TI DSP, MCU 및 Xilinx Zynq FPGA 프로그래밍 전문가 과정

2018.03.08
11 일차
강사 – Innova Lee(이상훈)
gcccompil3r@gmail.com

학생 – 신민철
akrn33@naver.com

이진트리

```
#include<stdio.h>
#include<malloc.h>
#include<stdlib.h>
#define EMPTY 0
struct node{
	int data;
	struct node* left;
	struct node* right;
};

typedef struct node tree;


tree* chg_node(tree* root)
{
	tree* tmp = root;

	if(!root->right)
		root = root->left;
	else if(!root->left)
```

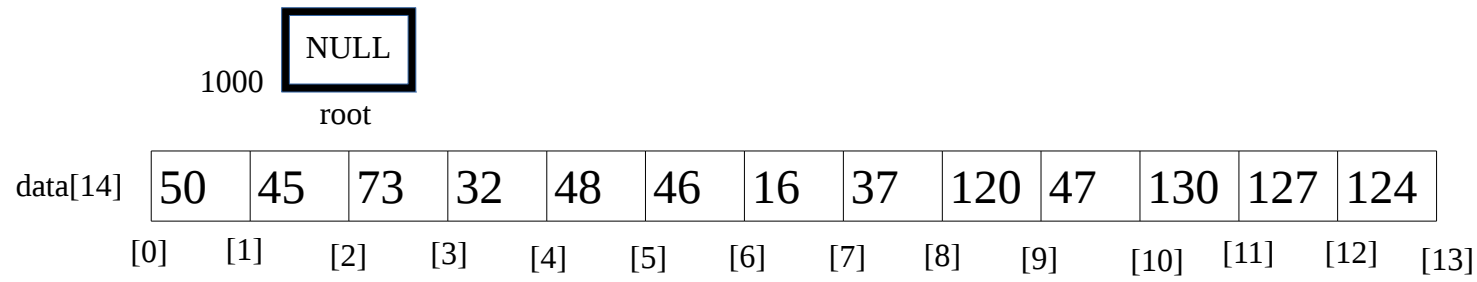
```
            root = root->right;

        free(tmp);

        return root;
}

tree* get_node()
{
        tree* tmp;
        tmp = (tree*)malloc(sizeof(tree));
        return tmp;
}

tree* find_max(tree* root, int* data)
{
        if(root->right)
                root->right = find_max(root->right, data);
        else
        {
                *data = root->data;
                root = chg_node(root);
        }
        return root;
}

tree* delete_tree(tree* root, int data)
{
        int num;
        tree* tmp;
        if(root == NULL)
        {
                printf("Not Found\n");
                return NULL;
        }
```

```
	else if(root->data > data)
		root->left = delete_tree(root->left, data);
	else if(root->data < data)
		root->right = delete_tree(root->right, data);
	else if(root->left && root->right)
	{
		root->left = find_max(root->left, &num);
		root->data = num;
	}
	else
		root = chg_node(root);
	return root;
}

void print_tree(tree* root)
{
	if(root)
	{
		printf("data = %d, ", root->data);

		if(root->left)
			printf("left = %d, ", root->left->data);
		else
			printf("left = NULL,");

		if(root->right)
			printf("right = %d\n", root->right->data);
		else
			printf("right = NULL\n");

		print_tree(root->left);
		print_tree(root->right);
	}
}
```

```
void tree_ins(tree** root, int data)
{
	if(*root == NULL)
	{
		*root = get_node();
		(*root)->data = data;
		return ;
	}
	else if((*root)->data >data)
		tree_ins(&(*root)->left, data);
	else if((*root)->data <data)
		tree_ins(&(*root)->right, data);

}

int main(void)
{
	printf("main----------------------------------\n");
	int i;
	int data[14] = {50, 45, 73, 32, 48, 46, 16, 37,
			120, 47, 130, 127,124};

	tree* root = NULL;
	printf("root do = %p",&root);
	printf("data do =  %p\n",data);
	for(i = 0; data[i]; i++)
	{
		tree_ins(&root, data[i]);
	}
	print_tree(root);

	delete_tree(root, 50);
	printf("After Delete\n");
```

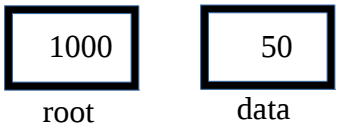
```
    print_tree(root);

    return 0;
}
```

Main

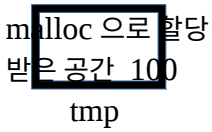


For

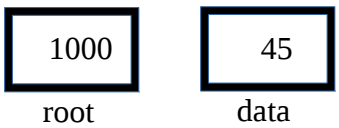
Tree_ins



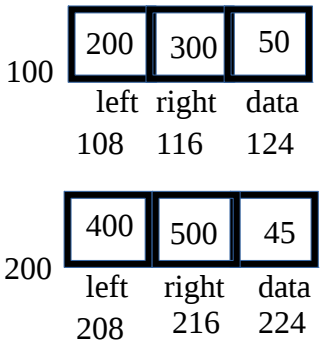
Get_node



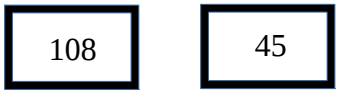
Tree_ins



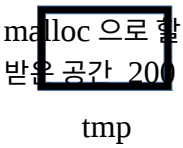
Heap



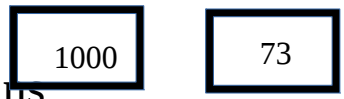
Tree_ins



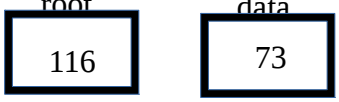
Get_node



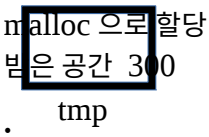
Tree_ins



Tree_ins



Get_node

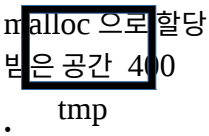


Tree_ins

Tree_ins

Tree_ins

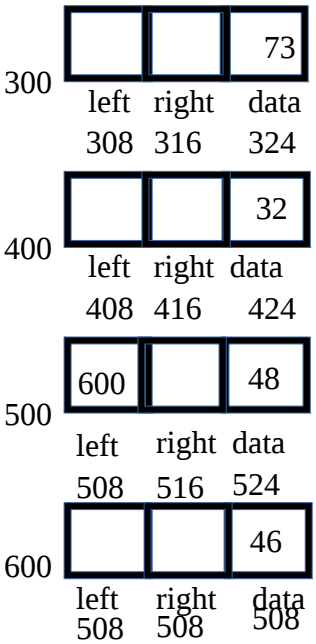
Get_node



Tree_ins

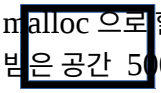
Tree_ins

Tree_ins



Get_node

malloc 으로 할당
빔은 공간 500

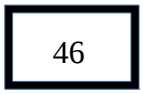


tmp

Tree_ins

1000

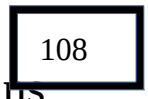
root



46

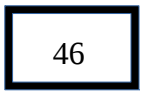
data

Tree_ins



108

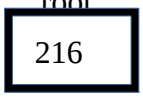
root



46

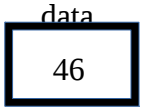
data

Tree_ins



216

root



46

data

Tree_ins

308

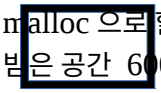
root

46

data

Get_node

malloc 으로 할당
빔은 공간 600



tmp

오늘내로 완성시키겠습니다. 죄송합니다.