



## Отчет по лабораторной работе № 08 по курсу Фундаментальная информатика

Студент группы М8О-104Б-22 Жарков Александр Алексеевич, № по списку 4

Контакты www, e-mail, icq, skype aleksandrzharkovv@gmail.com

Работа выполнена: « 15 » октября 2022 г.

Преподаватель: асп. каф. 806 Потенко М.А.

Входной контроль знаний с оценкой \_\_\_\_\_

Отчет сдан « 16 » ноября 2022 г., итоговая оценка \_\_\_\_

Подпись преподавателя \_\_\_\_\_

1. **Тема:** Системы программирования на языке Си

2. **Цель работы:** Изучить систему программирования на языке Си, научиться отладке программ

3. **Задание (вариант № ):** Написать простейшую программу, допустить ошибку в синтаксисе, исправить её, заново скомпилировать файл

4. **Оборудование (лабораторное):**  
ЭВМ \_\_\_\_\_, процессор \_\_\_\_\_, имя узла сети \_\_\_\_\_ с ОП \_\_\_\_\_ Мб,  
НМД \_\_\_\_\_ Мб. Терминал \_\_\_\_\_ адрес \_\_\_\_\_. Принтер \_\_\_\_\_  
Другие устройства \_\_\_\_\_

*Оборудование ПЭВМ студента, если использовалось:*

Процессор AMD RYZEN 5 5600H с ОП 16384 Мб, ssd 524288 Мб. Монитор IPS 1920\*1080

Другие устройства \_\_\_\_\_

5. **Программное обеспечение (лабораторное):**  
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 UNIX, наименование Ubuntu версия 20.04 LTS

интерпретатор команд bash версия 5.1.16

Система программирования \_\_\_\_\_ версия \_\_\_\_\_

Редактор текстов gedit версия 3.30.2

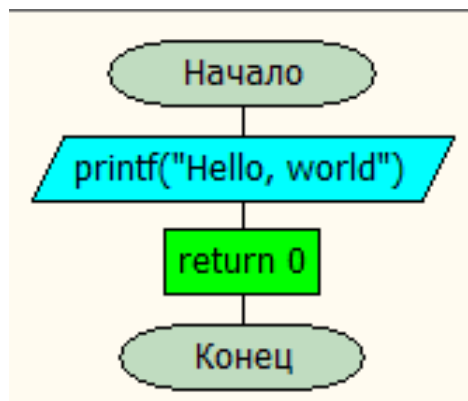
Утилиты операционной системы \_\_\_\_\_

Прикладные системы и программы \_\_\_\_\_

Местонахождение и имена файлов программ и данных на домашнем компьютере \_\_\_\_\_

**6. Идея, метод, алгоритм** решение задачи (в формах: словесной, псевдокода, графической [блок-схема, диаграмма, рисунок, таблица] или формальные спецификации с пред- и постусловиями)

Создадим через консоль файл с расширением .c (с этим расширением создаются файлы, содержащие код на языке Си). Откроем этот файл в текстовом редакторе gedit и напишем программу, которая выведет текст "Hello, world!", преднамеренно допустив несколько ошибок в синтаксисе, чтобы программа не смогла успешно скомпилироваться. Через терминал выполним компиляцию программы, используя ключи -Wall, -Werror, чтобы можно было увидеть все ошибки, допущенные при написании кода. Компиляция не сможет успешно завершиться, потому что компилятором будут обнаружены синтаксические ошибки, которые вместе с описанием будут выведены консолью. Нужно будет прочитать их содержание, описание, понять в чём именно заключается ошибка, исправить все ошибки и в итоге успешно скомпилировать программу и увидеть результат ее работы.



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

Командой mkdir создать рабочую директорию Lab, командой cd Lab перейти в рабочую директорию. Командой echo создать в ней файл lab8.c. Открыть этот файл в текстовом редакторе gedit командой gedit lab8.c. Это файл, в котором будет код, написанный на языке Си. Чтобы в результате работы программы вывести "Hello, world!", нам нужно в начале командой include подключить стандартную библиотеку вводы-вывода <stdio.h>. Затем создадим основную функцию main, которая будет выполняться в нашей программе. В фигурных скобках будут действия, выполняемые при работе функции. Каждую команду в функции будем писать на новой строке, с отступом в 4 пробела (Tab), соблюдая правила оформления кода. Напишем в функции команду printf("Hello, world!"), которая выведет Hello, world! и перенесет строку. Напишем в функции команду return 0, указывающую на завершение работы функции. После каждой команды (в конце каждой строки) поставим точку с запятой в соответствии с синтаксисом языка Си. Специально пропустим точку с запятой после return 0 и не поставим символ > в команде include <stdio.h>. Сохраним файл затем скомпилируем программу и выведем результат ее работы с помощью команд gcc lab8.c и ./a.out. Увидим ошибки в нашей программе, исправим их и заново выполним её.

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

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

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

```
alexandr@alexandr-Magicbook16:~$ mkdir Lab
alexandr@alexandr-Magicbook16:~$ cd Lab
alexandr@alexandr-Magicbook16:~/Lab$ touch lab8.c
alexandr@alexandr-Magicbook16:~/Lab$ gedit lab8.c
alexandr@alexandr-Magicbook16:~/Lab$ gcc lab8.c
lab8.c:1:18: error: missing terminating > character
  1 | #include <stdio.h
    |               ^
lab8.c: In function 'main':
lab8.c:4:17: error: expected ';' before '}' token
  4 |         return 0
    |                ^
  5 |     }
    |     ~
```

```
1 #include <stdio.h>
2 int main(){
3     printf("Hello, world!\n");
4     return 0
5 }
6
```

```
1 #include <stdio.h>
2 int main(){
3     printf("Hello, world!\n");
4     return 0;
5 }
6
```

```
alexandr@alexandr-Magicbook16:~/Lab$ gedit lab8.c
alexandr@alexandr-Magicbook16:~/Lab$ gcc lab8.c
alexandr@alexandr-Magicbook16:~/Lab$ ./a.out
Hello, world!
alexandr@alexandr-Magicbook16:~/Lab$
```

9. **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 ЭВМ, существенном участии преподавателя и других лиц в написании и отладке программы.

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

10. **Замечания автора** по существу работы: \_\_\_\_\_

---



---



---

11. **Выводы:** Я узнал тип расширения для файлов с кодом на языке си, научился создавать файлы с кодом на си через терминал, открывать их в текстовом редакторе gedit (этот текстовый редактор очень удобен, так как в нем есть поддержка кода для различных языков программирования), компилировать файл с ключами для вывода ошибок, распознавать и исправлять ошибки, которые выводятся при попытке компиляции программы. Компилятор gcc очень удобно выводит допущенные ошибки с подробным описанием и конкретным местом в коде, где они были допущены. Это позволяет быстро находить ошибки и исправлять их. По моему мнению, сам язык си очень хорош для обучения программированию, потому что он приучает программиста к строгости и точности (в синтаксисе, работе с памятью). Изучение языка си очень поможет мне в дальнейшем обучении программированию.

---



---



---

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

---



---

Подпись студента \_\_\_\_\_