

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования
«Гомельский государственный технический университет
имени П.О.Сухого»

Кафедра: «Информатика»
по курсу: «Операционные системы и среды»

Лабораторная работа №4
«Знакомство со стандартной утилитой
Gnu make для построения проектов в ОС Unix/Linux»

Допуск к защите:
Дата защиты:

Выполнил: студент группы ИП-31
Земченок Вадим Геннадьевич
Проверил: старший преподаватель
Самовендюк Николай Владимирович

Гомель 2020 г.

Цель: Ознакомиться с техникой компиляции программ на языке программирования C (C++) в среде ОС семейства Unix/Linux, а также получить практические навыки использования утилиты GNU make для сборки проекта.

Задание 1.

1. Ознакомиться с теоретическим материалом.
2. Воспользоваться утилитой make для автоматизации сборки проектов из лабораторной работы №4 «Инструментальные средства разработки Linux»
3. Создать make-файл с высоким уровнем автоматизированной обработки исходных файлов программы согласно следующим условиям:
 - имя скомпилированной программы (выполняемый или бинарный файл), флаги компиляции и имена каталогов с исходными файлами и бинарными файлами (каталоги, src, bin и т. п.) задаются с помощью переменных в makefile;
 - зависимости исходных файлов на языке C (C++) и цели в make-файле должны формироваться динамически;
 - наличие цели clean, удаляющей временные файлы;
 - каталог проекта должен быть структурирован следующим образом:
src — каталог с исходными файлами;
bin — каталог с бинарными файлами (скомпилированными);
makefile.

Вариант	Структура данных (STL)	Функции обработки (таблица 2)	Тип сортировки (таблица 3)
7	Array	Произведение отрицательных элементов на нечетных местах	начальная – обменом, конечная – выбором максимального элемента

Все функции размещаются в отдельных файлах.

4. Выполнить программу (скомпилировать, при необходимости отладить) для первого варианта сортировки данных.
5. Изменить тип сортировки и выполнить программу.
5. Показать, что при изменении одного исходного файла и последующем вызове make будут исполнены только необходимые команды компиляции (неизмененные файлы перекомпилированы не будут) и изменены атрибуты и/или размер объектных файлов (файлы с расширением .o).

Activities Text Editor Cpд 08:49

fileLib.cpp
~/Desktop/lab4/src

```
#include "fileLib.h"
#include <iostream>
#include <array>

using namespace std;

void sort(std::array<int,10> &arr)
{
    int temp;
    for(int i = 0; i < 10; i++)
    {
        for(int j = 0; j < 10-i-1; j++)
        {
            if(arr[j] > arr[j+1])
            {
                temp = arr[j];
                arr[j] = arr[j+1];
                arr[j+1] = temp;
            }
        }
    }
}

void output(array<int, 10> arr)
{
    for(int i = 0; i < 10; i++)
    {
        cout << arr[i] << " ";
    }
    cout << endl;
}

void proizv(array<int, 10> arr)
{
    int s = 1;
    for(int i = 0; i < 10; i+=1)
    {
        if(arr[i] < 0)
            s *= arr[i];
    }
    cout << "PROIZV = " << s << endl;
}
```

fileLib.h
~/Desktop/lab4/src

This file "/home/vadim/Desktop/lab4/src/fileLib.h" is already open in another window.
Do you want to edit it anyway?

Edit Anyway Don't Edit

```
#include <fstream>
#include <array>

using namespace std;

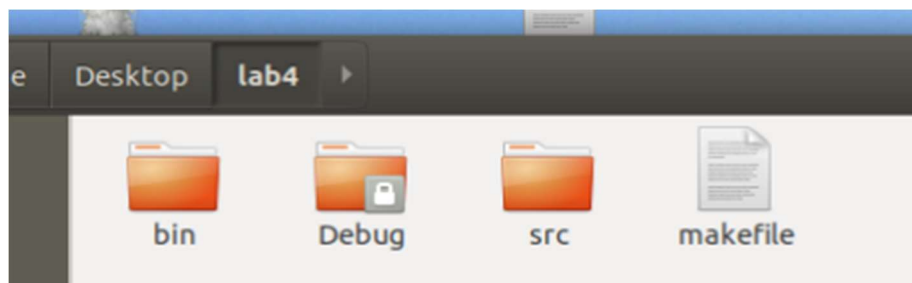
void sort(std::array<int,10> &arr);
void output(array<int, 10> arr);
void proizv(array<int, 10> arr);
```

