TI DSP, MCU 및 Xilinx Zynq FPGA 프로그래밍 전문가 과정

강사 - Innova Lee(이상훈) gcccompil3r@gmail.com 학생 - GJ (박현우) uc820@naver.com

1. 복습 - (5일차)

- 함수 포인터!!
- ✓ 함수 프로토타입이란?

리턴, 함수명, 인자에 대한 기술서이다.

그렇다면, void (* signal(int signum, void (* handler)(int)))(int); 위 함수에 대한 프로토타입은 뭘까?

이전에 공부했던 int (*p)[2]; -> int (*)[2] p 배열포인터 // int *p[2] 포인터 배열과는 다름

리턴: void (*)(int)

함수명: signal

인자: int signum 과 void(* handler)(int)

void (*p)(void): void를 리턴하고 void를 인자로 취하는 함수의 주소 값을 저장할 수 있는 변수 p

1. 복습 - (6일차)

- 복잡한 함수 포인터!!
- 1) int (* aaa(void))[2] // 실제 문법
- → int (*)[2] aaa(void) // 인간이 보기 편하게 만듬

풀이: int형 배열 2개짜리 묶음주소를 반환하고 void를 인자로 취하는 함수 aaa

- 2) int (*(* bbb(void))(void))[2]; // 실제 문법
- → int (*)[2](*)(void) bbb(void) // 인간이 보기 편하게 만듬

풀이: int형 배열 2개짜리 묶음주소를 반환하고 인자로 void를 취하는 함수 포인터를 리턴하고 void를 인자로 취하는 함수 bbb

- 3) int (*(*(*p[][2])(void))(void))[2] // 실제 문법
- → int (*)[2] (*)(void) (*)(void) p[][2]
- → int (*)[2] (*)(void) bbb(void) p[][2]

풀이: int형 배열2개 짜리 묶음주소를 반환하고 void를 인자로 취하는 함수의 주소 값을 리턴하고 인자로 void를 취하는 함수의 주소 값을 반환하는 배열변수 p

1. 복습 - (6일차)

- 그렇다면!! 대체 이렇게 복잡한 함수 포인터를 왜 쓸까?
- 1. 비동기 처리
- 2. HW 개발 관점에서 인터럽트
- 3. 시스템 콜(유일한 SW 인터럽트)

여기서 인터럽트(SW, HW)은 사실상 모두 비동기 동작에 해당한다.

→ 결국 1번(비동기 처리)가 핵심이라는 뜻이다.

그렇다면, 비동기 처리는 뭘까?

→ 기본적으로 동기 처리라는 것은 송신과 수신의 쌍방 합의하에 달성된다 (ex. 휴대폰 전화 통화.)

반면, 비동기 처리는 이메일 또는 카카오톡 메신저 등에 해당한다.

그래서 그냥 일단 송신해두면 상대방이 바쁘면 못 보지만

바쁘지 않은 상황이라면 메시지를 보고 답변을 준다.

이처럼 언제 어떤 이벤트가 발생할지를 알 수 없는 것들은 다루는 것이 비동기이며 함수 포인터를 사용하는 이유이다.

