

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

강사 – Innova Lee (이상훈)

gcccompil3r@gmail.com

학생-김민주

alswnqodrl@naver.com

01



10일차 내용 복습 (tree)

```
Terminal
alswnqodrl@alswnqodrl-Z20NH-AS51B1U: ~/my_proj/Hom

#include <stdio.h>
#include <malloc.h>
#include <stdlib.h>
#include <time.h>
#define EMPTY 0

typedef struct __tree
{
    int data;
    struct __tree *left;
    struct __tree *right;
}tree;

tree *get_node()
{
    tree *tmp;
    tmp=(tree*)malloc(sizeof(tree));
    tmp->left=EMPTY;
    tmp->right=EMPTY;
    return tmp;
}

tree *chg_node(tree *root)
{
    if(!root->left)
        return root->right;
    if(!root->right)
        return root->left;
}
```

```
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)
    {

```

01



10일차 내용 복습 (tree)

```
Terminal
alswnqodrl@alswnqodrl-Z20NH-AS51B1U: ~/my_proj/Homework/min
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 insert_tree(tree **root, int data)
{
    if(*root ==NULL)
    {
        *root = get_node();
        (*root)->data = data;
        return;
    }
    if((*root) -> data > data)
    {
        insert_tree(&(*root) -> left, data);
    }
    else
    {
        insert_tree(&(*root) -> right, data);
    }
}

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);
    }
}
```

102,1-8 63%

01



10일차 내용 복습 (tree)

```
Terminal
alswnqodrl@alswnqodrl-Z20NH-ASS1B1U: ~/my_proj/Homework/minjukim
return root;
}

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

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");
    }
}
```

```

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

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

int main(void)
{
    tree *root=EMPTY;
    int i;
    int data[14]={50, 45, 73, 32, 48, 46, 16, 37, 120, 47, 130, 127, 124};

    for(i=0; i < data[i]; i++)
    {
        insert_tree(&root, data[i]);
    }


    print_tree(root);
    delete_tree(root, 50);
    print_tree(root);

    return 0;
}

127,1 Bot
```



10일차 내용 복습 (tree)

