

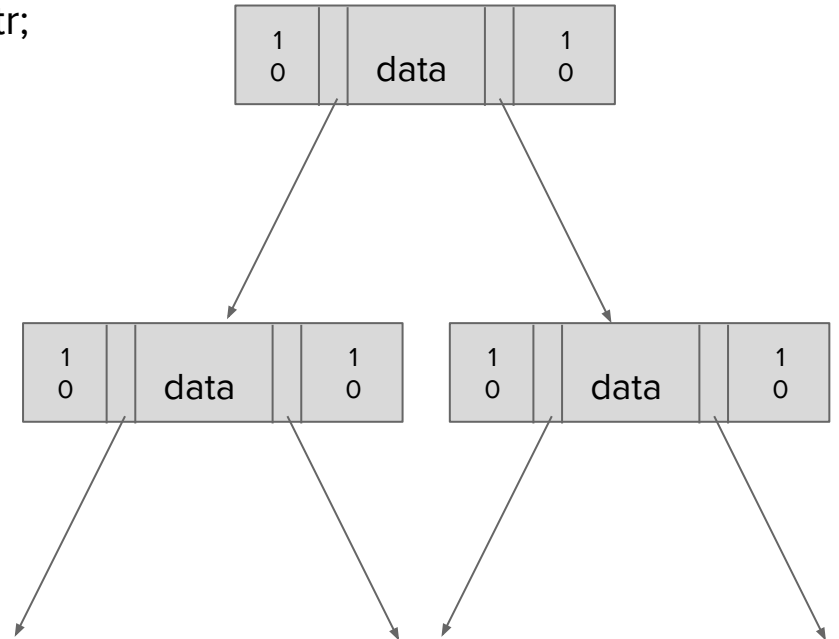
Lab 4 Threaded Tree

2019. 03. 28

lab 4. Threaded Tree

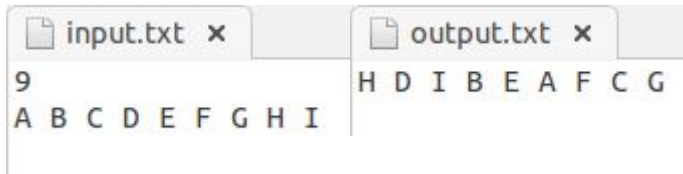
- Data Structure Description

```
typedef struct threaded_tree *threaded_ptr;  
typedef struct threaded_tree{  
    short int left_thread;  
    threaded_ptr left_child;  
    char data;  
    threaded_ptr right_child;  
    short int right_thread;  
};
```



lab 4. Threaded Tree

- **Implement inorder traversal only with Threaded Tree using linked list(not array).**
 - maximum number of node is 100.
 - first line of input.txt is number of nodes.
 - second line is data of nodes (separated with space).
 - make complete binary tree
 - define function `InsertNode(threaded_tree node, threaded_tree tree)` .
 - put the following nodes in input.txt .
 - print inorder traversal.
 - define false = 0, true = 1 (short (int) type).
- **You have to use file I/O like the previous assignment.**

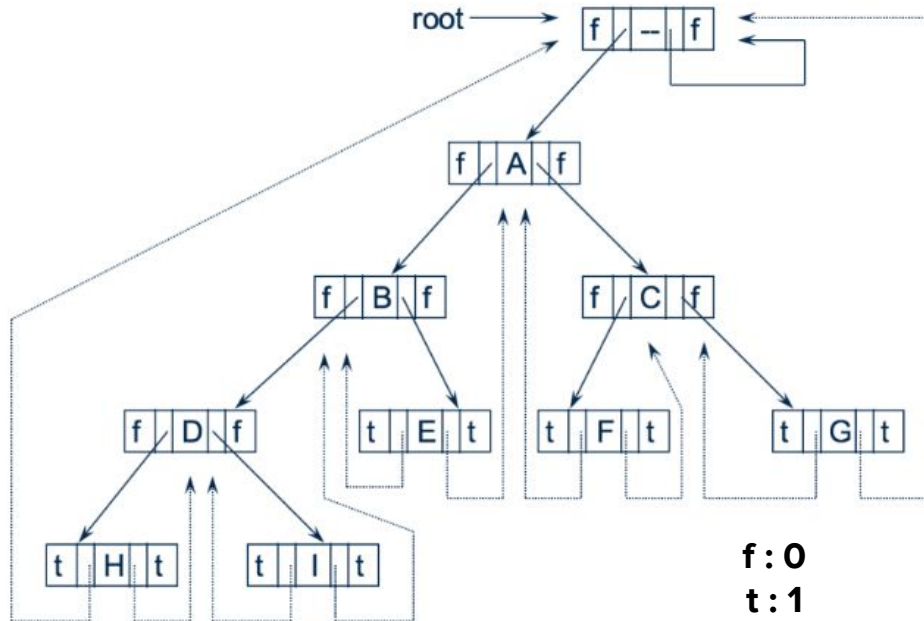


The image shows two side-by-side text editor windows. The left window, titled 'input.txt', contains the number '9' on the first line and the letters 'A B C D E F G H I' on the second line. The right window, titled 'output.txt', contains the letters 'H D I B E A F C G' on a single line.

```
input.txt x  output.txt x
9            H D I B E A F C G
A B C D E F G H I
```

lab 4. Threaded Tree

The tree you create should follow the structure below.



