

TI DSP, MCU 및 Xilinx Zynq FPGA 프로그램 전문가 과정

강사 – Innova Lee(이상훈)

gcccompil3r@gmail.com

학생 – 은태영

Zero_bird@naver.com

목차

- 문제 은행_1858
- 스택 함수의 push / pop

문제 은행_1858

float (* (* test(void (*p)(void)))(float (*)(int, int)))(int, int)
위와 같은 프로토타입의 함수가 구동되도록 프로그래밍 하시오.

float (*)(int, int) (*) (float (*)(int, int)) test(void (*p)(void))

Return : float (*)(int, int) (*) (float (*)(int, int))

Name : test

Parameter : void (*p)(void)

문제 은행_1858

Test의 Return 값 : float (*)(int, int) (*p) (float (*)(int, int))

Return : float (*) (int, int)

Name : p

Parameter : (float (*)(int, int))

변환 : float (* (* return2)(float (*)(int, int)))(int, int))

문제 은행_1858

```
tewill@tewill-B85M-D3H: ~/my_proj/lesson007
#include <stdio.h>

float return1(int n1, int n2)
{
    float res = 0;
    res = (n1 + n2) * 0.33;
    printf("return1 : %f\n", res);
    return res;
}

float (* return2(float *p(int, int)))(int, int)
{
    printf("return2 : %f\n", p(2,3));
    return p;
}

void parameter(void)
{
    printf("parameter print\n");
}

float (* (* test(void (*p)(void)))(float (*)(int, int)))(int, int)
{
    p();
    return return2;
}

int main(void)
{
    test(parameter)(return1)(4,5);
    return 0;
}
```

1,1 All

```
tewill@tewill-B85M-D3H: ~/my_proj/lesson007
tewill@tewill-B85M-D3H:~/my_proj/lesson007$ ./debug
parameter print
return1 : 1.650000
return2 : 0.000000
return1 : 2.970000
tewill@tewill-B85M-D3H:~/my_proj/lesson007$
```

문제 은행_1858

- 문제점 : Return2 본문 중, Return1의 return 값을 받지 못함.
- 해결 방안 :

스택 함수의 push / pop

Main

```
#include <malloc.h>
#include <stdlib.h>

#define EMPTY 0

struct node{
    int data;
    struct node *link;
};
typedef struct node Stack;
```

```
int main(void)
{
    Stack *top = EMPTY;
    push(&top,10);
    push(&top,20);
    push(&top,30);
    printf("%d\n", pop(&top));
    printf("%d\n", pop(&top));
    printf("%d\n", pop(&top));
    printf("%d\n", pop(&top));
    return 0;
}
```

Push

```
void push(Stack **top, int data)
{
    Stack *tmp;
    tmp = *top;
    *top = get_node();
    (*top)->data = data;
    (*top)->link = tmp;
}
```

스택 함수의 push / pop

Get_Node

```
Stack *get_node()
{
    Stack *tmp;
    tmp = (Stack*)malloc(sizeof(Stack));
    tmp->link=EMPTY;

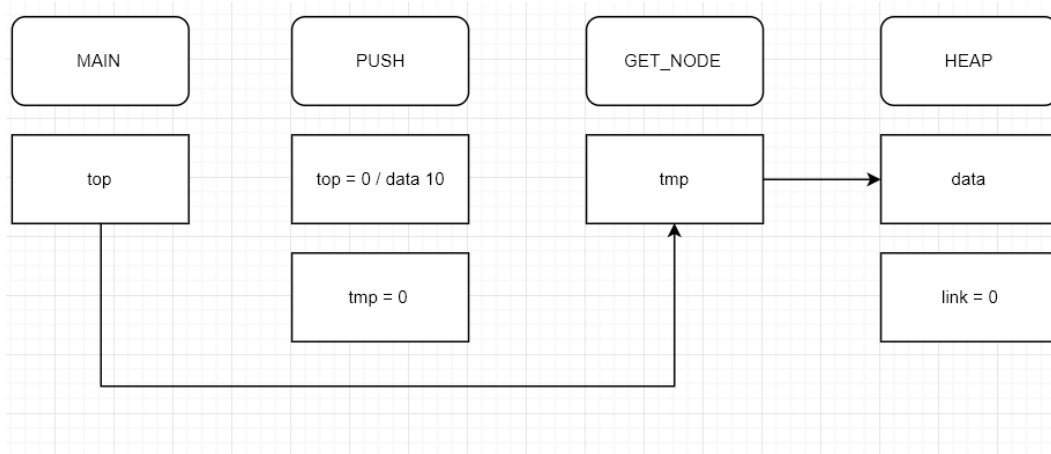
    return tmp;
}
```