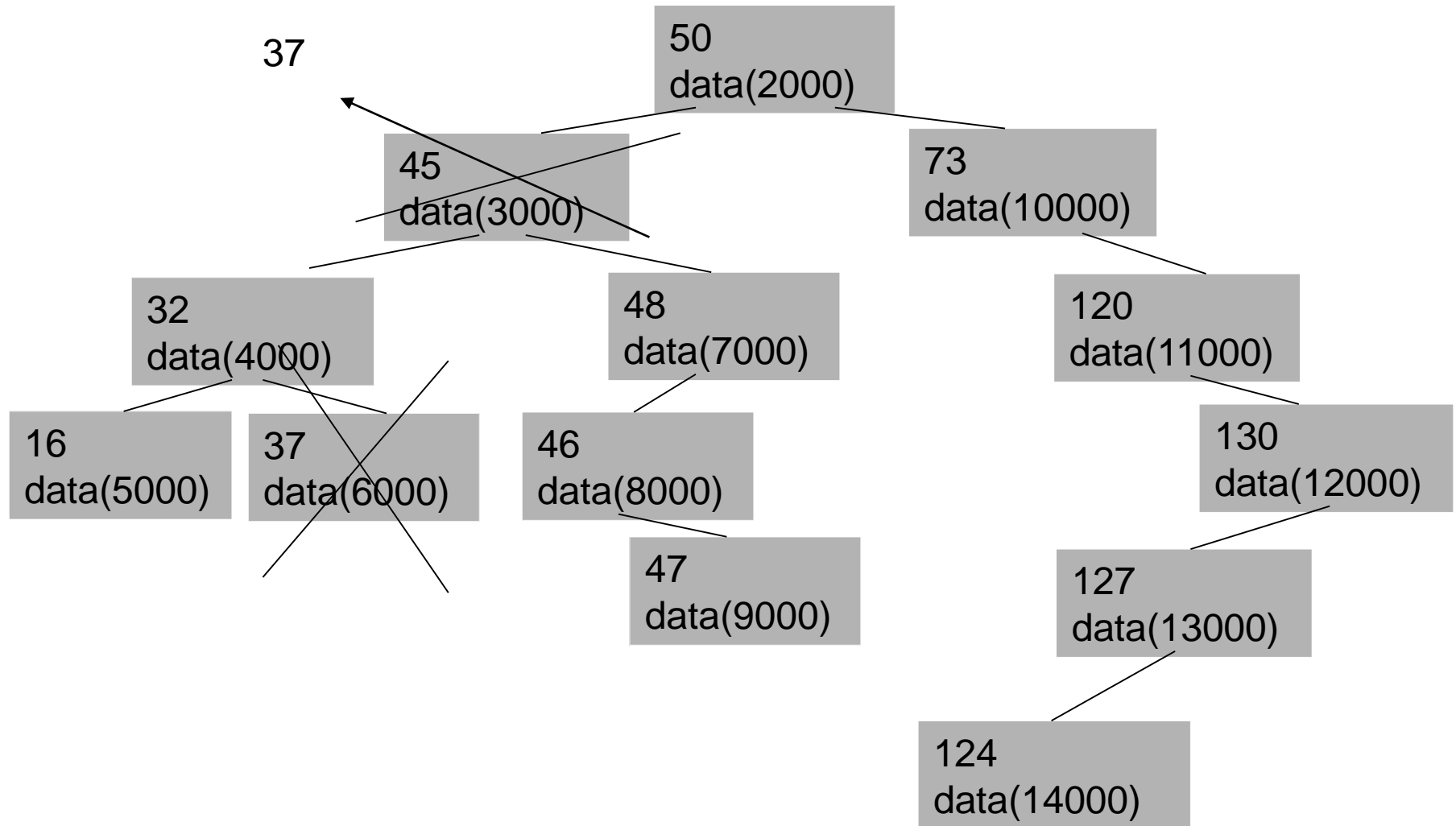


```
alswnqodrl@alswnqodrl-Z20NH-AS51B1U:~/my_proj/Homework/minjukim$ vi tree.c
alswnqodrl@alswnqodrl-Z20NH-AS51B1U:~/my_proj/Homework/minjukim$ gcc tree.c
alswnqodrl@alswnqodrl-Z20NH-AS51B1U:~/my_proj/Homework/minjukim$ ./a.out
data=50left=45right=73
data=45left=32right=48
data=32left=16right=37
data=16left=NULLright=NULL
data=37left=NULLright=NULL
data=48left=46right=NULL
data=46left=NULLright=47
data=47left=NULLright=NULL
data=73left=NULLright=120
data=120left=NULLright=130
data=130left=127right=NULL
data=127left=124right=NULL
data=124left=NULLright=NULL
data=48left=45right=73
data=45left=32right=46
data=32left=16right=37
data=16left=NULLright=NULL
data=37left=NULLright=NULL
data=46left=NULLright=47
data=47left=NULLright=NULL
data=73left=NULLright=120
data=120left=NULLright=130
data=130left=127right=NULL
data=127left=124right=NULL
data=124left=NULLright=NULL
alswnqodrl@alswnqodrl-Z20NH-AS51B1U:~/my_proj/Homework/minjukim$ vi tree.c
alswnqodrl@alswnqodrl-Z20NH-AS51B1U:~/my_proj/Homework/minjukim$
```

Main 2000 Root (1000)		
delete_tree 2000 root	45 data	Num
delete_tree 3000 root	45 data	37 Num (100)
find_max 4000 root	100 data	
0 ← find_max 6000 root	100 data	
chg_node 0 root		

