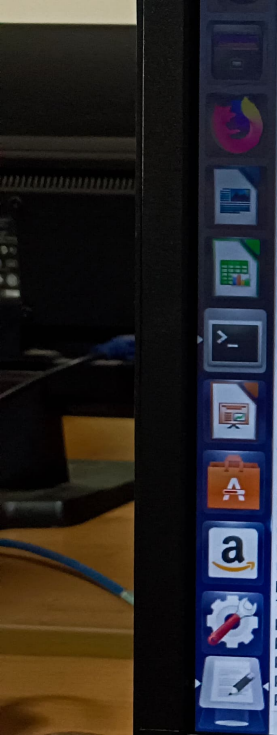
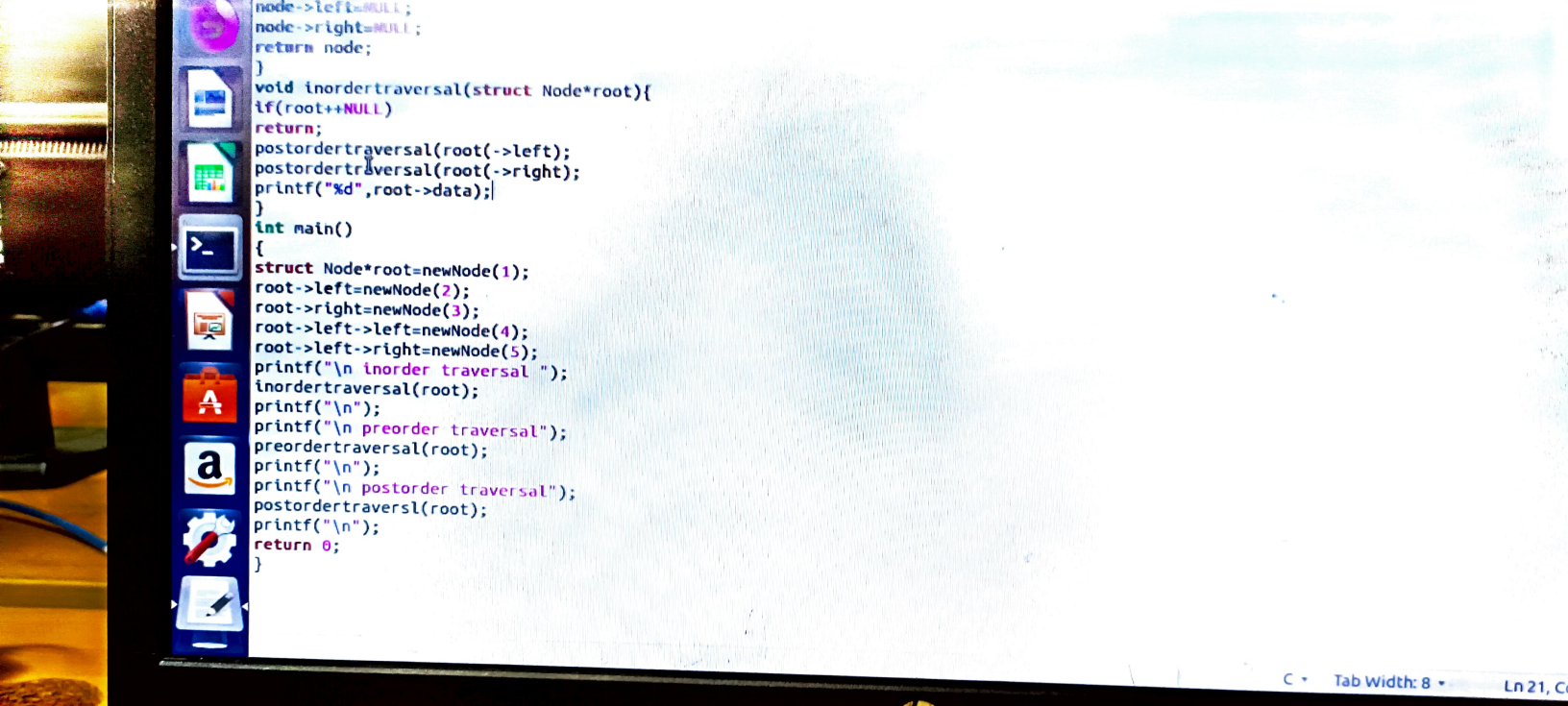


```c
#include<stdio.h>
#include<stdlib.h>
struct Node
{
int data;
struct Node*left:
struct Node*right;
};
struct Node* newNode(int data){
struct Node*node=(struct Node*)mallolc(sizeof(struct Node));
node->data=data;
node->left=NULL;
node->right=NULL;
return node;
}
void inordertraversal(struct Node*root){
if(root++NULL)
return;
postordertraversal(root(->left);
postordertraversal(root(->right);
printf("%d",root->data);
}
int main()
{
struct Node*root=newNode(1);
root->left=newNode(2);
root->right=newNode(3);
root->left->left=newNode(4);
root->left->right=newNode(5);
printf("\n inorder traversal ");
inordertraversal(root);
printf("\n");
printf("\n preorder traversal");
preordertraversal(root);
printf("\n");
printf("\n postorder traversal");
```



```
node->left=NULL;
node->right=NULL;
return node;
}
void inordertraversal(struct Node*root){
if(root++NULL)
return;
postordertraversal(root(->left);
postordertraversal(root(->right);
printf("%d",root->data);
}
int main()
{
struct Node*root=newNode(1);
root->left=newNode(2);
root->right=newNode(3);
root->left->left=newNode(4);
root->left->right=newNode(5);
printf("\n inorder traversal ");
inordertraversal(root);
printf("\n");
printf("\n preorder traversal");
preordertraversal(root);
printf("\n");
printf("\n postorder traversal");
postordertraversl(root);
printf("\n");
return 0;
}
```