Pop

```
int pop(Stack **top)
{
    Stack *tmp;
    int num;
    tmp = *top;
    if(*top == EMPTY)
    {
        printf("Stack is empty!!!\n");
        return 0;
    }
    num = tmp->data;
    *top = (*top)->link;
    free(tmp);
    return num;
}
```


스택 함수의 push / pop

Main에서 node에 대한 top 포인터를 생성하고, 값에 0을 넣는다.



Push 함수 호출 후, top과 data의 값을 받아 온다.

Node에 대한 Tmp 포인터를 생성하고, 그 안에 top의 값 0을 넣는다.

Get_node 함수를 호출한다.

Node 에 대한 Tmp 포인터를 생성하고, 그 안에 node를 malloc하여 만든 Heap 의 주소를 넣는다.

스택 함수의 push / pop

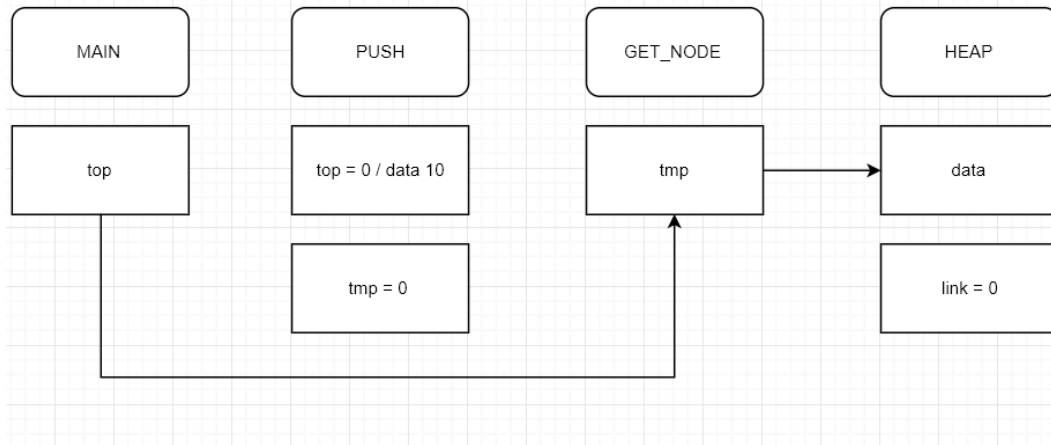
생성한 Heap의 link를 0으로 초기화 한다.

Get_node 에서 return을 통해 tmp의 값을 top의 값에 넣는다.

이를 통해 top은 Heap의 주소를 나타낸다.

Top의 값을 통하여 heap안에 있는 data에 push에서 받아온 data 10을 넣는다.

Top을 통해 heap안에 있는 link에 push의 tmp값 0을 집어 넣는다.



스택 함수의 push / pop

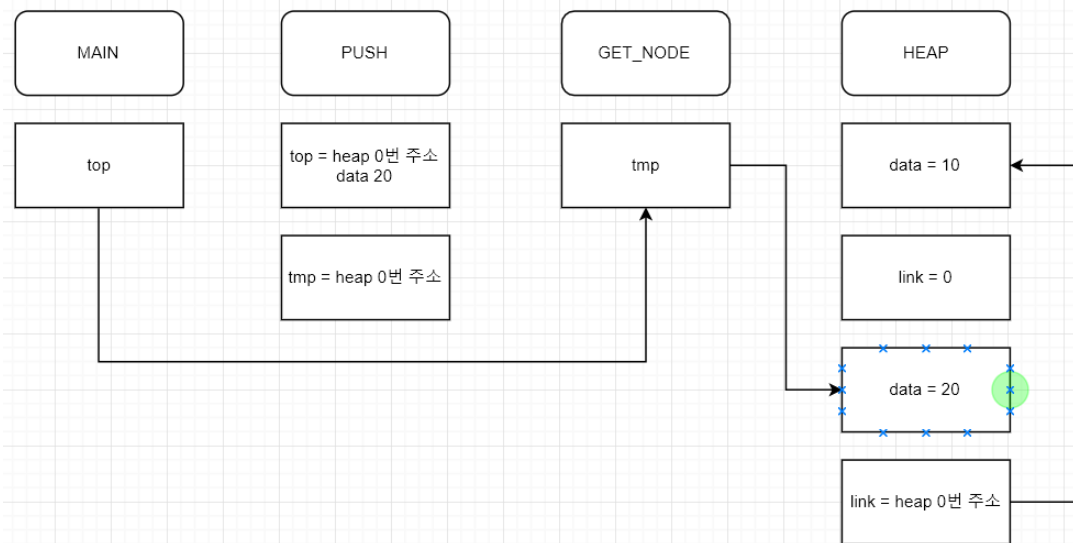
Main으로 돌아와서 2번째 push를 실행한다.