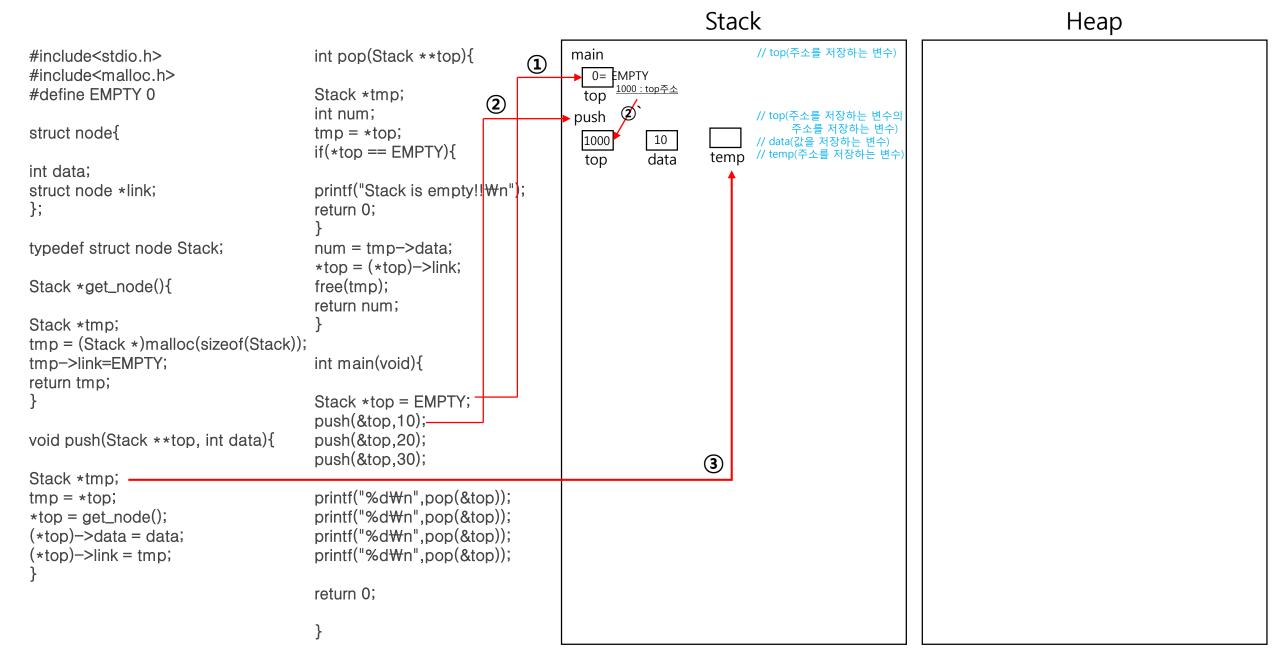
✓ 즉, 사람도 임기응변을 잘 해야 하듯이, 컴퓨터 관점에서도 임기응변을 잘하도록 만들어주는 것이 바로 함수 포인터이다. (결론: 비동기 처리 – 함수 포인터)

