Федеральное государственное автономное

образовательное учреждение

высшего образования

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт Космических и информационных технологий

институт

Кафедра «Информатика»

кафедра

**ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ**

Лабораторная работа No 4. Управление файлами в ОС GNU/Linux

тема

Преподаватель \_\_\_\_\_\_\_\_\_ А.С. Кузнецов

подпись, дата инициалы, фамилия

Студент КИ18-17/1б 031831229 \_\_\_\_\_\_\_\_\_ В.А. Прекель

номер группы, зачетной книжки подпись, дата инициалы, фамилия

Красноярск 2019

Содержание

[Содержание 2](#_Toc24305909)

[1 Цель работы с постановкой задачи 4](#_Toc24305910)

[1.1 Цель работы 4](#_Toc24305911)

[1.2 Задача работы 4](#_Toc24305912)

[1.3 Описание и пояснение к работе 4](#_Toc24305913)

[2 Описание использованных при выполнении задания функций Linux API управления файлами 5](#_Toc24305914)

[3 Примеры использования (вызова) этих функций в представленном программном коде 8](#_Toc24305915)

[3.1 Обертки над функциями низкоуровнего ввода-вывода из VS (например, \_open()) или Linux API (например open()) (Lab\_04\_Lib/IOWrapper.c): 8](#_Toc24305916)

[3.2 Использование lseek() для перемещение позиции в файле на место указанной записи (Lab\_04\_Lib/RecordFile.c): 9](#_Toc24305917)

[3.1 Использование write() для записи записи в файл с записями (Lab\_04\_Lib/RecordFile.c): 10](#_Toc24305918)

[3.2 Использование close() для закрытия файла с записями (Lab\_04\_Lib/RecordFile.c): 10](#_Toc24305919)

[3.1 Использование read() для чтения записи из файла с записями (Lab\_04\_Lib/RecordFile.c): 10](#_Toc24305920)

[3.2 Использование ftruncate() для отсечения записей в файле (Lab\_04\_Lib/RecordFile.c): 10](#_Toc24305921)

[3.3 Использование access() для проверки файла на существование, читаемость и записываемость (Lab\_04\_Lib/RecordFile.c): 11](#_Toc24305922)

[4 Содержимое файла makefile. 11](#_Toc24305923)

[4.1 ./Makefile 11](#_Toc24305924)

[4.2 ./Lab\_04\_Lib/Makefile 11](#_Toc24305925)

[4.1 ./Lab\_04\_Console/Makefile 12](#_Toc24305926)

[5 Примеры работы программ в виде перехватов содержимого экрана 12](#_Toc24305927)

[5.1 Снимок экрана запуска № 1 (WSL Ubuntu 18.04, make, GCC) 12](#_Toc24305928)

[5.2 Снимок экрана запуска № 2 (Windows 10, mingw32-make, MSVC) 13](#_Toc24305929)

[5.3 Запуск №3 (WSL Ubuntu 18.04, make, Clang) 14](#_Toc24305930)

[5.4 Шестнадцатеричный дамп после запуска №4 17](#_Toc24305931)

# Цель работы с постановкой задачи

## Цель работы

Программная реализация обработки текстовой информации, хранящейся во внешней памяти, с использованием системных вызовов низкоуровневого ввода-вывода.

## Задача работы

Требуется разработать программу, позволяющую считывать, модифицировать существующие и добавлять новые структуры (записи) фиксированной длины из/во входные и выходные файлы. При реализации должны использоваться только средства низкоуровневого ввода-вывода. Все операции выполняются только над содержимым файловых объектов, а не над содержимым информационных структур, хранящихся во внутренней памяти. Использование высокоуровневых средств является ошибочным. Аналогично, выполнение операций во внутренней памяти с сохранением и загрузкой результатов в файлы является ошибочным.

**Вариант 17.** Структура данных: архипелаг; количество островов; количество обитаемых островов. Создать два запроса, позволяющих определить, имеются ли архипелаги, состоящие только из необитаемых островов, и получить список архипелагов с указанием количества островов в них.

## Описание и пояснение к работе

Используется система сборки GNU Make. Используется интерфейс командной строки. Для сборки под Linux и GCC и вывода справки требуется:

make  
./Lab\_04\_Console/Lab\_04\_Console -h

Дополнительно поддерживаются другие компиляторы и платформа Windows.

Linux, Clang:

make CC=clang

Windows, MSVC

VsDevCmd.bat -arch=amd64  
mingw32-make -f Makefile-cl.mk

Windows, Clang:

VsDevCmd.bat -arch=amd64  
mingw32-make -f Makefile-cl.mk CC=clang-cl

Windows, MinGW:

mingw32-make -f Makefile-mingw.mk

# Описание использованных при выполнении задания функций Linux API управления файлами

**int open(const char \*pathname, int flags);**

**open()** используется, чтобы преобразовать путь к файлу в описатель файла (небольшое неотрицательно целое число, которое используется с вызовами read, write и т.п. при последующем вводе-выводе). Если системный вызов завершается успешно, возвращенный файловый описатель является наименьшим описателем, который еще не открыт процессом. В качестве аргументов принимается строка с именем файла и флаги, определяющие способ открытия. С помощью данной функции можно создавать новый файл, для этого предназначен третий аргумент функции – права доступа к файлу.