Top과 지정된 data 값을 받아온다.

Tmp 포인터를 생성 후, top이 가리키는 값, heap 0번 주소를 집어 넣는다.

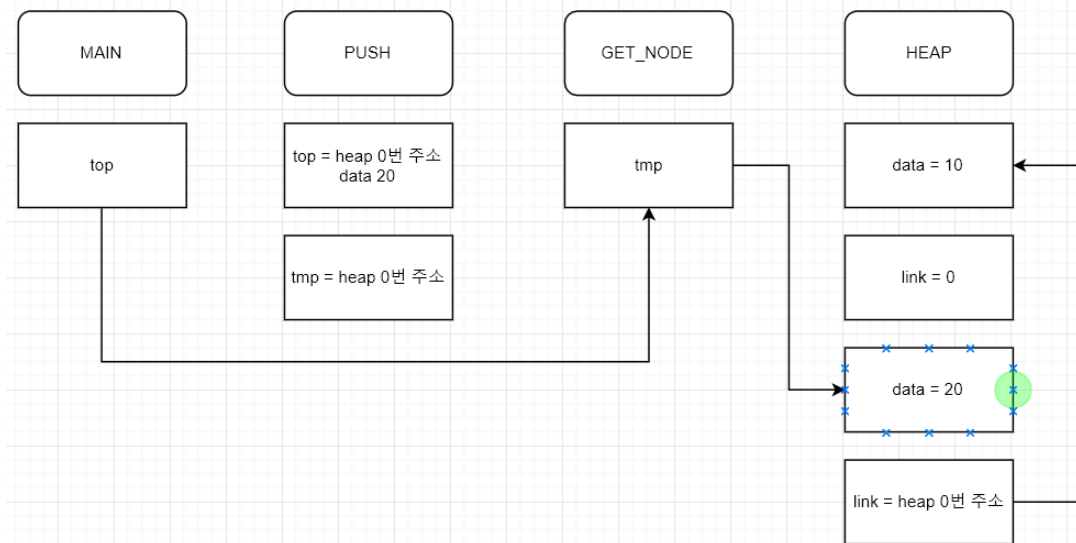
Get_node로 이동하여, tmp 포인터를 만들고, Heap으로 만들어진 node의 주소를 집어 넣는다.

Heap의 1번 node의 link를 0으로 초기화 한다.



스택 함수의 push / pop

Get_node의 return 값을 통하여 top의 값을 Heap 1번 주소로 변경한다.



