Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет» Кафедра ИИТ

Лабораторная работа №5

по дисциплине «Операционные системы и системное программирование»

Тема: «ВВОД/ВЫВОД»

Вариант 7

Выполнил: студент 2-го курса группы ПО-6 Лавренчик Д.О.

Проверил: Давидюк Ю.И.

Брест 2022

**Лабораторная работа №5**

**Ход работы**

***Задание для выполнения:***

Написать программу, которая получает со стандартного потока ввода содержимое любого текстового файла и выводит его в стандартный поток вывода, поменяв местами буквы в середине слов (первая и последняя буквы слов остаются на своих местах). Протестировать на различных файлах с использованием конвейеров в различных комбинациях вашей программы и команд cat, sort, head, tail.

**Тестовые данные:**



**Код программы:**

#include <iostream>

#include <vector>

#include <string> #include

<unistd.h>

using namespace std;

vector<string> read\_STDIN\_by\_words();//получаем слова в вектор

vector<string> reverse\_words(vector<string> words); string reverse\_word(string word); void stdout\_print(vector<string> words);//выввод

int main()

{

vector<string> stdin\_words = read\_STDIN\_by\_words();

vector<string> reversed\_words = reverse\_words(stdin\_words);

stdout\_print(reversed\_words); return 0;

} vector<string> read\_STDIN\_by\_words()

{

vector<string> words;

string current\_word = "";

char buf[1];

while (read(STDIN\_FILENO, buf, 1) > 0)

{

if (\*buf == ' ')

{

words.push\_back(current\_word); current\_word = "";

continue; } if (\*buf == '\n') {

words.push\_back(current\_word); words.push\_back("\n"); current\_word

= ""; continue;

}

current\_word += \*buf;

}

return words;

} vector<string> reverse\_words(vector<string> words)

{

vector<string> reversed\_words;

for (auto&& word : words)

{

reversed\_words.push\_back(reverse\_word(word));

}

return reversed\_words; } string reverse\_word(string word)

{ if (word.length() <=

2) { return word;

}

string reversed\_word = ""; char first = word[0]; char last = word[word.length() - 1];

for (size\_t i = word.length() - 2; i > 0; i--)

{

reversed\_word += word[i];

}

reversed\_word = first + reversed\_word + last;

return reversed\_word;

} void stdout\_print(vector<string> words) { for (auto&& word : words)

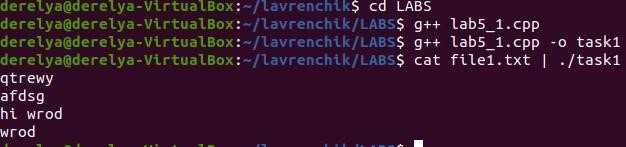
{

word += word[0] == '\n' ? '\0' : ' ';

write(STDOUT\_FILENO, &word[0], word.length());

} }

**Результат выполнения программы:**



**Задание 2**

В текущем каталоге создайте файл, в котором сформируйте словарь слов с переставленными буквами, в виде: слово – совло, словарь – свлораь, **Код программы:**

#include <iostream>

#include <vector>

#include <string>

#include <unistd.h>

#include <fcntl.h>

using namespace std;

void create\_dict();

string reverse\_word(string word); int

main()

{

create\_dict();

return 0;

}

void create\_dict()

{

int file = open("map", O\_WRONLY | O\_CREAT | O\_APPEND);

string current\_word = ""; char buf[1];

while (read(STDIN\_FILENO, buf, 1) > 0)

{

if (\*buf == ' ' || \*buf == '\n')

{

string line = current\_word + " - " + reverse\_word(current\_word)

+ "\n";

write(file, &line[0], line.length());

current\_word = ""; continue;

}

current\_word += \*buf;

}

close(file);

}

string reverse\_word(string word) {

if (word.length() <= 2)

{

return word;

}

string reversed\_word = ""; char first =

word[0]; char last = word[word.length() - 1];

for (size\_t i = word.length() - 2; i > 0; i--)

{

reversed\_word += word[i];

}

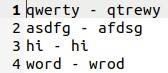
reversed\_word = first + reversed\_word + last;

return reversed\_word;

}



Полученный файл:



**Вариант 2**

Задание: Написать программу, которая получает со стандартного потока ввода

содержимое каталога /etc и выводит в стандартный поток вывода информацию только о файлах,

владельцем которых является система (root). Протестировать на различных файлах с

использованием конвейеров в различных комбинациях вашей программы и команд ls, sort, head,

tail.