**int close(int fd);**

**close()** закрывает файловый дескриптор, который после этого не ссылается ни на один и файл и может быть использован повторно. Все блокировки, находящиеся на соответствующем файле, снимаются (независимо от того, был ли использован для установки блокировки именно этот файловый дескриптор). Если fd является последней копией какого-либо файлового дескриптора, то ресурсы, связанные с ним, освобождаются; если дескриптор был последней ссылкой на файл, удаленный с помощью unlink(), то файл окончательно удаляется. Закрытие файла вызывает разную реакцию ОС, в зависимости от типа файла. Например, закрытие сокета может сопровождаться разрывом сетевого соединения между двумя сетевыми узлами, взаимодействующими через сокет.

**ssize\_t read(int fd, void \*buf, size\_t count);**

**read()** пытается записать count байтов файлового описателя fd в буфер, адрес которого начинается с buf. Очевидно, что файл должен быть открыт к моменту выполнения данной операции. Природа данных, которые считываются из файла посредством функции read(), ей неинтересна. Она работает с байтовыми последовательностями.

**ssize\_t write(int fd, const void \*buf, size\_t count);**

**write()** записывает до count байтов из буфера buf в файл, на который ссылается файловый описатель fd. Очевидно, что файл должен быть открыт к моменту выполнения данной операции. Природа данных, которые записываются посредством функции write(), ей неинтересна.

**int unlink(const char \*pathname);**

**unlink()** удаляет имя из файловой системы. Если это имя было последней ссылкой на файл и больше нет процессов, которые держат этот файл открытым, данный файл удаляется и место, которое он занимает освобождается для дальнейшего использования. На самом деле эта функция всего лишь удаляет элемент из каталога, в котором располагается указанный файл. Если это имя было последней ссылкой на файл, и больше нет процессов, которые держат этот файл открытым, данный файл удаляется, и место, которое он занимает, освобождается для дальнейшего использования. Если имя было последней ссылкой на файл, но какие-либо процессы все еще держат этот файл открытым, то файл будет оставлен пока последний дескриптор, указывающий на него, не будет закрыт. Если имя указывает на символьную ссылку, ссылка будет удалена. Если имя указывает на сокет, именованный конвейер или устройство, то имя будет удалено, но процессы, которые открыли любой из этих объектов, могут продолжать его использовать.

**off\_t lseek(int fildes, off\_t offset, int whence);**

**lseek()** устанавливает смещение для файлового дескриптора fildes в значение аргумента offset в соответствии с директивой whence. Функция возвращает новую позицию файлового указателя, или -1, в случае ошибки. Если аргумент whence равен SEEK\_SET, то второй аргумент интерпретируется как смещение от начала файла. Если whence равен SEEK\_CUR, то смещение производится от текущей позиции в файле. Если whence равен SEEK\_END, то аргумент offset интерпретируется как смещение от конца файла. Функция неприменима к некоторым типам файлов, например, сокетам.

**int ftruncate(int fd, off\_t length);**

ftruncate() устанавливает длину обычного файла с файловым дескриптором fd в length байт. Если файл до этой операции был длиннее, то отсеченные данные теряются. Если файл был короче, то он увеличивается, а добавленная часть заполняется нулевыми байтами.

**int access(const char \*pathname, int mode);**

**access()** проверяет, имеет ли процесс права на чтение или запись, или же просто проверяет, существует ли файл (или другой объект файловой системы), с именем pathname. Если pathname является символьной ссылкой, то проверяются права доступа к файлу, на который она ссылается. определяет, имеет ли вызвавший ее процесс право доступа к указанному файлу. Функция способна проверить любую комбинация привилегий чтения, записи, выполнения и существования файла. СЛАЙД 32.Она принимает два аргумента: имя проверяемого файла и битовое объединение флагов R\_OK, W\_OK и X\_OK. Они соответствуют правам чтения, записи и выполнения. При наличии у процесса привилегий возвращается 0. Если файл существует, а нужные привилегии отсутствуют, то возвращается -1, а в переменной errno оказывается значение EACCESS либо EROFS.