lab 4. Threaded Tree

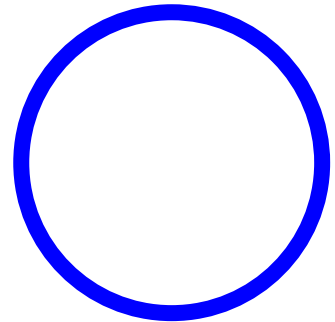
■ iterative in-order traversal using **stack**

```
void iterInorder (Tree node) {  
    int top = -1;  
    Tree stack[MAX_SIZE];  
    for (;;) {  
        for (; node; node = node->leftChild)  
            push(node);  
  
        node = pop(); // pop parent  
        if (!node) break;  
        printf("%d", node->data);  
        node = node->rightChild;  
    }  
}
```

■ inorder traversal **recursive**

```
void inorder(Tree ptr) {  
    if(ptr) {  
        inorder(ptr->left_child);  
        printf("%d", ptr->data);  
        inorder(ptr->right_child);  
    }  
}
```

```
void tinorder(threaded_ptr tree) {  
    threaded_ptr temp = tree;  
    for (;;) {  
        temp = insucc(temp);  
        if (temp == tree) break;  
        printf("%3c", temp->data);  
    }  
}
```

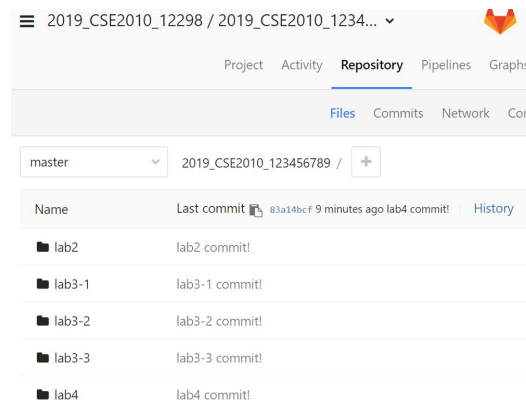


lab 4. Threaded Tree

- Submission

- Project directory name : lab4
- Source file name : p4.c
- Executable file name : p4.out
- You should upload in the hconnect (Gitlab) server.

```
daewook@daewook-VirtualBox:~/2019_CSE2010_123456789/lab4$ gcc p4.c -o p4.out
daewook@daewook-VirtualBox:~/2019_CSE2010_123456789/lab4$ ./p4.out
daewook@daewook-VirtualBox:~/2019_CSE2010_123456789/lab4$ cat output.txt
H D I B E A F C G
daewook@daewook-VirtualBox:~/2019_CSE2010_123456789/lab4$ git add .
daewook@daewook-VirtualBox:~/2019_CSE2010_123456789/lab4$ git commit -m "lab4 commit!"
[master 83a14bc] lab4 commit!
```



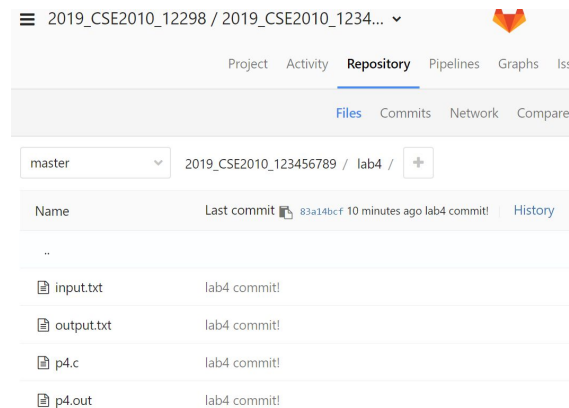
2019_CSE2010_12298 / 2019_CSE2010_123456789

Project Activity **Repository** Pipelines Graphs

Files Commits Network Compare

master 2019_CSE2010_123456789 / +

Name	Last commit	History
lab2	lab2 commit!	
lab3-1	lab3-1 commit!	
lab3-2	lab3-2 commit!	
lab3-3	lab3-3 commit!	
lab4	lab4 commit!	



2019_CSE2010_12298 / 2019_CSE2010_123456789

Project Activity **Repository** Pipelines Graphs Issues

Files Commits Network Compare

master 2019_CSE2010_123456789 / lab4 / +

Name	Last commit	History
..		
input.txt	lab4 commit!	
output.txt	lab4 commit!	
p4.c	lab4 commit!	
p4.out	lab4 commit!	

DeadLine

Wednesday, 03 April, 23 : 59 pm