Вариант 2. Создайте отдельные текстовые файлы для каждого пользователя, и поместите

туда имена всех созданных им файлов.

Код программы:

#include <stdio.h>

#include <sys/stat.h>

#include <fcntl.h>

#include <unistd.h>

#include <cstring>

#include <vector>

#include <string>

#include <iostream>

using namespace std;

void stdout\_print(vector<string> words) {

for (auto &&word : words)

{

write(1, &word[0], word.length());

write(1,"\n",1);

}

}

int main() {

vector<string> lines;

vector<string> result;

vector<string> result\_2;

string current\_word = "";

char buf[1];

while (read(0, buf, 1) > 0)

{

if (\*buf == '\n')

{

lines.push\_back(current\_word);

lines.push\_back("\n");

current\_word = "";

}

else{

current\_word += \*buf;

}

}

for (int i = 1; i < lines.size(); ++i) {

if (lines[i][13] == 'r' || lines[i][14] == 'r') {

result.push\_back(lines[i]);

}

}

stdout\_print(result);

int fd;

char name[] = "file\_var2.txt";

for (int i = 1; i < lines.size(); ++i) {

if (lines[i][19] == 'r') {

result\_2.push\_back(lines[i]);

}

}

for (auto &&word : result\_2)

{

(void) umask(0);

fd=open(name, O\_WRONLY | O\_CREAT | O\_APPEND, 0666);

write(fd, &word[0], word.length());

write(fd,"\n",1);

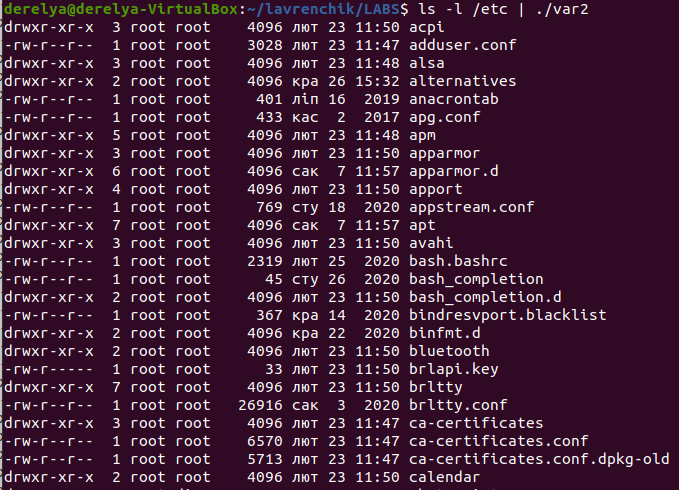
}

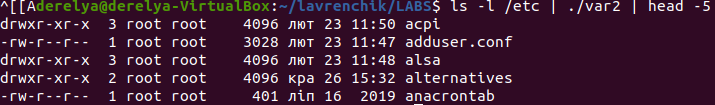
close(fd);

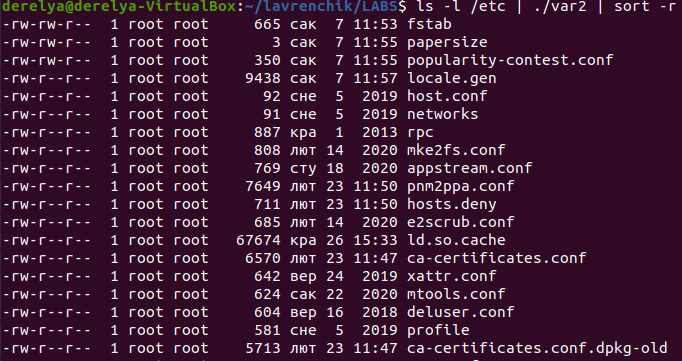
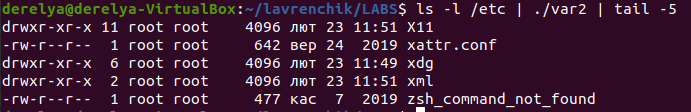
return 0;