# Примеры использования (вызова) этих функций в представленном программном коде

## Обертки над функциями низкоуровнего ввода-вывода из VS (например, \_open()) или Linux API (например open()) (Lab\_04\_Lib/IOWrapper.c):

**int** OpenWrap(**char**\* path, **int** flags, **int** mode)  
{  
#ifdef **\_MSC\_VER  
 return** \_open(path, flags, mode);  
#else  
 **return** open(path, flags, mode);  
#endif  
}  
  
**int** CloseWrap(**int** fd)  
{  
#ifdef **\_MSC\_VER  
 return** \_close(fd);  
#else  
 **return** close(fd);  
#endif  
}  
  
**int** ReadWrap(**int** fd, **void**\* pBuffer, **unsigned int** size)  
{  
#ifdef **\_MSC\_VER  
 return** \_read(fd, pBuffer, size);  
#else  
 **return** read(fd, pBuffer, size);  
#endif  
}  
  
**int** WriteWrap(**int** fd, **void**\* pBuffer, **unsigned int** size)  
{  
#ifdef **\_MSC\_VER  
 return** \_write(fd, pBuffer, size);  
#else  
 **return** write(fd, pBuffer, size);  
#endif  
}  
  
**int** UnlinkWrap(**char**\* path)  
{  
#ifdef **\_MSC\_VER  
 return** \_unlink(path);  
#else  
 **return** unlink(path);  
#endif  
}  
  
**int** LseekWrap(**int** fd, **int** offset, **int** whence)  
{  
#ifdef **\_MSC\_VER  
 return** \_lseek(fd, offset, whence);  
#else  
 **return** lseek(fd, offset, whence);  
#endif  
}  
  
**int** FtruncateWrap(**int** fd, **unsigned int** length)  
{  
#ifdef **\_MSC\_VER  
 return** \_chsize(fd, length);  
#else  
 **return** ftruncate(fd, length);  
#endif  
}  
  
**int** AccessWrap(**char**\* path, **int** mode)  
{  
#ifdef **\_MSC\_VER  
 return** \_access(path, mode);  
#else  
 **return** access(path, mode);  
#endif  
}

## Использование lseek() для перемещение позиции в файле на место указанной записи (Lab\_04\_Lib/RecordFile.c):

**int** SeekRecord(**int** fd, Meta\* pMeta, **int** index)  
{  
 **if** (index == **META\_INDEX**)  
 {  
 **return** LseekWrap(fd, 0, **SEEK\_SET**);  
 }  
 **if** (pMeta == **NULL**)  
 {  
 **return FILE\_UNSUCCESSFUL**;  
 }  
 **return** LseekWrap(fd, **sizeof**(Meta) + index \* pMeta->RecordSize, **SEEK\_SET**);  
}

## Использование write() для записи записи в файл с записями (Lab\_04\_Lib/RecordFile.c):

**int** WriteRecord(**int** fd, Meta\* pMeta, **void**\* pRecord, **int** index)  
{  
 **RETURN\_IF\_NOT\_SUCCESSFUL**(SeekRecord(fd, pMeta, index));  
 **return** WriteWrap(fd, pRecord, pMeta->RecordSize);  
}

## Использование close() для закрытия файла с записями (Lab\_04\_Lib/RecordFile.c):

**int** CloseRecordFile(**int** fd)  
{  
 **return** CloseWrap(fd);  
}

## Использование read() для чтения записи из файла с записями (Lab\_04\_Lib/RecordFile.c):

**int** ReadMeta(**int** fd, Meta\* pMeta)  
{  
 **RETURN\_IF\_NOT\_SUCCESSFUL**(SeekRecord(fd, pMeta, META\_INDEX));  
 **return** ReadWrap(fd, pMeta, **sizeof**(Meta));  
}

## Использование ftruncate() для отсечения записей в файле (Lab\_04\_Lib/RecordFile.c):

**int** ChangeSizeRecordFile(**int** fd, Meta\* pMeta, **int** n)  
{  
 **if** (n > pMeta->Count)  
 {  
 **return FILE\_UNSUCCESSFUL**;  
 }  
 **return** FtruncateWrap(fd, **sizeof**(Meta) + n \* pMeta->RecordSize);  
}

## Использование access() для проверки файла на существование, читаемость и записываемость (Lab\_04\_Lib/RecordFile.c):

**bool** IsExistWritableReadableRecordFile(**char**\* path)  
{  
 **return** AccessWrap(path, **READ\_WRITE\_ACCESS**) != **FILE\_UNSUCCESSFUL**;  
}