2. 함수 포인터 문제

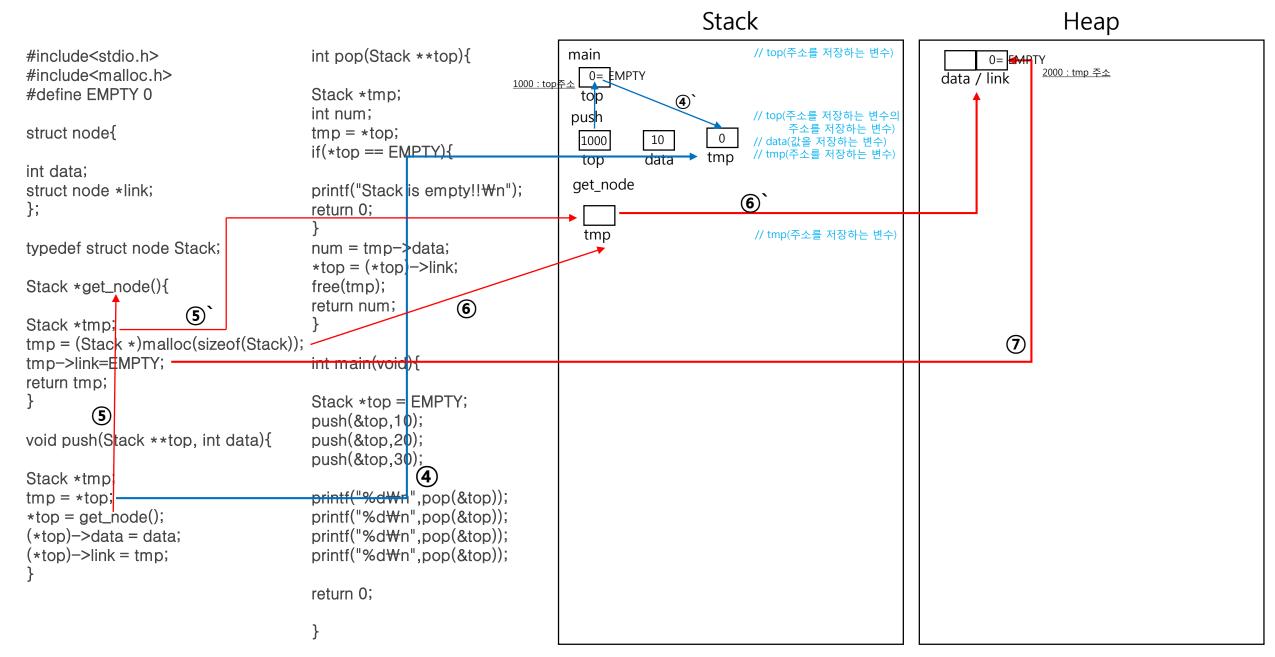
float (* (* test(void (*p)(void)))(float (*)(int, int)))(int, int) 위와 같은 프로토타입의 함수가 구동되도록 프로그래밍 하시오.

```
1 #include<stdio.h>
                                                                                       hyunwoopark@hyunwoopark-P65-P67SG:~/hw$ ./a.out
                                                                                       aaa called
 3 //void (*p)(void)
                                                                                      test2 res = 9.000000
  4 void aaa(void){
                                                                                       main res = 10.000000
                                                                                       hyunwoopark@hyunwoopark-P65-P67SG:~/hw$
        printf("aaa called\n");
 8 }
9 float bbb(int n1, int n2){
        return n1 + n2;
 12 }
 14 //float (*)(int,int) (*)(float (*)(int,int))
15 //float (*)(int, int) ccc(float(*p)(int,int))
16 float (* test2(float(*p)(int,int)))(int,int){
17
18
19
20
21
22
23
24
25
26
        float res;
        res = p(3,6);
        printf("test2 res = %f\n",res);
        return bbb;
27 //float (*(* test(void (*p)(void)) )(float (*)(int,int)) )(int,int)
28 //float (*)(int,int) (*)(float (*)(int,int)) test(void (*p)(void))
29 float (*(* test(void (*p)(void)))(float (*)(int,int)))(int,int){
31
32
        p(); // aaa called;
33
34
        return test2;
35 }
36
37 int main(int agrc, char **argv){
38
39
40
41
42
43
44
        float res;
        res = test(aaa)(bbb)(4,6);
        printf("main res = %f\n",res);
        return 0;
45 }
"func_pointer.c" 45L, 712C
                                                                                모두
```

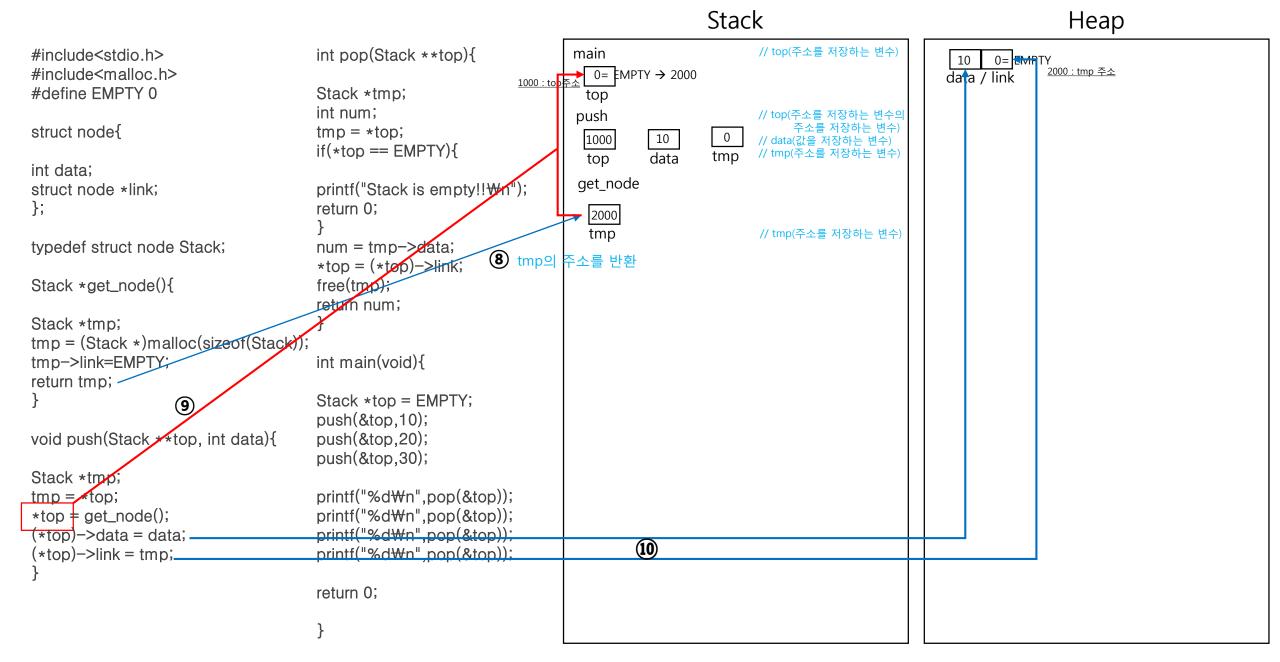
3. 스택(Stack) 코드 이해(그림 그리기) - (1)



