**Отчет по лабораторной работе №23** по курсу\_1\_\_\_\_\_

|  |  |
| --- | --- |
|  | Студент группы М80-105Б-21 Жилов Андрей, № по списку 3  Контакты www, e-mail: [klzxrcn3692@outlook.com](mailto:klzxrcn3692@outlook.com)\_\_\_\_\_\_\_\_\_\_\_  Работа выполнена: «12» мая 2022г.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Преподаватель: В. К. Титов каф. 806\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Входной контроль знаний с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Отчет сдан «12» мая 2022 г., итоговая оценка \_\_\_\_\_\_\_\_\_  Подпись преподавателя \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

1. **Тема:** Динамические структуры данных. Обработка деревьев.
2. **Цель работы:** Составить программу на языке Си для построения и обработки дерева.
3. **Задание:** (*Вариант №13*):

Проверить является ли дерево линейным списком вершин.

1. **Оборудование(лабораторное)**:

ЭВМ -, процессор -, имя узла сети - с ОП - ГБ,

НМД - ГБ, терминал- адрес -, принтер –

Другие устройства –

*Оборудование ПЭВМ студента, если использовалось:*

Процессор AMD Ryzen 3500U, с ОП 8 ГБ, НМД 256 ГБ. Монитор 1920x1080 пикс.

Другие устройства –

1. **Программное обеспечение:**

Операционная система семейства -, наименование - версия –

интерпретатор команд – версия

Система программирования - версия –

Редактор текстов - версия –

Утилиты операционной системы –

Прикладные системы и программы –

Местонахождение и имена файлов программ и данных -

*Программное обеспечение ЭВМ студента, если использовалось:*

Операционная система семейства GNU/Linux, наименование Kali версия 5.10.0

интерпретатор команд Bash версия 5.1.8(1)-release

Система программирования \_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_версия\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

редактор текстов emacs версия 27.1

Утилиты операционной системы cmp, diff, tail, od, vi, vim, ed, head, du, xargs, indent, comm, grep, tee, sum, mc, awk, iconv, file,df,split, wc, join, tr,cut,tar,sed,patch,touch, paste, mktemp, dd,sort, uniq, nroff, gzip, bzip2, md5, find, cpp

Прикладные системы и программы –

Местонахождение и имена файлов программ и данных -

1. **Идея, метод, алгоритм** решения задачи (в формах: словесной, псевдокода, графической [блок-схема, диаграмма, рисунок,

При запуске программы вызывается меню из 7 функций.

0: Остановка работы программы

1: Генерируется дерево из случайных символов, количество которых вводит пользователь, после вызова функции.

2: На экран выводится дерево

3: Добавляет элемент в дерево.

4: Удаляет элемент, который ввел пользователь из дерева. Для этого проходимся по дереву, пока не найдем нужный элемент. Если у узла нет дочерних узлов, то у его родителя нужно просто заменить указатель на 0. Если у узла есть только один дочерний узел, то нужно создать новую связь между родителем удаляемого узла и его дочерним узлом. Если у узла два дочерних узла, то нужно найти следующий за ним элемент, его правого потомка подвесить на место найденного элемента, а удаляемый узел заменить найденным узлом.

5: Удаляет дерево.

6: Выводит является ли дерево линейным списком, для этого проходимся по дереву и проверяем, чтобы указатели на brother были равны 0.