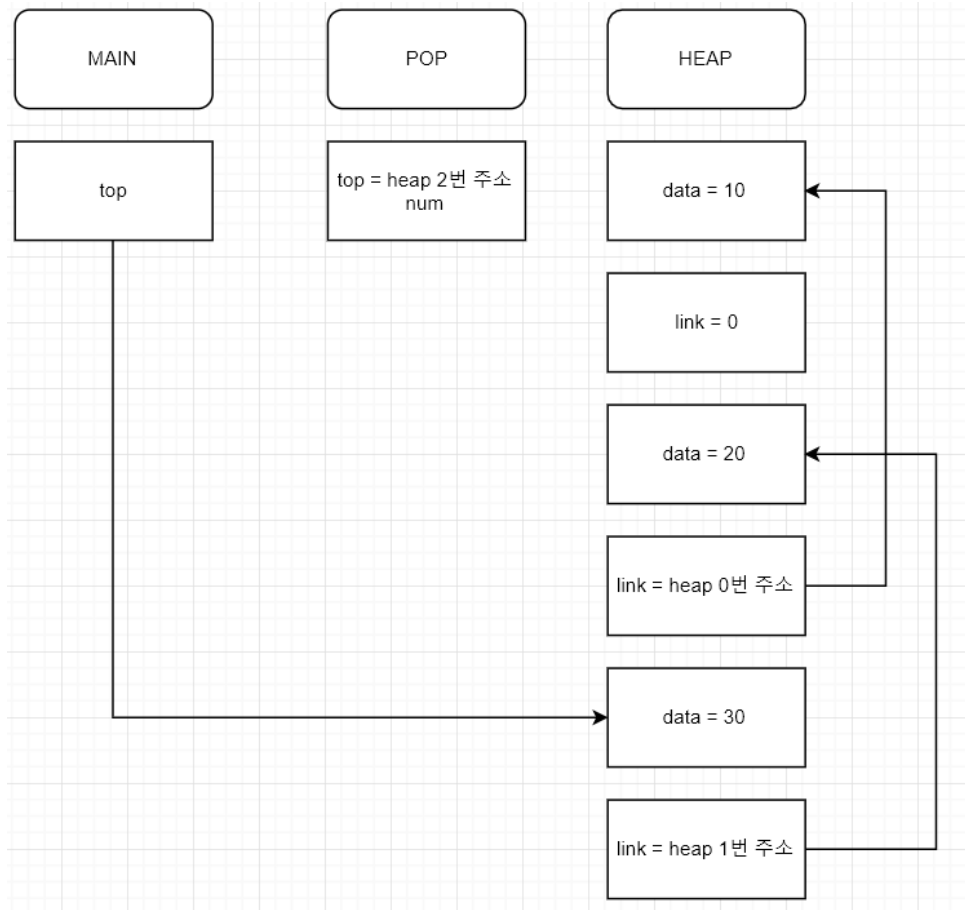
Top을 통해, Heap 1번 data 값에 20을 넣는다.

그 후, link에 push의 tmp 값, Heap의 0번 주소를 집어 넣는다.

이를 통하여, heap의 link는 0번 주소를 가리키게 된다.

이를, push 가 반복되는 만큼 진행을 한다.

스택 함수의 push / pop



Pop 함수를 실행한다.

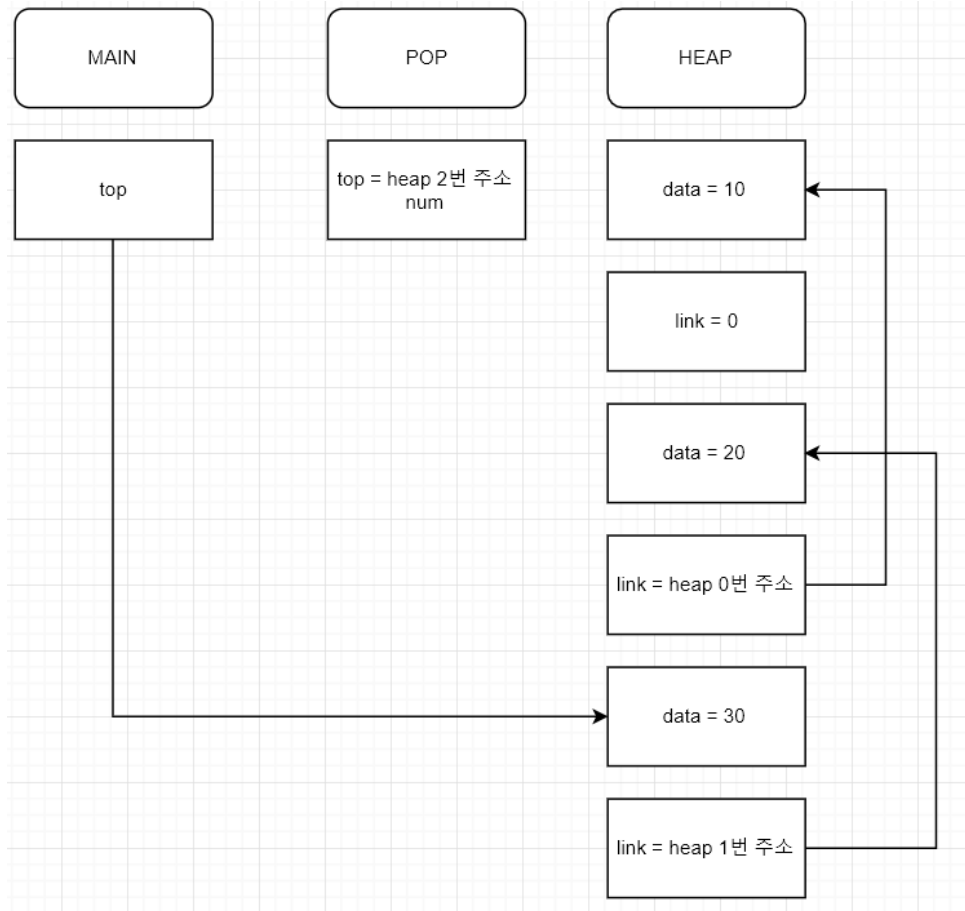
Main의 top을 인자로 받고, node의 포인터 tmp와 값을 저장할 num을 만든다.