3. 스택(Stack) 코드 이해(그림 그리기) - (2)

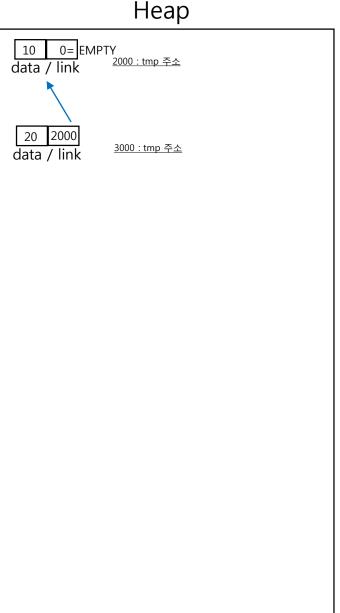


3. 스택(Stack) 코드 이해(그림 그리기) - (3)



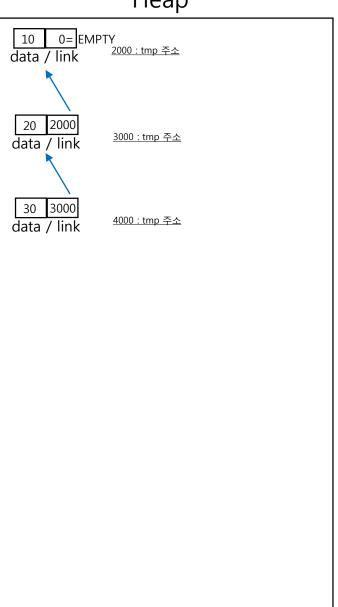
3. 스택(Stack) 코드 이해(그림 그리기) - (4)

```
Stack
                                                                                                    // top(주소를 저장하는 변수)
                                       int pop(Stack **top){
                                                                          main
#include<stdio.h>
#include<malloc.h>
                                                                             0 = EMPTY \rightarrow 2000 \rightarrow 3000
                                                                   1000 : top뉴소
#define EMPTY 0
                                       Stack *tmp;
                                                                            top
                                       Int num;
                                                                                                   // top(주소를 저장하는 변수의
                                                                          push
                                       Tmp = *top;
struct node{
                                                                                              0
                                                                            1000
                                                                                     10
                                       If(*top == EMPTY){
                                                                                                   // tmp(주소를 저장하는 변수)
                                                                                             tmp
                                                                                    data
                                                                            top
int data;
                                                                          get_node
struct node *link;
                                       Printf("Stack is empty!!₩n");
                                       Return 0;
                                                                            2000
                                                                            tmp
                                                                                                    // tmp(주소를 저장하는 변수)
typedef struct node Stack;
                                       Num = tmp -> data;
                                                                          push
                                       *top = (*top) - > link;
                                       Free(tmp);
Stack *get_node(){
                                                                                    20
                                                                                             2000
                                                                           1000
                                       Return num;
                                                                                             tmp
                                                                                    data
                                                                           top
Stack *tmp;
                                                                          get_node
                                            ① ~ 10과 같은 방식으로 반복
tmp = (Stack *)malloc(sizeof(Stack));
                                                                           3000
                                       Int main(void){
tmp->link=EMPTY;
                                                                           tmp
return tmp;
                                       Stack *top = EMPTY;
                                       push(&top,10);
void push(Stack **top, int data){
                                       push(&top,20);
                                       push(&top,30);
Stack *tmp;
                                       printf("%d\foralln",pop(&top));
tmp = *top;
*top = get_node();
                                       printf("%d\foralln",pop(&top));
                                       printf("%d\foralln",pop(&top));
(*top)->data = data;
(*top)->link = tmp;
                                       printf("%dWn",pop(\&top));
                                       return 0;
```