02

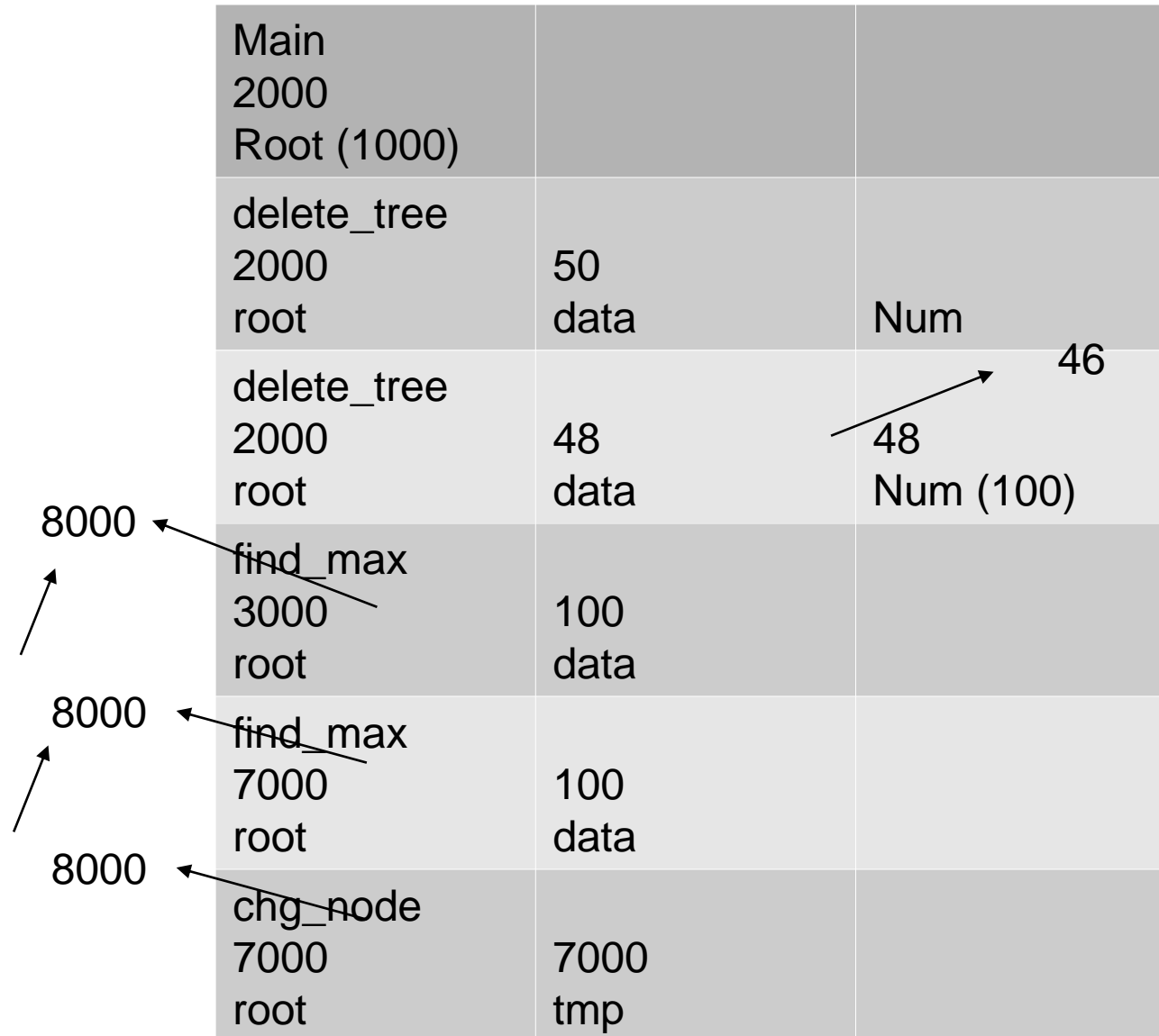
10일차 내용 복습 – 그림그리기 (tree 45 delete)