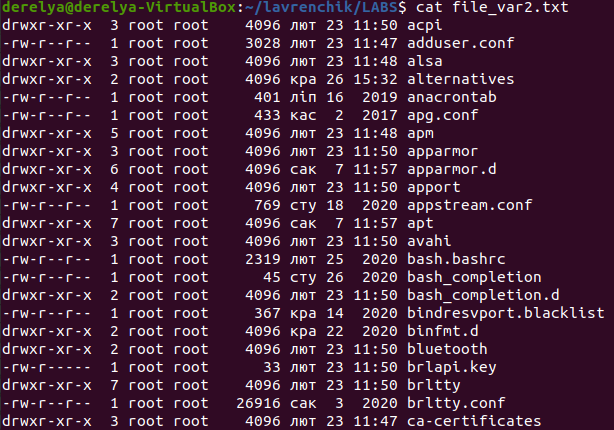
}

Результат работы:









**Вариант 4**

Написать программу, которая получает со стандартного потока ввода права доступа к файлам каталога, и выводит в стандартный поток вывода те из них, у которых установлен бит запуска владельцем. Протестировать на различных каталогах с использованием конвейеров в различных комбинациях вашей программы и команд ls, sort, head, tail.

Выведите в отдельные файлы списки запускаемых владельцем, читаемых

владельцем и записываемых владельцем файлов.

Код программы:

#include <sys/types.h>

#include <fcntl.h>

#include <string>

#include <unistd.h>

#include <stdio.h>

#include <sys/stat.h>

#include <cstring>

#include <vector>

#include <iostream>

using namespace std;

void stdout\_print(vector<string> words) {

for (auto &&word : words)

{

write(1, &word[0], word.length());

write(1,"\n",1);

}

}

void write\_file(vector<string> vec,const char name[]) {

int fd;

(void) umask(0);

fd = open(name, O\_WRONLY | O\_CREAT, 0666);

for (auto &&str : vec)

{

write(fd, &str[0], str.length());

write(fd,"\n",1);

}

close(fd);

}

int main()

{

vector<string> lines;

vector<string> result;

vector<string> resultR;

vector<string> resultW;

vector<string> resultX;

string current\_word = "";

char buf[1];

while (read(0, buf, 1) > 0)

{

if (\*buf == '\n')

{

lines.push\_back(current\_word);

lines.push\_back("\n");

current\_word = "";

}

else{

current\_word += \*buf;

}

}

for (int i = 1; i < lines.size(); ++i) {

if (lines[i][3] == 'x') {

result.push\_back(lines[i]);

}

}

stdout\_print(result);

for (int i = 0; i < lines.size(); i++)

{

if (lines[i][1]=='r')

{

resultR.push\_back(lines[i]);

}

if (lines[i][2]=='w')

{

resultW.push\_back(lines[i]);

}if (lines[i][3]=='x')

{

resultX.push\_back(lines[i]);

}

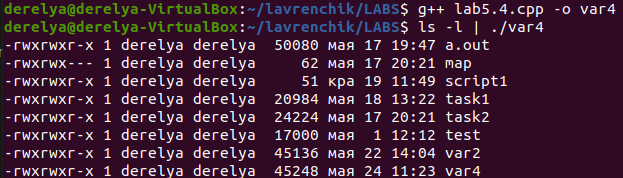
}

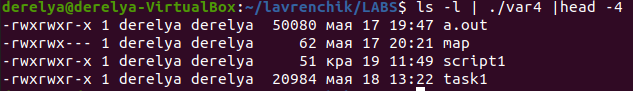
write\_file(resultR,"R.txt");

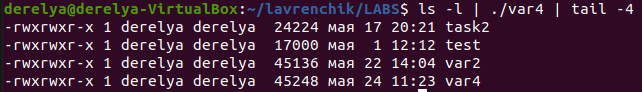
write\_file(resultW,"W.txt");