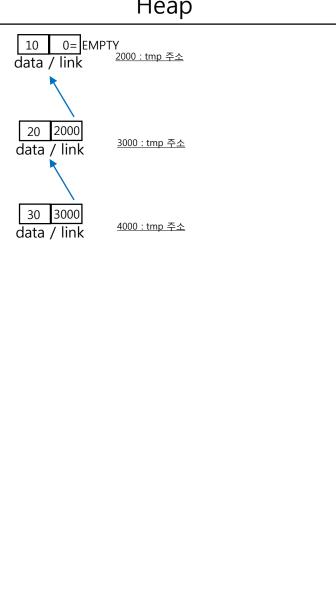
3. 스택(Stack) 코드 이해(그림 그리기) - (5)

```
Stack
                                                                                                    // top(주소를 저장하는 변수)
                                       int pop(Stack **top){
                                                                          main
#include<stdio.h>
#include<malloc.h>
                                                                             0 = \text{EMPTY} \rightarrow 2000 \rightarrow 3000 \rightarrow 4000
                                                                   1000 : top뉴소
#define EMPTY 0
                                       Stack *tmp;
                                                                            top
                                       Int num;
                                                                                                    // top(주소를 저장하는 변수의
                                                                          push
                                       Tmp = *top;
struct node{
                                                                                               0
                                                                            1000
                                                                                     10
                                       If(*top == EMPTY){
                                                                                                    // tmp(주소를 저장하는 변수)
                                                                                             tmp
                                                                                     data
                                                                            top
int data;
                                                                           get_node
struct node *link;
                                       Printf("Stack is empty!!₩n");
                                       Return 0;
                                                                            2000
                                                                            tmp
                                                                                                    // tmp(주소를 저장하는 변수)
typedef struct node Stack;
                                       Num = tmp -> data;
                                                                          push
                                       *top = (*top)->link;
                                       Free(tmp);
Stack *get_node(){
                                                                                     20
                                                                                             2000
                                                                           1000
                                                                                             tmp
                                       Return num;
                                                                                    data
                                                                           top
Stack *tmp;
                                                                          get node
tmp = (Stack *)malloc(sizeof(Stack));
                                                                           3000
                                       Int main(void){
tmp->link=EMPTY;
                                                                           tmp
return tmp;
                                       Stack *top = EMPTY;
                                                                          push
                                       push(&top.10);
                                                                                             3000
                                                                                     30
                                                                           1000
void push(Stack **top, int data){
                                       push(&top,20);
                                                                                             tmp
                                                                                    data
                                                                           top
                                       push(&top,30);
                                               ① ~ ⑩과 같은 방식으로 반복
                                                                           get node
Stack *tmp;
                                       printf("%dWn",pop(&top));
tmp = *top;
                                                                            4000
                                       printf("%dWn",pop(&top));
*top = get_node();
                                                                            tmp
                                       printf("%d\foralln",pop(&top));
(*top)->data = data;
(*top)->link = tmp;
                                       printf("%dWn",pop(\&top));
                                       return 0;
```