main.cpp
~/Desktop/lab4/src

```
#include <string>
#include <iterator>
#include <iostream>
#include <algorithm>
#include <array>
#include "fileLib.h"

int main()
{
    setlocale(0, "");

    std::array<int, 10> a1{ 1,2,-3,4,2,-6,-5,8,-2,10 };
    sort(a1);
    proizv(a1);
    output(a1);
}
```



vadim@UbuntuVB: ~/Desktop/lab4/Debug

File Edit View Search Terminal Help

```
vadim@UbuntuVB:~/Desktop/lab4$ sudo make -f makefile
g++ -g -O -I ./src/ -o Debug/./src/fileLib.o -c src/fileLib.cpp
g++ -g -O -I ./src/ -o Debug/./src/main.o -c src/main.cpp
g++ Debug/./src/fileLib.o Debug/./src/main.o -o Debug/main
vadim@UbuntuVB:~/Desktop/lab4$ cd ./Debug
vadim@UbuntuVB:~/Desktop/lab4/Debug$ ./main
-6 -5 -3 -2 1 2 2 4 8 10
vadim@UbuntuVB:~/Desktop/lab4/Debug$
```

```

-6 -5 -3 -2 1 2 2 4 8 10
vadin@UbuntuVB:~/Desktop/lab4/Debug$ cd ./src
vadin@UbuntuVB:~/Desktop/lab4/Debug/src$ ls -l
total 104
-rw-r--r-- 1 root root 85 мая 20 08:42 fileLib.d
-rw-r--r-- 1 root root 52208 мая 20 08:42 fileLib.o
-rw-r--r-- 1 root root 76 мая 20 08:42 main.d
-rw-r--r-- 1 root root 42040 мая 20 08:42 main.o
vadin@UbuntuVB:~/Desktop/lab4/Debug/src$

```

```

vadin@UbuntuVB:~/Desktop/lab4$ sudo make -f makefile
g++ -g -O -I ./src/ -o Debug/./src/fileLib.o -c src/fileLib.cpp
g++ -g -O -I ./src/ -o Debug/./src/main.o -c src/main.cpp
g++ Debug/./src/fileLib.o Debug/./src/main.o -o Debug/main
vadin@UbuntuVB:~/Desktop/lab4$ cd ./Debug
vadin@UbuntuVB:~/Desktop/lab4/Debug$ ./main
-6 -5 -3 -2 1 2 2 4 8 10
vadin@UbuntuVB:~/Desktop/lab4/Debug$ cd ./src
vadin@UbuntuVB:~/Desktop/lab4/Debug/src$ ls -l
total 104
-rw-r--r-- 1 root root 85 мая 20 08:42 fileLib.d
-rw-r--r-- 1 root root 52208 мая 20 08:42 fileLib.o
-rw-r--r-- 1 root root 76 мая 20 08:42 main.d
-rw-r--r-- 1 root root 42040 мая 20 08:42 main.o
vadin@UbuntuVB:~/Desktop/lab4/Debug/src$ cd ..
vadin@UbuntuVB:~/Desktop/lab4/Debug$ cd..
cd..: command not found
vadin@UbuntuVB:~/Desktop/lab4/Debug$ cd ..
vadin@UbuntuVB:~/Desktop/lab4$ sudo make -f makefile
g++ -g -O -I ./src/ -o Debug/./src/fileLib.o -c src/fileLib.cpp
g++ -g -O -I ./src/ -o Debug/./src/main.o -c src/main.cpp
g++ Debug/./src/fileLib.o Debug/./src/main.o -o Debug/main
vadin@UbuntuVB:~/Desktop/lab4$ cd ./Debug
vadin@UbuntuVB:~/Desktop/lab4/Debug$ ./main
PROIZV = 180
-6 -5 -3 -2 1 2 2 4 8 10
vadin@UbuntuVB:~/Desktop/lab4/Debug$ cd ./src
vadin@UbuntuVB:~/Desktop/lab4/Debug/src$ ls -l
total 104
-rw-r--r-- 1 root root 85 мая 20 08:42 fileLib.d
-rw-r--r-- 1 root root 52208 мая 20 08:45 fileLib.o
-rw-r--r-- 1 root root 76 мая 20 08:45 main.d
-rw-r--r-- 1 root root 42336 мая 20 08:45 main.o
vadin@UbuntuVB:~/Desktop/lab4/Debug/src$