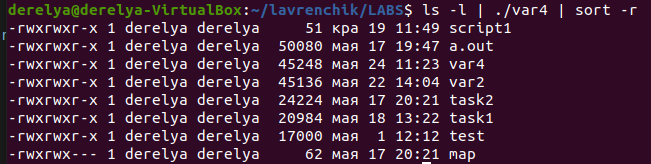
write\_file(resultX,"X.txt");

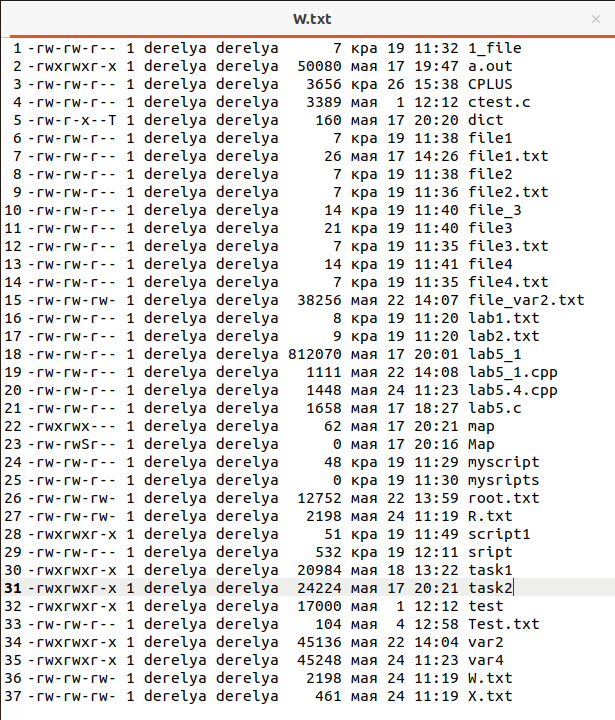
}

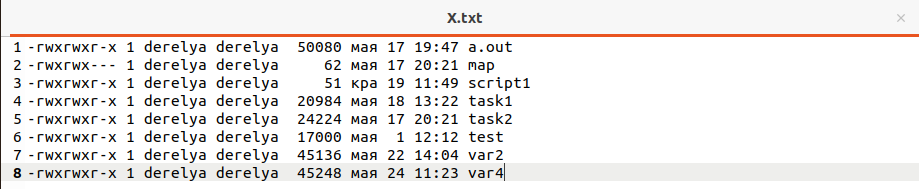


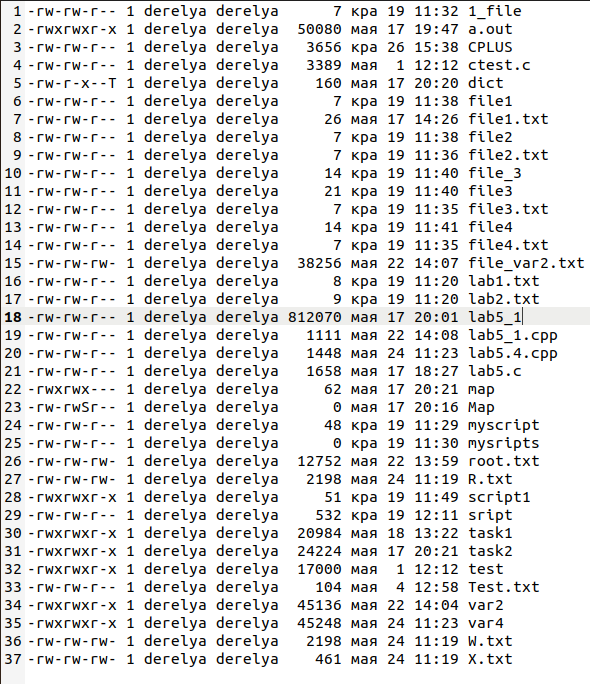












**Вывод:** в ходе выполнения данной лабораторной работы были изучены основы ввода вывода в операционной системе UNIX.