# Содержимое файла makefile.

## ./Makefile

MAKE = make  
  
.PHONY: Lab\_04\_Lib Lab\_04\_Console clean  
  
all: Lab\_04\_Console Lab\_04\_Lib  
  
Lab\_04\_Lib:  
 **$(MAKE)** -C $@  
  
Lab\_04\_Console: Lab\_04\_Lib  
 **$(MAKE)** -C $@  
  
clean:  
 **$(MAKE)** $@ -C Lab\_04\_Lib  
 **$(MAKE)** $@ -C Lab\_04\_Console

## ./Lab\_04\_Lib/Makefile

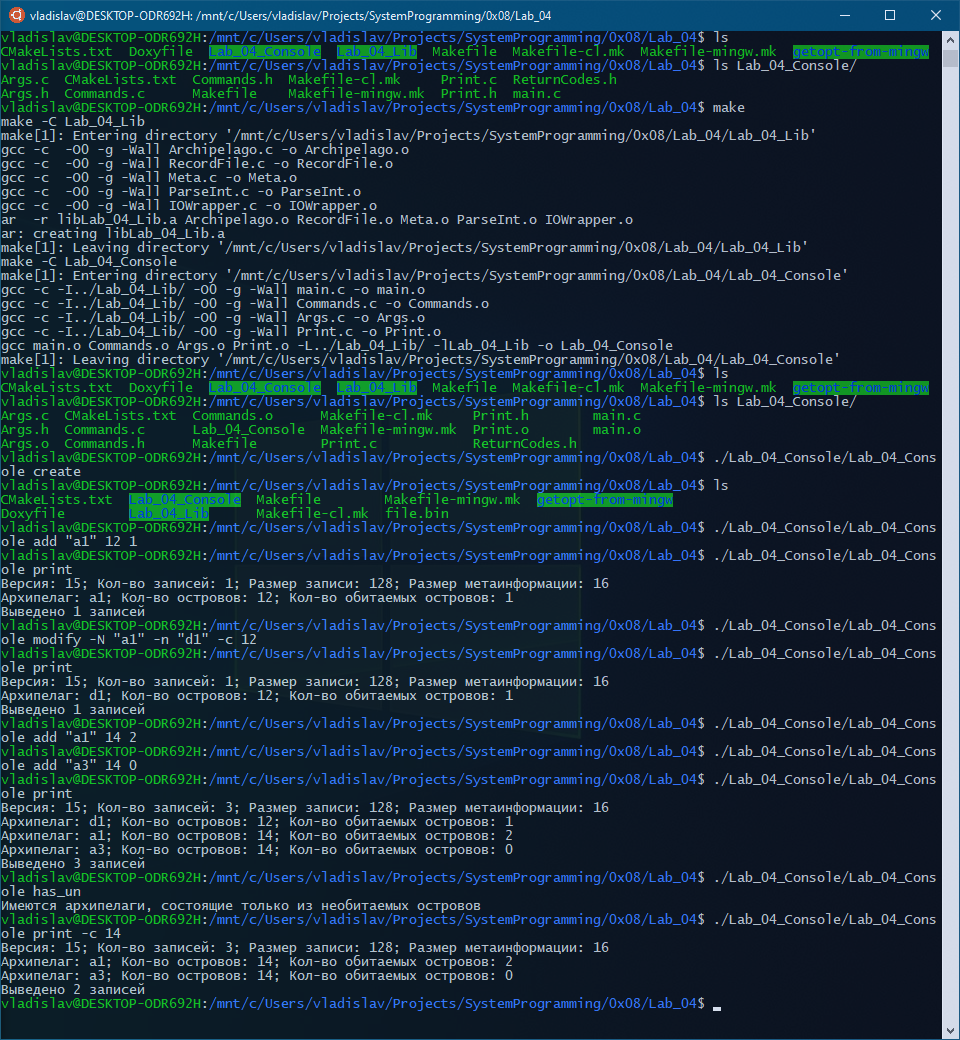
CC = gcc  
AR = ar  
  
  
CFLAGS = -O0 -g -Wall  
LDFLAGS =  
INC =  
  
  
  
SOURCES = Archipelago.c RecordFile.c Meta.c ParseInt.c IOWrapper.c  
OBJECTS = **$(SOURCES:.c=.o)**OUTFILE = libLab\_04\_Lib.a  
  
all: **$(SOURCES) $(OUTFILE)  
  
$(OUTFILE)**: **$(OBJECTS)  
 $(AR) $(LDFLAGS)** -r **$(OUTFILE) $(OBJECTS)**%.o: ***%***.c  
 **$(CC)** -c **$(INC) $(CFLAGS)** $< -o $@  
  
.PHONY: clean  
clean:  
 rm **$(OUTFILE)** \*.o

## ./Lab\_04\_Console/Makefile

CC = gcc  
  
  
  
CFLAGS = -O0 -g -Wall  
LDFLAGS = -L../Lab\_04\_Lib/ -lLab\_04\_Lib  
INC = -I../Lab\_04\_Lib/  
  
  
  
SOURCES = main.c Commands.c Args.c Print.c  
OBJECTS = **$(SOURCES:.c=.o)**OUTFILE = Lab\_04\_Console  
  
all: **$(SOURCES) $(OUTFILE)  
  