7:Выводит меню на экран

1. **Сценарий выполнения работы** [план работы, первоначальный текст программы в черновике (можно на отдельном листе) и тесты либо соображения по тестированию].

lr23.c:

#include <stdio.h>

#include<stdlib.h>

#include<time.h>

typedef char tdata;

int i,r,q;

tdata m;

struct node ;

typedef node \* link;

struct node

{

tdata data;

link son;

link brother;

};

void printtree(link t)

{

static int l=0;

if(t)

{

l++;

printtree(t->brother);

for(i=0;i<l;i++)

printf(" ");

printf("\\\_\_%c\n",t->data);

printtree(t->son);

l--;

}

}

link search(link tree, tdata c)

{

if(tree)

{

if(tree->data==c)

return tree;

link t=search(tree->son,c);

if(!t)

t=search(tree->brother,c);

return t;

}

return 0;

}

link searchfather(link tree, tdata c)

{

link t;

if(tree)

{

if(tree->son&&tree->son->data==c)

return tree;

t=tree->son;

while(t)

{

if(t->brother&&t->brother->data==c)

return tree;

t=t->brother;

}

t=0;

t=searchfather(tree->son,c);

if(!t)

t=searchfather(tree->brother,c);

return t;

}

return 0;

}

link searchbrother(link tree, tdata c)

{

if(tree)

{

if(tree->brother&&tree->brother->data==c)

return tree;

link t=searchbrother(tree->brother,c);

if(!t)

t=searchbrother(tree->son,c);

return t;

}

return 0;

}

void insertbrother(link &t,tdata v)

{

if(!t)

{

t=new node;

t->data=v;

t->son=0;

t->brother=0;

}

else

insertbrother(t->brother,v);

}

void insertson(link &t,tdata v)

{

if(!(t->son))

{

t->son=new node ;

t->son->data=v;

t->son->son=0;

t->son->brother=0;

}

else

insertbrother(t->son,v);

}

void inserttree(link &t,tdata v)

{

if(!t)

{

t=new node;

t->data=v;

t->son=0; t->brother=0;

}

else

{

r=rand()%2;

if(r)

inserttree(t->son,v);

else

inserttree(t->brother,v);

}

}

void action(link tree)

{

link t=tree;

while(t)

{

if(t->brother!=0)

{

printf("Tree is not a linel list\n");

return;

}

t=t->son;

}

printf("Tree is a linel list\n");

}

void mix(tdata \*m, int n)

{

int r; tdata s;

for(i=0;i<n;i++)

{

r=rand()%n;

s=m[i];

m[i]=m[r];

m[r]=s;

}

}

int main()

{

long a;

srand(time(&a));

int k=7,n;

tdata \*m, f, s;

link tree=0,t;

while(k)

{

if(k==7)

{

printf("Input from 0 to 7 actions:\n"

"0 - Exit\n"

"1 - Add random tree\n"

"2 - Print tree\n"

"3 - Insert item\n"

"4 - Delete item\n"

"5 - Delete tree\n"

"6 - Main action\n"

"7 - Print menu\n"

);

}

else if(k==1)

{

printf("please enter a numper of nodes\n");

scanf("%d",&n);

m = new tdata[n];

for(int i=0;i<n;++i)

{

m[i]=100+i;

}

mix(m,n);

for(int i = 0; i<n;++i)

{

inserttree(tree,m[i]);

}

}

else if(k==2)

{

if(!tree)

printf("Tree is empty\n");

else

printtree(tree);

}

else if(k==3)

{

printf("\nInput value of root ==>");

scanf(" %c",&f);

t=new node ;

t->data=f;

t->son=0;

t->brother=0;

if(!tree)

tree=t;

else

{

t->son=tree;

tree=t;

}

}

else if(k==4)

{

printf("\nInput value of node ==>");

scanf(" %c",&f);

if(tree->data==f)

if(tree->brother==0)

{

tree=0;

continue;

}

else

tree=tree->brother;

else

{

t=searchbrother(tree,f);

if(t)

t->brother=t->brother->brother;

else

{

t=searchfather(tree,f);

if(t)

if(t->son->brother)

t->son=t->son->brother;

else t->son=0;

else printf("\nNode not found\n");

}

}

}

else if(k==5)

{

tree=0;

}

else if(k==6)

{

action(tree);

}

printf("Input from 0 to 7 actions:\n");

scanf("%d",&k);

}

return 0;