Tmp에 top의 값을 저장한다.

If문을 통하여 top이 가리키는 값이 0인지 확인을 한다.

아닐 경우, tmp가 가리키는 Heap 2번 주소 안에 있는 data를 준비한 num에 넣는다.

스택 함수의 push / pop

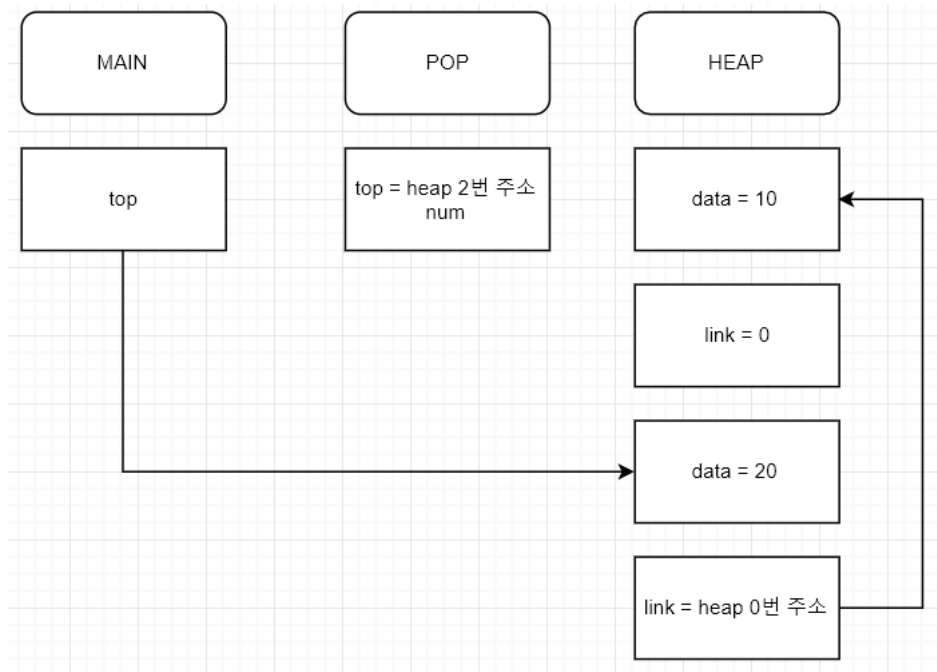


top의 값을 link가 가리키는 1번 주소로 변경한다.

미리 저장해 두었던 tmp를 이용하여 Heap으로 잡혀있던 2번 주소를 free를 통해 제거한다.

Return 을 통하여 받아 둔 num 값을 내보낸다.

스택 함수의 push / pop



해당 과정을 거쳐서, top은 Heap의 1번 주소를 나타내며, pop을 계속할 경우 마지막 남아있는 데이터까지 이동하게 된다.

마지막 데이터 이후, top은 heap의 0번 주소에 있던 0 값을 받아오게 되며, 그 후로 pop을 호출 시, "Stack is empty!!!" 가 출력되며 pop을 빠져 나온다.