$(OUTFILE)**: **$(OBJECTS)  
 $(CC) $(OBJECTS) $(LDFLAGS)** -o $@  
  
%.o: ***%***.c  
 **$(CC)** -c **$(INC) $(CFLAGS)** $< -o $@  
  
.PHONY: clean  
clean:  
 rm **$(OUTFILE)** \*.o

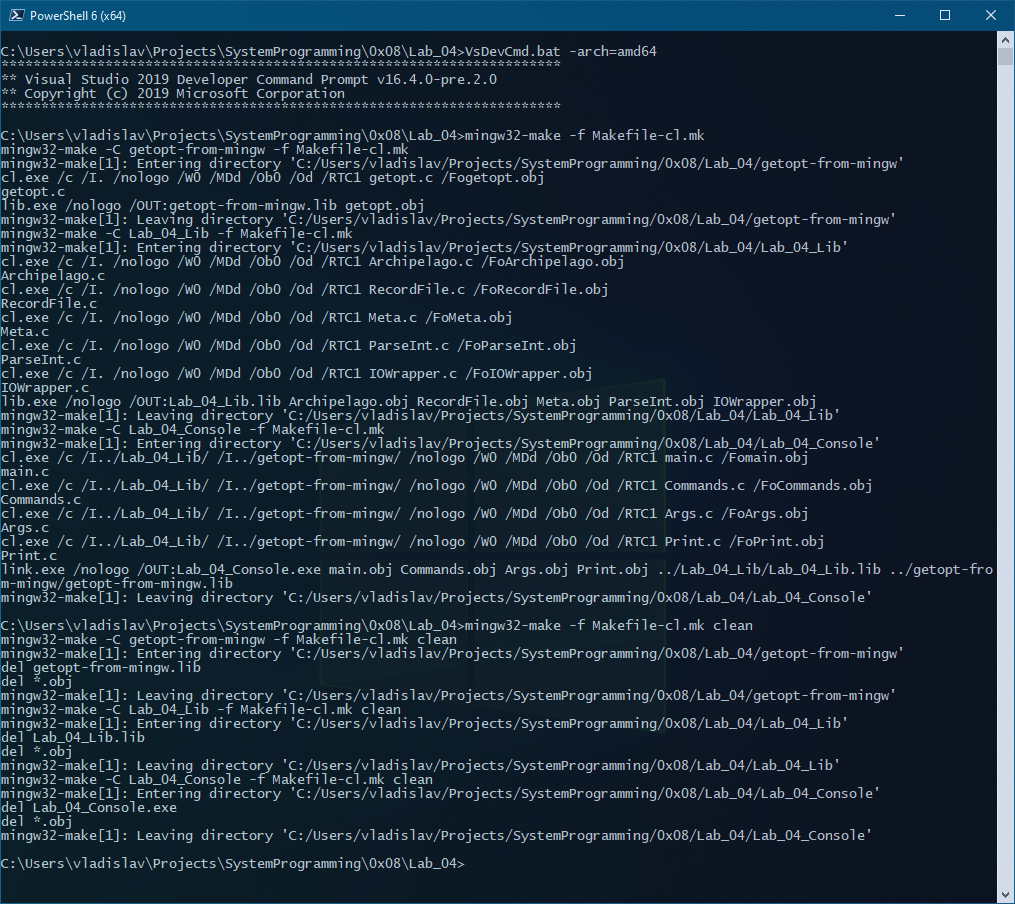
# Примеры работы программ в виде перехватов содержимого экрана

## Снимок экрана запуска № 1 (WSL Ubuntu 18.04, make, GCC)



1. Сборка используя GNU Make и пример работы нескольких команд

## Снимок экрана запуска № 2 (Windows 10, mingw32-make, MSVC)



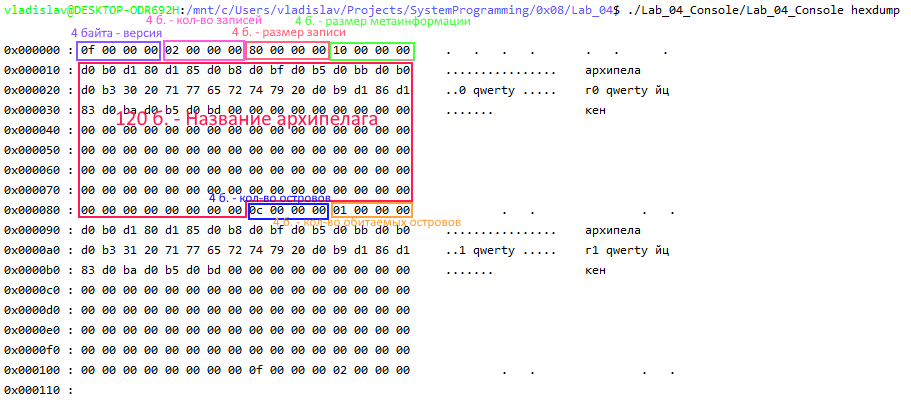
1. Сборка используя mingw32-make и MSVC на Windows и очистка