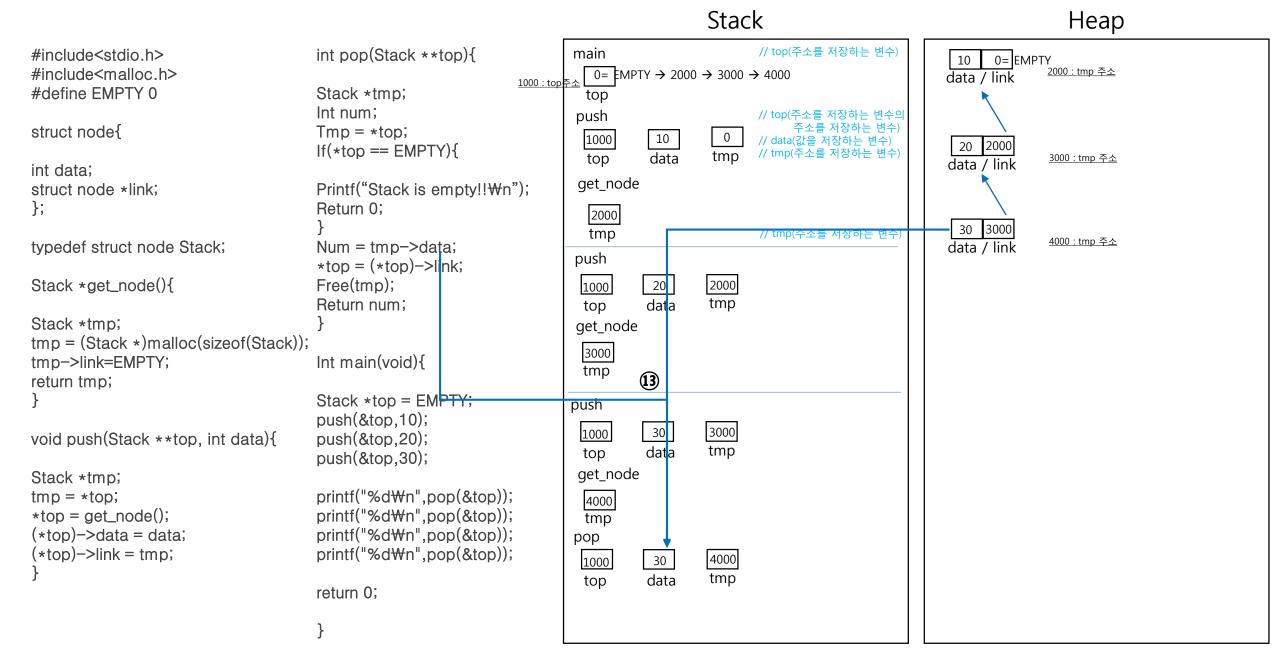


3. 스택(Stack) 코드 이해(그림 그리기) - (6)

