Лабораторная работа 1: ввод-вывод

Цель работы:

Познакомиться с потоками ввода-вывода стандартной библиотеки C++.

Реализовать с их помощью утилиты myhead и myhexdump.

Структура проекта:

task01-io

|-- .clang-format [опционально]

|-- .gitignore

|-- .gitlab-ci.yml [создан автоматически]

|-- Makefile

|-- myhead.cpp

`-- myhexdump.cpp

После выполнения команды make должны создаваться два исполняемых файла: myhead и myhexdump.

Пример мейкфайла:

CXXFLAGS=-g -Wall -Wextra -Werror -std=c++20

all: myhead myhexdump

**Задание 1: myhead**

Разработайте утилиту myhead, аналог стандартной

head(1). Утилита выводит lines первых

строк или bytes первых байт файла file. Если опции не переданы, то выводит

десять первых строк.

Синтаксис запуска: myhead <file> [[-n <lines>] | [-c <bytes>]]

Здесь и далее угловыми скобками <> будем обозначать обязательные параметры,

квадратными [] — опциональные.

При возникновении ошибки утилита завершается с кодом 1 и выводит в stderr

сообщение об ошибке.

Примеры запуска:

$ ./myhead /etc/locale.gen

# This file lists locales that you wish to have built. You can find a list

# of valid supported locales at /usr/share/i18n/SUPPORTED, and you can add

# user defined locales to /usr/local/share/i18n/SUPPORTED. If you change

# this file, you need to rerun locale-gen.

# aa\_DJ ISO-8859-1

# aa\_DJ.UTF-8 UTF-8

# aa\_ER UTF-8

# aa\_ER@saaho UTF-8

$ ./myhead /etc/locale.gen -n 2

# This file lists locales that you wish to have built. You can find a list

# of valid supported locales at /usr/share/i18n/SUPPORTED, and you can add

$ ./myhead /etc/locale.gen -c 6

# This

$ ./myhead /etc/locale.gen -n -2

Error: number of lines must be non-negative.

Обработку опций командной строки можете реализовать вручную или

воспользоваться функцией getopt(3).

**Задание 2: myhexdump**

Синтаксис запуска: myhexdump <file>

Пример запуска:

$ ./myhexdump Makefile

00000000 4f 50 54 4c 56 4c 20 3f 3d 20 2d 4f 33 0a 43 58 |OPTLVL ?= -O3.CX|

00000010 58 46 4c 41 47 53 20 3f 3d 20 2d 73 74 64 3d 63 |XFLAGS ?= -std=c|

00000020 2b 2b 32 30 20 2d 57 61 6c 6c 20 2d 57 65 78 74 |++20 -Wall -Wext|

00000030 72 61 20 2d 57 65 72 72 6f 72 20 2d 70 65 64 61 |ra -Werror -peda|

00000040 6e 74 69 63 20 24 28 4f 50 54 4c 56 4c 29 0a 0a |ntic $(OPTLVL)..|

00000050 61 6c 6c 3a 20 68 65 6c 6c 6f 0a |all: hello.|

0000005b

Формат одной строки вывода:

1. Смещение от начала файла в шестнадцатеричном формате. Выводятся 8 разрядов, дополненных нулями слева.
2. 2 пробела.
3. 8 байт в шестнадцатеричной системе счисления с лидирующим нулем, разделенные пробелом. Значения [a-d] в нижнем регистре.
4. 2 пробела.
5. 8 байт в формате из п.3.
6. 2 пробела.
7. Разделитель |.
8. Текстовое представление выведенных 8 байт. Если значению байта соответствует буква, цифра, знак препинания в ascii или пробел, то вывести этот символ, иначе вывести точку.
9. Разделитель |.

Если у вас возникают вопросы по формату, ориентируйтесь на вывод hexdump -C.

Для форматированного вывода используйте инструмент на выбор:

* std::format (Standard format specification).
* Манипуляторы.

**Задание 3: доработка myhead и myhexdump**

Доработайте утилиты myhead и myhexdump таким образом, чтобы позиционный

параметр file стал необязательным. Если файл не передан, то утилиты должны

читать данные со стандартного потока ввода.

После этого появится возможность объединять утилиты в конвейер:

$ ./myhexdump /bin/ls | ./myhead -n 2

00000000 7f 45 4c 46 02 01 01 00 00 00 00 00 00 00 00 00 |.ELF............|

00000010 03 00 3e 00 01 00 00 00 a0 6a 00 00 00 00 00 00 |..>......j......|

$ ./myhead /bin/ls -c 8 | ./myhexdump

00000000 7f 45 4c 46 02 01 01 00 |.ELF....|

00000008