## Запуск №3 (WSL Ubuntu 18.04, make, Clang)

vladislav@DESKTOP-ODR692H:/mnt/c/Users/vladislav/Projects/SystemProgramming/0x08/Lab\_04**$** make CC=clang-8  
make -C Lab\_04\_Lib  
make[1]: Entering directory **'/mnt/c/Users/vladislav/Projects/SystemProgramming/0x08/Lab\_04/Lab\_04\_Lib'**clang-8 -c -O0 -g -Wall Archipelago.c -o Archipelago.o  
clang-8 -c -O0 -g -Wall RecordFile.c -o RecordFile.o  
clang-8 -c -O0 -g -Wall Meta.c -o Meta.o  
clang-8 -c -O0 -g -Wall ParseInt.c -o ParseInt.o  
clang-8 -c -O0 -g -Wall IOWrapper.c -o IOWrapper.o  
ar -r libLab\_04\_Lib.a Archipelago.o RecordFile.o Meta.o ParseInt.o IOWrapper.o  
ar: creating libLab\_04\_Lib.a  
make[1]: Leaving directory **'/mnt/c/Users/vladislav/Projects/SystemProgramming/0x08/Lab\_04/Lab\_04\_Lib'**make -C Lab\_04\_Console  
make[1]: Entering directory **'/mnt/c/Users/vladislav/Projects/SystemProgramming/0x08/Lab\_04/Lab\_04\_Console'**clang-8 -c -I../Lab\_04\_Lib/ -O0 -g -Wall main.c -o main.o  
clang-8 -c -I../Lab\_04\_Lib/ -O0 -g -Wall Commands.c -o Commands.o  
clang-8 -c -I../Lab\_04\_Lib/ -O0 -g -Wall Args.c -o Args.o  
clang-8 -c -I../Lab\_04\_Lib/ -O0 -g -Wall Print.c -o Print.o  
clang-8 main.o Commands.o Args.o Print.o -L../Lab\_04\_Lib/ -lLab\_04\_Lib -o Lab\_04\_Console  
make[1]: Leaving directory **'/mnt/c/Users/vladislav/Projects/SystemProgramming/0x08/Lab\_04/Lab\_04\_Console'**vladislav@DESKTOP-ODR692H:/mnt/c/Users/vladislav/Projects/SystemProgramming/0x08/Lab\_04**$** ./Lab\_04\_Console/Lab\_04\_Console create  
vladislav@DESKTOP-ODR692H:/mnt/c/Users/vladislav/Projects/SystemProgramming/0x08/Lab\_04**$** ./Lab\_04\_Console/Lab\_04\_Console add -P **"aqwewqewqeqwewqew1"** 12 1  
Версия: 15; Кол-во записей: 1; Размер записи: 128; Размер метаинформации: 16  
Архипелаг: aqwewqewqeqwewqew1; Кол-во островов: 12; Кол-во обитаемых островов: 1  
Выведено 1 записей  
vladislav@DESKTOP-ODR692H:/mnt/c/Users/vladislav/Projects/SystemProgramming/0x08/Lab\_04**$** ./Lab\_04\_Console/Lab\_04\_Console add -P **"aqwewqewqeqwewqew2"** 13 1  
Версия: 15; Кол-во записей: 2; Размер записи: 128; Размер метаинформации: 16  
Архипелаг: aqwewqewqeqwewqew1; Кол-во островов: 12; Кол-во обитаемых островов: 1  
Архипелаг: aqwewqewqeqwewqew2; Кол-во островов: 13; Кол-во обитаемых островов: 1  
Выведено 2 записей  
vladislav@DESKTOP-ODR692H:/mnt/c/Users/vladislav/Projects/SystemProgramming/0x08/Lab\_04**$** ./Lab\_04\_Console/Lab\_04\_Console add -P **"aqwewqewqeqwewqew3"** 14 2  
Версия: 15; Кол-во записей: 3; Размер записи: 128; Размер метаинформации: 16  
Архипелаг: aqwewqewqeqwewqew1; Кол-во островов: 12; Кол-во обитаемых островов: 1  
Архипелаг: aqwewqewqeqwewqew2; Кол-во островов: 13; Кол-во обитаемых островов: 1  
Архипелаг: aqwewqewqeqwewqew3; Кол-во островов: 14; Кол-во обитаемых островов: 2  
Выведено 3 записей  
vladislav@DESKTOP-ODR692H:/mnt/c/Users/vladislav/Projects/SystemProgramming/0x08/Lab\_04**$** ./Lab\_04\_Console/Lab\_04\_Console add -P **"aqwewqewqeqwewqew4"** 14 2  
Версия: 15; Кол-во записей: 4; Размер записи: 128; Размер метаинформации: 16  
Архипелаг: aqwewqewqeqwewqew1; Кол-во островов: 12; Кол-во обитаемых островов: 1  
Архипелаг: aqwewqewqeqwewqew2; Кол-во островов: 13; Кол-во обитаемых островов: 1  
Архипелаг: aqwewqewqeqwewqew3; Кол-во островов: 14; Кол-во обитаемых островов: 2  
Архипелаг: aqwewqewqeqwewqew4; Кол-во островов: 14; Кол-во обитаемых островов: 2  
Выведено 4 записей  
vladislav@DESKTOP-ODR692H:/mnt/c/Users/vladislav/Projects/SystemProgramming/0x08/Lab\_04**$** ./Lab\_04\_Console/Lab\_04\_Console add -P **"aqwewqewqeqwewqew5"** 15 2  
