr1[3][3] = {{1, 3, 6}, {7, 1, 9}, {3, 7, 5}}과 int arr2[3][3] = {{3, 5, 7}, {9, 1, 8}, {5, 1, 2}}중 배열의 각 요소의 합이 가장 큰 것은 무엇인가 ?(프로그래밍)

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

```

int main (void){
int arr1[3][3] = {{1, 2, 3}, {7, 1, 9}, {3, 7, 5}};
int arr2[3][3] = {{3, 5, 7}, {9, 1, 8}, {5, 1, 2}};
int i, j, sum1, sum2;

```

```

for(i = 0; i < 3; i++)
{   for(j = 0; j < 3; j++)
        { sum1 += arr1[i][j];
          sum2 += arr2[i][j]; }
}

```

```

printf("arr1의 합은 %d이고 arr2의 합은 %d이다.\\n arr1 < arr2 일 때, 값은 %d이다
\\n 이 값은 참이면 1 거짓이면 0이다.\\n", sum1, sum2, sum1 < sum2);

```

```

return 0; }

```

>arr1의 합은 1942520886이고 arr2의 합은 32754이다. arr1 < arr2 일 때, 값은 0이다.
이 값은 참이면 1 거짓이면 0이다.
이므로 arr2 가 더 크다.

배점 : 2점

12. 아무 문자열을 calloc을 통해 동적할당하고 그 문자열을 배열로 옮겨보라

배점 : 2점

16. 이것이 없으면 C 언어를 사용할 수 없다. 이것은 무엇일까 ? (힌트 : 함수가 실행 하려면 이것이 필요하다)

>헤더파일?? 컴파일러???? 스택??? 메모리 주소???

배점 : 4점

18. 1, 3, 4, 7, 11, 18, 29, 47, 76, ... 형태로 숫자가 진행된다. 40번째 숫자는 무엇일까 ?

배점 : 4점

19. 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, ... 형태로 숫자가 진행된다. 1 ~ 30번째까지의 수들로 홀수들의 합을 하고 짝수들의 합을 구한다. 홀수들의 합 - 짝수들의 합의 결과를 출력하시오.

배점 : 2점

20. 3 by 3 행렬의 덧셈을 계산하는 프로그램을 작성하시오.

#include< stdio.h>

```
int main (void) {
int arr1[3][3] = {{1, 2, 3}, {2, 3, 4}, {3, 4, 5}};
int arr2[3][3] = {{2, 4, 5}, {1, 3, 2}, {2, 4, 5}};
int arr3[3][3] = {0};
int i, j;

for(i = 0; i < 3; i++)
{
    for(j = 0; j < 3; j++)
        { arr3[i][j] = arr1[i][j] + arr2[i][j]; }
}

for(i = 0; i < 3; i++)
{
    for(j = 0; j < 3; j++)
        { printf("%d\t", arr3[i][j]); }
}
```

```

        printf("%d\n");
    } return 0;
}

```

[복합문제]

배점 : 3점

21. 난수를 활용해서 Stack을 구성한다. (같은 숫자가 들어가지 않게 하고 20개 이상 넣는다. 이때 들어가는 숫자는 1 ~ 50사이의 숫자로 넣는다)

배점 : 5점

22. 그 중에서 홀 수만 빼내서 AVL Tree를 구성하도록 한다.

배점 : 5점

23. 나머지(짝 수)는 Binary Tree로 구성한다.

모의고사 3

5. int arr1[3][3] = {{1, 3, 6}, {7, 1, 9}, {3, 7, 5}}과

int arr2[3][3] = {{3, 5, 7}, {9, 1, 8}, {5, 1, 2}}중

배열의 각 요소의 합이 가장 큰 것은 무엇인가?(프로그래밍)

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

```

int sum1 (int arr1[3][3])                //배열을 함수인자로 받을 때는 포인터 사용x
{
    //배열의 이름이 주소이기 때문에 이중포인터 효과?
    int sum1 = 0, i, j;
    for(i = 0; i < 3; i++)
    {
        for(j = 0; j < 3; j++)
            { sum1 += arr1[i][j]; }
    }
    return sum1;                          //void형이면 return이 안됨!!!!
}

```

```

int sum2(int arr2[3][3])                //[[]]식으로 빈칸으로 두면 안됨!!!!!!
{
    int sum2 = 0, i, j;
    for(i = 0; i < 3; i++)
    {
        for(j = 0; j < 3; j++)
            { sum2 += arr2[i][j]; }
    }
    return sum2;
}

```

```
}
```

```
int main (void) {  
int arr1[3][3] = {{1, 3, 6}, {7, 1, 9}, {3, 7, 5}};  
int arr2[3][3] = {{3, 5, 7}, {9, 1, 8}, {5, 1, 2}};  
    sum1(arr1);  
    sum2(arr2);  
printf("arr1의 합 : %d,\n arr2의 합 : %d\n", sum1(arr1), sum2(arr2));  
    if(arr1 < arr2)  
        printf("그러므로 arr2가 더 크다.\n");  
    else  
        printf("그러므로 arr1이 더 크다.\n");  
  
return 0; }
```

9. "What doesn't kill me makes me strong"이란 문자열에서 me가 몇 번 사용되는지 검출하는 프로그램을 작성하시오.

10. 구조체에 이름과 급여를 받도록 하고
이름은 마음대로 짓고 급여는 rand()를 활용
급여의 평균을 출력하고 가장 높은 한 사람의 이름과 급여를 출력하시오.
(값이 같을 수 있음에 유의해야 한다)

11. vi 편집기에서 3줄을 복사하고자 할 때 명령어는 ? > y3y

12. 리눅스 명령어에 대해 5개 이상 아는대로 작성하시오.

Pwd 현재 위치 확인 vi 파일명.c 메모장 생성 gcc 파일명.c 파일 생성 mkdir 파일명 파일 생성 rm -rf 파일명 파일 삭제

13. vi 편집기에서 7줄을 지우고자할 때 어떤 명령어를 사용하는가 ? >d7d

14. C언어를 사용할 때 중요하게 바라보는 메모리 구조에 대해 설명하시오.

15. 메모리 계층 구조에 대해 설명하시오.

16. int (* (* test(int a))(void))(int);

위의 프로토타입을 해석하시오.

int (*)(int) (*)(void) test(int a)

리턴 : int (*)(int)(*)(void)

이름 : test

인자 : int a

17. void (* signal(int num, void (*handler)(int)))(int val);

위의 프로토타입에 대해 설명하시오.

void (*)(int val) signal(int num, void (*handler)(int))

리턴 : void (*)(int val)

이름 : signal

인자 : (int num, void (*handler)(int))

18. 1, 3, 5, 7, 9, ... 의 수열과

1, 1, 2, 3, 5, 8, 13, ... 의 수열이 존재한다.

첫 번째 수열에서 홀수번째 index들과

두 번째 수열에서 짝수번째 index들의 합을 구하시오.

(index는 40까지만 진행한다)

19. Interrupt란 무엇인가 ?

20. gdb 명령어에서 x, p의 용도는 각각 무엇인가 ?

또 bt라는 명령어는 무엇을 의미하는가 ?

bt : 오류가 발생한 함수를 역으로 찾아간다.