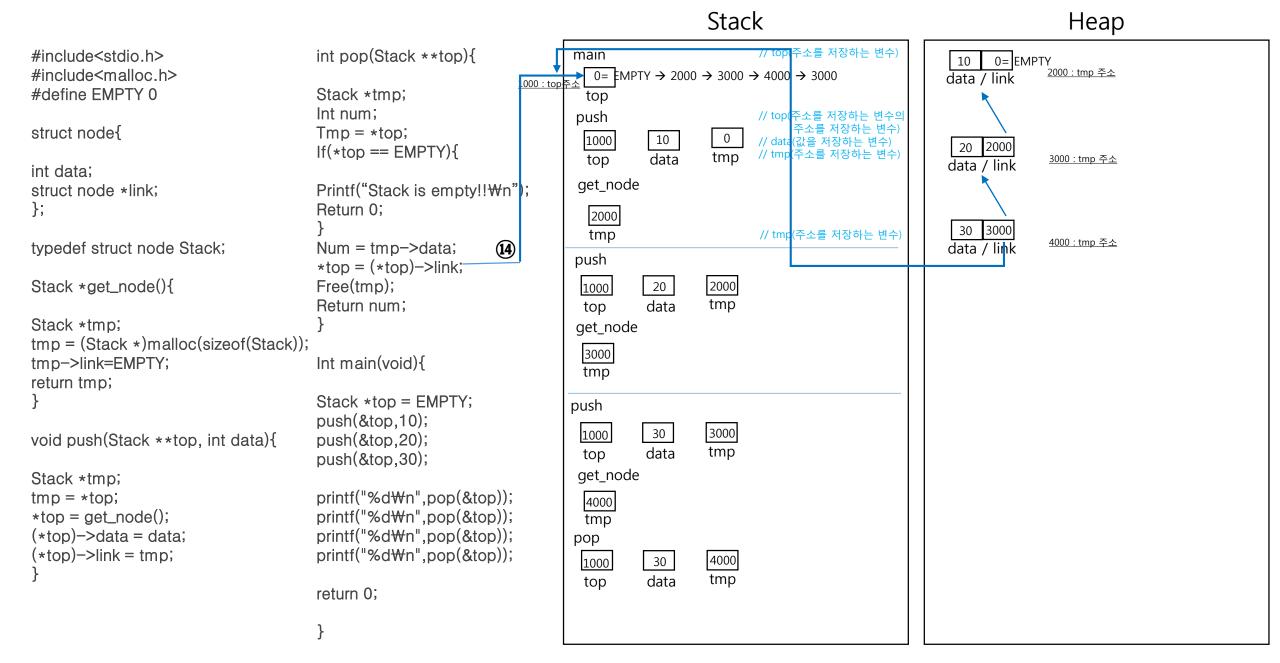
```
Stack
                                                                                                    // top(주소를 저장하는 변수)
                                       int pop(Stack **top){
                                                                           main
#include<stdio.h>
#include<malloc.h>
                                                                             0 = EMPTY \rightarrow 2000 \rightarrow 3000 \rightarrow 4000
                                                                   1000 : top뉴소
#define EMPTY 0
                                       Stack *tmp;
                                                                            top
                                       Int num;
                                                                           push
                                       Tmp = *top;
struct node{
                                                                                     10
                                                                                               0
                                                                            1000
                                       If(*top == EMPTY){
                                                                                              tmp
                                                                                     data
                                                                            top
int data;
                                                                           get_node
struct node *link;
                                       Printf("Stack is empty!!₩n");
                                       Return 0;
                                                                             2000
                                                                             tmp
                                                                                                     // mp(주소를 저장하는 변수)
typedef struct node Stack;
                                       Num = tmp->data;
                                                                           push
                                       *tpp = (*top) - > link;
                                       Free(tmp);
Stack *get_node(){
                                                                                     20
                                                                                             2000
                                                                            1000
                                       Return num;
                                                                                             tmp
                                                                                     data
                                                                            top
Stack *tmp;
                                                                           get node
                                                                                         (12)
tmp = (Stack *)malloc(sizeof(Stack));
                                                                            3000
                                       Int main(void){
tmp->link=EMPTY;
                                                                            tmp
return tmp;
                                       Stack *top = EMPTY;
                                                                          push
                                       push(&top,10);
                                                                                             3000
tmp
                                                                                     30
                                                                            1000
void push(Stack **top, int data){
                                       push(&top,20);
                                                                                     data
                                                                            top
                                       push(&top,30);
                                                                           get node
Stack *tmp;
                                       printf("%d\n",pop(&top)); (ii)
tmp = *top;
                                                                            4000
                                       printf("%d₩n",pop(&top));
*top = get_node();
                                                                            tmp
                                       printf("%d\foralln",pop(&top));
(*top)->data = data;
                                                                           pop
                                       printf("%d₩n",pop(&top));
(*top)->link = tmp;
                                                                                             4000
                                                                            1000
                                                                                             tmp
                                                                                     data
                                                                            top
                                       return 0;
```



3. 스택(Stack) 코드 이해(그림 그리기) - (7)

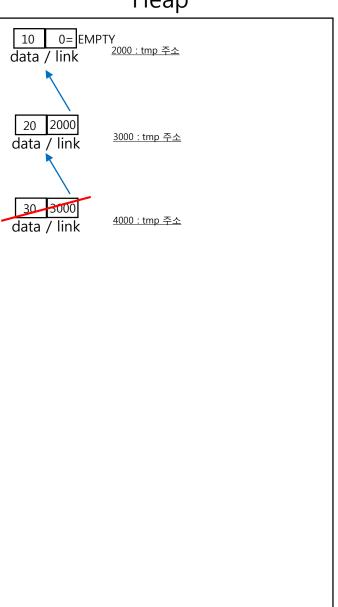


3. 스택(Stack) 코드 이해(그림 그리기) - (8)