Версия: 15; Кол-во записей: 5; Размер записи: 128; Размер метаинформации: 16  
Архипелаг: aqwewqewqeqwewqew1; Кол-во островов: 12; Кол-во обитаемых островов: 1  
Архипелаг: aqwewqewqeqwewqew2; Кол-во островов: 13; Кол-во обитаемых островов: 1  
Архипелаг: aqwewqewqeqwewqew3; Кол-во островов: 14; Кол-во обитаемых островов: 2  
Архипелаг: aqwewqewqeqwewqew4; Кол-во островов: 14; Кол-во обитаемых островов: 2  
Архипелаг: aqwewqewqeqwewqew5; Кол-во островов: 15; Кол-во обитаемых островов: 2  
Выведено 5 записей  
vladislav@DESKTOP-ODR692H:/mnt/c/Users/vladislav/Projects/SystemProgramming/0x08/Lab\_04**$** ./Lab\_04\_Console/Lab\_04\_Console print -c 14  
Версия: 15; Кол-во записей: 5; Размер записи: 128; Размер метаинформации: 16  
Архипелаг: aqwewqewqeqwewqew3; Кол-во островов: 14; Кол-во обитаемых островов: 2  
Архипелаг: aqwewqewqeqwewqew4; Кол-во островов: 14; Кол-во обитаемых островов: 2  
Выведено 2 записей  
vladislav@DESKTOP-ODR692H:/mnt/c/Users/vladislav/Projects/SystemProgramming/0x08/Lab\_04**$** ./Lab\_04\_Console/Lab\_04\_Console print -c 12  
Версия: 15; Кол-во записей: 5; Размер записи: 128; Размер метаинформации: 16  
Архипелаг: aqwewqewqeqwewqew1; Кол-во островов: 12; Кол-во обитаемых островов: 1  
Выведено 1 записей  
vladislav@DESKTOP-ODR692H:/mnt/c/Users/vladislav/Projects/SystemProgramming/0x08/Lab\_04**$** ./Lab\_04\_Console/Lab\_04\_Console print -i 1  
Версия: 15; Кол-во записей: 5; Размер записи: 128; Размер метаинформации: 16  
Архипелаг: aqwewqewqeqwewqew1; Кол-во островов: 12; Кол-во обитаемых островов: 1  
Архипелаг: aqwewqewqeqwewqew2; Кол-во островов: 13; Кол-во обитаемых островов: 1  
Выведено 2 записей  
vladislav@DESKTOP-ODR692H:/mnt/c/Users/vladislav/Projects/SystemProgramming/0x08/Lab\_04**$** ./Lab\_04\_Console/Lab\_04\_Console print -i 2  
Версия: 15; Кол-во записей: 5; Размер записи: 128; Размер метаинформации: 16  
Архипелаг: aqwewqewqeqwewqew3; Кол-во островов: 14; Кол-во обитаемых островов: 2  
Архипелаг: aqwewqewqeqwewqew4; Кол-во островов: 14; Кол-во обитаемых островов: 2  
Архипелаг: aqwewqewqeqwewqew5; Кол-во островов: 15; Кол-во обитаемых островов: 2  
Выведено 3 записей  
vladislav@DESKTOP-ODR692H:/mnt/c/Users/vladislav/Projects/SystemProgramming/0x08/Lab\_04**$** ./Lab\_04\_Console/Lab\_04\_Console print -c 14 -i 2  
Версия: 15; Кол-во записей: 5; Размер записи: 128; Размер метаинформации: 16  
Архипелаг: aqwewqewqeqwewqew3; Кол-во островов: 14; Кол-во обитаемых островов: 2  
Архипелаг: aqwewqewqeqwewqew4; Кол-во островов: 14; Кол-во обитаемых островов: 2  
Выведено 2 записей  
vladislav@DESKTOP-ODR692H:/mnt/c/Users/vladislav/Projects/SystemProgramming/0x08/Lab\_04**$** ./Lab\_04\_Console/Lab\_04\_Console print -o -c 14 -i 2  
Версия: 15; Кол-во записей: 5; Размер записи: 128; Размер метаинформации: 16  
Архипелаг: aqwewqewqeqwewqew3; Кол-во островов: 14; Кол-во обитаемых островов: 2  
Архипелаг: aqwewqewqeqwewqew4; Кол-во островов: 14; Кол-во обитаемых островов: 2  
Архипелаг: aqwewqewqeqwewqew5; Кол-во островов: 15; Кол-во обитаемых островов: 2  
Выведено 3 записей  
vladislav@DESKTOP-ODR692H:/mnt/c/Users/vladislav/Projects/SystemProgramming/0x08/Lab\_04**$** ./Lab\_04\_Console/Lab\_04\_Console has\_un  
Отсутствуют архипелаги, состоящие только из необитаемых островов  
vladislav@DESKTOP-ODR692H:/mnt/c/Users/vladislav/Projects/SystemProgramming/0x08/Lab\_04**$** ./Lab\_04\_Console/Lab\_04\_Console modify -I 3 -P -i 0  
Версия: 15; Кол-во записей: 5; Размер записи: 128; Размер метаинформации: 16  
Архипелаг: aqwewqewqeqwewqew1; Кол-во островов: 12; Кол-во обитаемых островов: 1  