}

*Пункты 1-7 отчета составляются сторого до начала лабораторной работы.*

*Допущен к выполнению работы.*  **Подпись преподавателя \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

1. **Распечатка протокола**  (подклеить листинг окончательного варианта программы с тестовыми примерами, подписанный преподавателем).

lain@navi-1:/mnt/c/Users/lain/Desktop/23$ cat head\_lr23.txt

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

| |

| Лабораторная работа №23 |

| Динамические струкуры данных |

| Обработка деревьев |

| Студента группы М8О-105Б-21 |

| Жилова Андрея |

| |

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

lain@navi-1:/mnt/c/Users/lain/Desktop/23$ g++ lr23.c

lain@navi-1:/mnt/c/Users/lain/Desktop/23$ ./a.out

Input from 0 to 7 actions:

0 - Exit

1 - Add random tree

2 - Print tree

3 - Insert item

4 - Delete item

5 - Delete tree

6 - Main action

7 - Print menu

Input from 0 to 7 actions:

1

please enter a numper of nodes

21

Input from 0 to 7 actions:

2

\\_\_w

\\_\_u

\\_\_g

\\_\_k

\\_\_s

\\_\_q

\\_\_e

\\_\_f

\\_\_x

\\_\_r

\\_\_o

\\_\_j

\\_\_t

\\_\_p

\\_\_n

\\_\_m

\\_\_v

\\_\_i

\\_\_d

\\_\_l

\\_\_h

Input from 0 to 7 actions:

7

Input from 0 to 7 actions:

0 - Exit

1 - Add random tree

2 - Print tree

3 - Insert item

4 - Delete item

5 - Delete tree

6 - Main action

7 - Print menu

Input from 0 to 7 actions:

3

Input value of root ==>p

Input from 0 to 7 actions:

2

\\_\_p

\\_\_w

\\_\_u

\\_\_g

\\_\_k

\\_\_s

\\_\_q

\\_\_e

\\_\_f

\\_\_x

\\_\_r

\\_\_o

\\_\_j

\\_\_t

\\_\_p

\\_\_n

\\_\_m

\\_\_v

\\_\_i

\\_\_d

\\_\_l

\\_\_h

Input from 0 to 7 actions:

4

Input value of node ==>l

Input from 0 to 7 actions:

2

\\_\_p

\\_\_w

\\_\_u

\\_\_g

\\_\_k

\\_\_s

\\_\_q

\\_\_e

\\_\_f

\\_\_x

\\_\_r

\\_\_o

\\_\_j

\\_\_t

\\_\_p

\\_\_n

\\_\_m

\\_\_v

\\_\_i

\\_\_d

\\_\_h

Input from 0 to 7 actions:

7

Input from 0 to 7 actions:

0 - Exit

1 - Add random tree

2 - Print tree

3 - Insert item

4 - Delete item

5 - Delete tree

6 - Main action

7 - Print menu

Input from 0 to 7 actions:

6

Tree is not a linel list

Input from 0 to 7 actions:

5

Input from 0 to 7 actions:

2

Tree is empty

Input from 0 to 7 actions:

1

please enter a numper of nodes

2

Input from 0 to 7 actions:

2

\\_\_e

\\_\_d

Input from 0 to 7 actions:

6

Tree is a linel list

Input from 0 to 7 actions:

0

1. **Дневник отладки** должен содержать дату и время сеансов отладки и основные события (ошибки в сценарии и программе, нестандартные ситуации) и краткие комментарии к ним. В дневнике отладки приводятся сведения об использовании других ЭВМ, существенном участии преподавателя и других лиц в написании и отладке программы.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № | Лаб. или дом. | Дата | Время | Событие | Действие по исправлению | Примечание |
|  |  |  |  |  |  |  |

1. **Замечания автора по существу работы:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

1. **Выводы**:

В результате выполнения данной лабораторной работы я научился работать с деревьями на языке Си

Недочеты при выполнении задания могут быть устранены следующим образом:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Подпись студента \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_