03



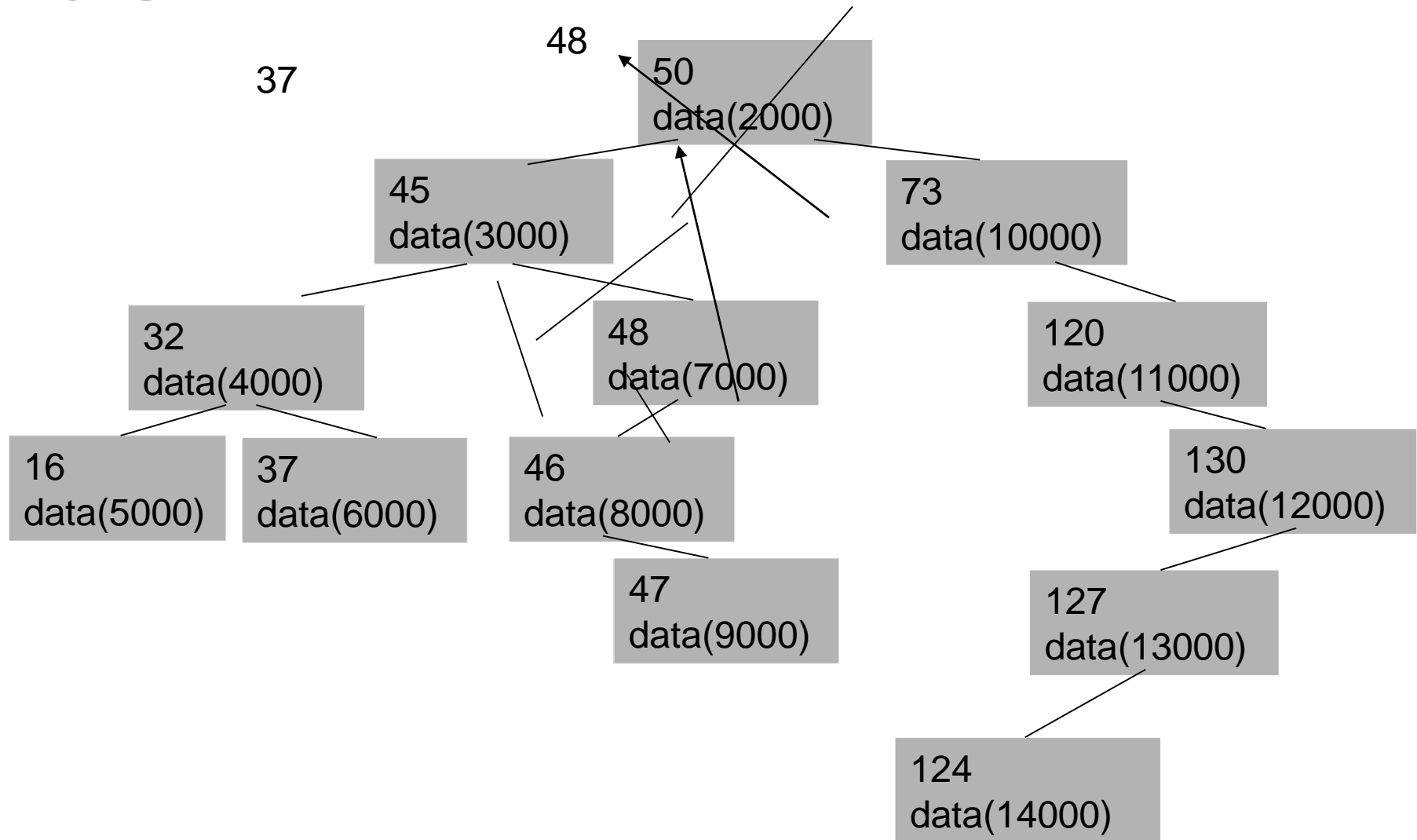
10일차 내용 복습 – 그림그리기 (tree 50 delete)



03



10일차 내용 복습 – 그림그리기 (tree 50 delete)



10일차 내용 복습 – 그림그리기 (tree 127 delete)

14000	delete_tree 13000 root	127 data
13000	delete_tree 12000 root	127 data
14000	chg_node 13000 root	13000 tmp

04

10일차 내용 복습 – 그림그리기 (tree 127 delete)

