Xilinx Zynq FPGA, TI DSP, MCU기반의 프로그래밍 및 회로 설계 전문가 과정

강사 - Innov (이상훈)
gcccompil3r@gmail.com
학생 - 이유성
dbtjd1102@naver.com

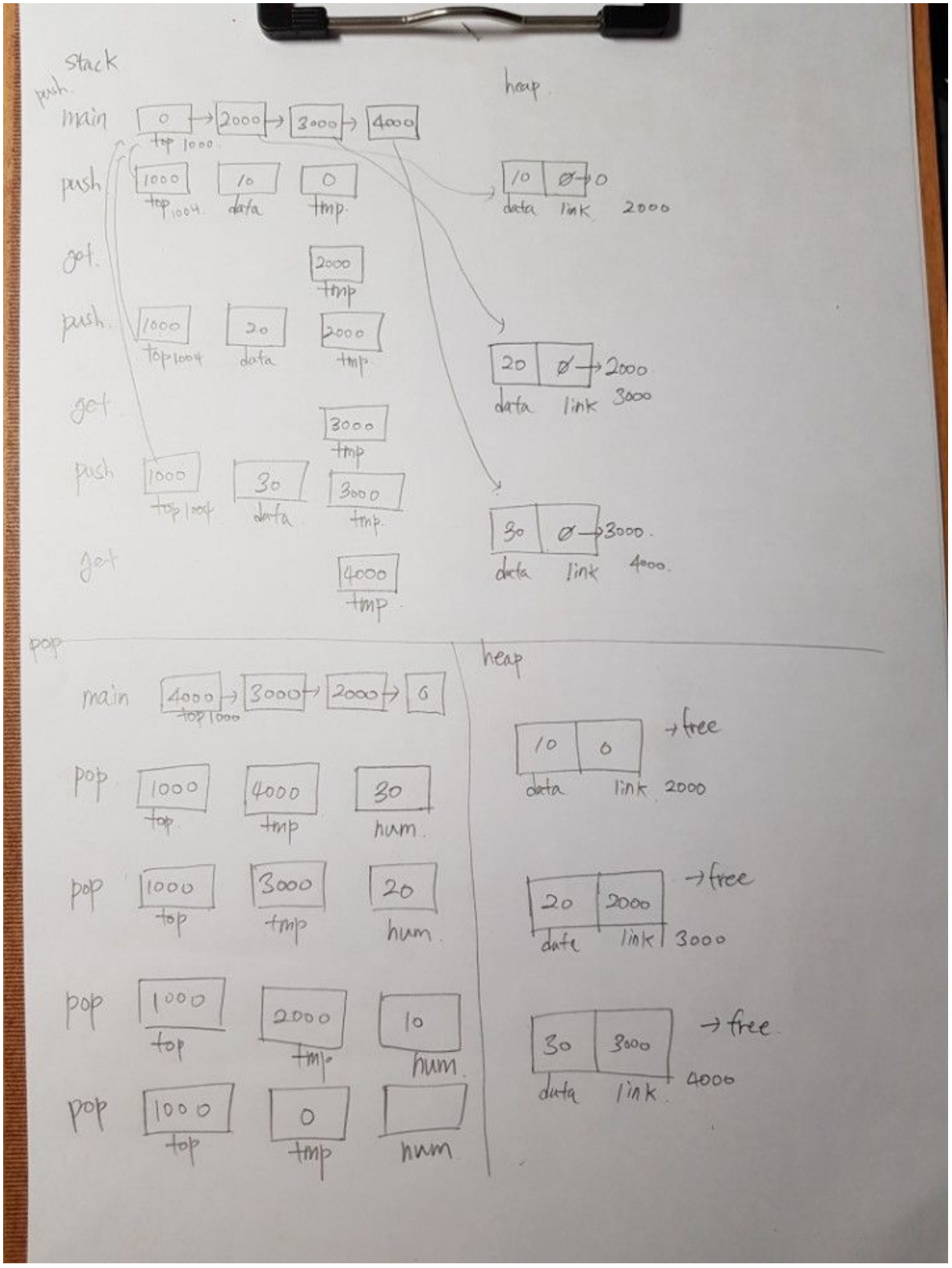
8일차 복습

stack 후입선출 queue 선입선출 (이진트리)

Stack

```
#include <stdio.h>
#include<malloc.h>
#define EMPTY 0

struct node{

        int data ;
        struct node *link;
};

typedef struct node Stack;



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

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

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;
}

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;
}
```


stack
heap
push
main
0
2000
3000
4000
top 1000
push
1000
top 1004
10
data
0
tmp
10
0
data
link
2000
get
2000
tmp
push
1000
top 1004
20
data
2000
tmp
20
2000
data
link
3000
get
3000
tmp
push
1000
top 1004
30
data
3000
tmp
30
3000
data
link
4000
get
4000
tmp
pop
heap
main
4000
3000
2000
0
top 1000
pop
1000
top
4000
tmp
30
num
10
0
data
link
2000
free
pop
1000
top
3000
tmp
20
num
20
2000
data
link
3000
free
pop
1000
top
2000
tmp
10
num
30
3000
data
link
4000
free
pop
1000
top
0
tmp
num

Queue

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

#define EMPTY 0



struct node
{
	int data;
	struct node *link;

};
typedef struct node Queue;



Queue *get_nude()
{
	Queue *tmp;
	tmp = (Queue *)malloc(sizeof(Queue));
	tmp->link =EMPTY;
	return tmp;

}


void dequeue(Queue **head , int data)
{



}



void enqueue(Queue **head , int data)
{


	if(*head ==EMPTY)
	{
		*head = get_nude();
		(*head)->data = data;
		return ;
	}
	enqueue(&(*head)->link,data);

	printf("test\n"); //재귀함수 호출하는 만큼 test
```

```
}

void print_queue(Queue *head)
{
	Queue *tmp = head;

	while(tmp)
	{
		printf("%d\n", tmp->data);
		tmp = tmp->link;
		}

}

int main(void)
{
	Queue *head  =EMPTY ;
	enqueue(&head,10);
	enqueue(&head,20);
	enqueue(&head,30);

	print_queue(&head)


	return 0;

}
```