```

Листинг make файла

```

#### #####
#
# MAKEFILE FOR C/C++ PROJECT
# Author: swm8023 <swm8023@gmail.com>
# Date: 2014/01/30
#
#### #####

.PHONY: all clean
all:

```

```

# annotation when release version
DEBUG      := y
TARGET_PROG := main

# project directory
DEBUG_DIR  := ./Debug
RELEASE_DIR := ./Release
BIN_DIR    := $(if $(DEBUG), $(DEBUG_DIR), $(RELEASE_DIR))

# shell command
CC      := gcc
CXX     := g++
RM      := rm -rf
MKDIR   := mkdir -p
SED     := sed
MV      := mv

# init sources & objects & depends
sources_all := $(shell find . -name "*.c" -o -name "*.cpp" -o -name "*.h")
sources_c   := $(filter %.c, $(sources_all))
sources_cpp := $(filter %.cpp, $(sources_all))
sources_h   := $(filter %.h, $(sources_all))
objs        := $(addprefix $(BIN_DIR)/,$(strip $(sources_cpp:.cpp=.o)
$(sources_c:.c=.o)))
deps        := $(addprefix $(BIN_DIR)/,$(strip $(sources_cpp:.cpp=.d)
$(sources_c:.c=.d)))

# create directory
$(foreach dirname,$(sort $(dir $(sources_c) $(sources_cpp))),\
$(shell $(MKDIR) $(BIN_DIR)/$(dirname)))

# compile & link variable
CFLAGS     := $(if $(DEBUG),-g -O, -O2)
CFLAGS     += $(addprefix -I ,$(sort $(dir $(sources_h))))
CXXFLAGS   = $(CFLAGS)
LDFLAGS    :=
LOADLIBES  += #-L/usr/include/mysql
LDLIBS     += #-lpthread -lmysqlclient

# add vpath
vpath %.h $(sort $(dir $(sources_h)))
vpath %.c $(sort $(dir $(sources_c)))
vpath %.cpp $(sort $(dir $(sources_cpp)))

# generate depend files
# actually generate after object generated, beacasue it only used when next make)
ifneq "$$(MAKECMDGOALS)" "clean"
sinclude $(deps)
endif

# make-depend(depend-file,source-file,object-file,cc)
define make-depend
$(RM) $1; \
$4 $(CFLAGS) -MM $2 | \
$(SED) 's,\($(notdir $3)\): ,,$3: ,' > $1.tmp; \
$(SED) -e 's/#.*//' \
-e 's/^[^:]*: *//' \
-e 's/ *\$\$//' \
-e '/^$$/ d' \
-e 's/$$/ :/' < $1.tmp >> $1.tmp; \
$(MV) $1.tmp $1;
endif

# rules to generate objects file
$(BIN_DIR)/%.o: %.c

```

```

    @$(call make-depend,$(patsubst %.o,%.d,$@),$<,$@,$(CC))
    $(CC) $(CFLAGS) -o $@ -c $<

$(BIN_DIR)/%.o: %.cpp
    @$(call make-depend,$(patsubst %.o,%.d,$@),$<,$@,$(CXX))
    $(CXX) $(CXXFLAGS) -o $@ -c $<

# add-target(target,objs,cc)
define add-target
    REAL_TARGET += $(BIN_DIR)/$1
    $(BIN_DIR)/$1: $2
        $3 $(LDFLAGS) $$^ $(LOADLIBES) $(LDLIBS) -o $$@
endef

# call add-target
$(foreach targ,$(TARGET_PROG),$(eval $(call add-target,$(targ),$(objs),$(CXX))))

all: $(REAL_TARGET) $(TARGET_LIBS)

clean:
    $(RM) $(BIN_DIR)

```

Вывод: я успешно ознакомился с техникой компиляции программ на языке программирования C (C++) в среде ОС семейства Unix/Linux, а также получил практические навыки использования утилиты GNU make для сборки проекта.