Архипелаг: aqwewqewqeqwewqew2; Кол-во островов: 13; Кол-во обитаемых островов: 1  
Архипелаг: aqwewqewqeqwewqew3; Кол-во островов: 14; Кол-во обитаемых островов: 2  
Архипелаг: aqwewqewqeqwewqew4; Кол-во островов: 14; Кол-во обитаемых островов: 0  
Архипелаг: aqwewqewqeqwewqew5; Кол-во островов: 15; Кол-во обитаемых островов: 2  
Выведено 5 записей  
vladislav@DESKTOP-ODR692H:/mnt/c/Users/vladislav/Projects/SystemProgramming/0x08/Lab\_04**$** ./Lab\_04\_Console/Lab\_04\_Console has\_un  
Имеются архипелаги, состоящие только из необитаемых островов  
vladislav@DESKTOP-ODR692H:/mnt/c/Users/vladislav/Projects/SystemProgramming/0x08/Lab\_04**$** ./Lab\_04\_Console/Lab\_04\_Console remove -I 2 -P  
Версия: 15; Кол-во записей: 4; Размер записи: 128; Размер метаинформации: 16  
Архипелаг: aqwewqewqeqwewqew1; Кол-во островов: 12; Кол-во обитаемых островов: 1  
Архипелаг: aqwewqewqeqwewqew2; Кол-во островов: 13; Кол-во обитаемых островов: 1  
Архипелаг: aqwewqewqeqwewqew4; Кол-во островов: 14; Кол-во обитаемых островов: 0  
Архипелаг: aqwewqewqeqwewqew5; Кол-во островов: 15; Кол-во обитаемых островов: 2  
Выведено 4 записей  
vladislav@DESKTOP-ODR692H:/mnt/c/Users/vladislav/Projects/SystemProgramming/0x08/Lab\_04**$** ./Lab\_04\_Console/Lab\_04\_Console remove -I 1 -s -P  
Версия: 15; Кол-во записей: 3; Размер записи: 128; Размер метаинформации: 16  
Архипелаг: aqwewqewqeqwewqew1; Кол-во островов: 12; Кол-во обитаемых островов: 1  
Архипелаг: aqwewqewqeqwewqew5; Кол-во островов: 15; Кол-во обитаемых островов: 2  
Архипелаг: aqwewqewqeqwewqew4; Кол-во островов: 14; Кол-во обитаемых островов: 0  
Выведено 3 записей  
vladislav@DESKTOP-ODR692H:/mnt/c/Users/vladislav/Projects/SystemProgramming/0x08/Lab\_04**$** ./Lab\_04\_Console/Lab\_04\_Console modify -I 0 -n **"архипелаг0 qwerty йцукен"**vladislav@DESKTOP-ODR692H:/mnt/c/Users/vladislav/Projects/SystemProgramming/0x08/Lab\_04**$** ./Lab\_04\_Console/Lab\_04\_Console print  
Версия: 15; Кол-во записей: 3; Размер записи: 128; Размер метаинформации: 16  
Архипелаг: архипелаг0 qwerty йцукен; Кол-во островов: 12; Кол-во обитаемых островов: 1  
Архипелаг: aqwewqewqeqwewqew5; Кол-во островов: 15; Кол-во обитаемых островов: 2  
Архипелаг: aqwewqewqeqwewqew4; Кол-во островов: 14; Кол-во обитаемых островов: 0  
Выведено 3 записей  
vladislav@DESKTOP-ODR692H:/mnt/c/Users/vladislav/Projects/SystemProgramming/0x08/Lab\_04**$** ./Lab\_04\_Console/Lab\_04\_Console modify -I 1 -n **"архипелаг1 qwerty йцукен"** -P  
Версия: 15; Кол-во записей: 3; Размер записи: 128; Размер метаинформации: 16  
Архипелаг: архипелаг0 qwerty йцукен; Кол-во островов: 12; Кол-во обитаемых островов: 1  
Архипелаг: архипелаг1 qwerty йцукен; Кол-во островов: 15; Кол-во обитаемых островов: 2  
Архипелаг: aqwewqewqeqwewqew4; Кол-во островов: 14; Кол-во обитаемых островов: 0  
Выведено 3 записей  
vladislav@DESKTOP-ODR692H:/mnt/c/Users/vladislav/Projects/SystemProgramming/0x08/Lab\_04**$** ./Lab\_04\_Console/Lab\_04\_Console remove -n **"aqwewqewqeqwewqew4"** -P  
Версия: 15; Кол-во записей: 2; Размер записи: 128; Размер метаинформации: 16  
Архипелаг: архипелаг0 qwerty йцукен; Кол-во островов: 12; Кол-во обитаемых островов: 1  
Архипелаг: архипелаг1 qwerty йцукен; Кол-во островов: 15; Кол-во обитаемых островов: 2  
Выведено 2 записей  
vladislav@DESKTOP-ODR692H:/mnt/c/Users/vladislav/Projects/SystemProgramming/0x08/Lab\_04**$**

## Шестнадцатеричный дамп после запуска №4

Первые 16 байтов в файле – метаинформация. Затем идут записи, каждая по 128 байт.



1. Вывод команды hexdump и объяснение байтов