3. 스택(Stack) 코드 이해(그림 그리기) - (9)

```
Stack
                                                                                                      // top(주소를 저장하는 변수)
                                        int pop(Stack **top){
                                                                            main
#include<stdio.h>
#include<malloc.h>
                                                                              0 = \text{EMPTY} \rightarrow 2000 \rightarrow 3000 \rightarrow 4000 \rightarrow 3000
                                                                    1000 : top뉴소
#define EMPTY 0
                                        Stack *tmp;
                                                                             top
                                        Int num;
                                                                            push
                                                                                                      // top(주소를 저장하는 변수의
                                        Tmp = *top;
struct node{
                                                                                                 0
                                                                             1000
                                                                                       10
                                        If(*top == EMPTY){
                                                                                                      // tmp(주소를 저장하는 변수)
                                                                                               tmp
                                                                                      data
                                                                             top
int data;
                                                                            get_node
                                        Printf("Stack is empty!!₩n");
struct node *link;
                                        Return 0;
                                                                              2000
                                                                              tmp
                                                                                                      // tmp(주소를 저장하는 변수)
typedef struct node Stack;
                                        Num = tmp -> data;
                                                                            push
                                        *top = (*top) - > link;
                                                                  (15)
                                        Free(tmp);__
Stack *get_node(){
                                                                             1000
                                                                                      20
                                                                                               2000
                                        Return num;
                                                                                               tmp
                                                                                      data
                                                                             top
Stack *tmp;
                                                                            get node
tmp = (Stack *)malloc(sizeof(Stack));
                                                                             3000
                                        Int main(void){
tmp->link=EMPTY;
                                                                             tmp
return tmp;
                                        Stack *top = EMPTY;
                                                                           push
                                        push(&top,10);
                                                                                               3000
                                                                                      30
                                                                             1000
void push(Stack **top, int data){
                                        push(&top,20);
                                                                                               tmp
                                                                                      data
                                                                             top
                                        push(&top,30);
                                                                            get node
Stack *tmp;
                                        printf("%d\foralln",pop(&top));
tmp = *top;
                                                                             4000
                                        printf("%dWn",pop(&top));
*top = get_node();
                                                                             tmp
                                        printf("%d₩n",pop(&top));
(*top)->data = data;
                                                                            pop
                                        printf("%d₩n",pop(&top));
(*top)->link = tmp;
                                                                             TOOO
                                                                                               tmp
                                                                                      data
                                                                             top
                                        return 0;
```



3. 스택(Stack) 코드 이해(그림 그리기) - (10)

```
int pop(Stack **top){
#include<stdio.h>
#include<malloc.h>
                                                                 1000 : top뉴소
#define EMPTY 0
                                      Stack *tmp;
                                      Int num;
                                      Tmp = *top;
struct node{
                                      If(*top == EMPTY){
int data;
                                      Printf("Stack is empty!!₩n");
struct node *link;
                                      Return 0;
typedef struct node Stack;
                                      Num = tmp -> data;
                                      *top = (*top)->link;
                                      Free(tmp);
Stack *get_node(){
                                      Return num;
Stack *tmp;
tmp = (Stack *)malloc(sizeof(Stack));
                                      Int main(void){
tmp->link=EMPTY;
return tmp;
                                      Stack *top = EMPTY;
                                      push(&top,10);
void push(Stack **top, int data){
                                      push(&top,20);
                                      push(&top,30);
Stack *tmp;
                                      printf("%dWn",pop(&top));
tmp = *top;
                                      printf("%dWn",pop(&top));
*top = get_node();
                                      printf("%d\foralln",pop(&top));
(*top)->data = data;
                                      printf("%d₩n",pop(&top));
(*top)->link = tmp;
                                      return 0;
```

Stack // top(주소를 저장하는 변수) $0 = \text{EMPTY} \rightarrow 2000 \rightarrow 3000 \rightarrow 4000 \rightarrow 3000 \rightarrow 2000 \rightarrow 0$

// tmp(주소를 저장하는 변수)

top // top(주소를 저장하는 변수의 push 0 1000 10 tmp data top

2000

get_node

main

tmp

top

push 1000

20 data

2000 tmp

3000

tmp

tmp

get node 3000 tmp

push 30 1000

data top

get node

4000 tmp

pop

pop

1000 30 top pop

data

① ~ ⑤와 같은 방식으로 반복

