|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | **Министерство науки и высшего образования Российской Федерации**  Калужский филиал  федерального государственного бюджетного  образовательного учреждения высшего образования  ***«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»***  ***(КФ МГТУ им. Н.Э. Баумана)*** |

**ФАКУЛЬТЕТ** ***ИУК «Информатика и управление»***

**КАФЕДРА** \_\_***ИУК4 «Программное обеспечение ЭВМ, информационные технологии»***

**ЛАБОРАТОРНАЯ РАБОТА №4**

**«Основы системного администрирования Linux»**

**ДИСЦИПЛИНА: «Операционные системы»**

|  |  |  |
| --- | --- | --- |
| Выполнил: студент гр. ИУК4-52Б | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ( Карельский М.К. )  (Подпись) |
| Проверил: | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ( Красавин Е.В. )  (Подпись) |
| Дата сдачи (защиты):  Результаты сдачи (защиты): | | |
|  | - Балльная оценка:  - Оценка: | |

Калуга, 2022

**Цель:** приобретение практических навыков по работе с командами для работы с компиляторами для языков программирования С и С++, и для управления учетными записями пользователей, групп пользователей, правами доступа к файлам и каталогам в ОС Linux.

**Задачи:**

1. Получить навыки работы с конвейером в ОС Linux.
2. Получить навыки работы с компиляторами для языков программирования С и С++ в ОС Linux.
3. Получить навыки работы с командами для управления учетными записями пользователей в ОС Linux.
4. Получить навыки работы с командами для работы с группами пользователей в ОС Linux.
5. Получить навыки работы с командами для управления доступом к файлам и каталогам в ОС Linux.

**Задание:**

Продемонстрировать работу команд:

1. Добавления и удаления учетных записей
2. Изменения информации о пользователе
3. Смены пароля
4. Создания групп
5. Работы с группами
6. Изменения прав доступа к файлам и каталогам

Для демонстрации работы команд gcc и g++ написать программу на С/С++ с использованием функций ввода, вывода, использование условных и циклических операторов согласно заданию, указанному в варианте.

**Вариант 8**

Дано натуральное число 𝑁 > 2. Вычислить

**Результат:**

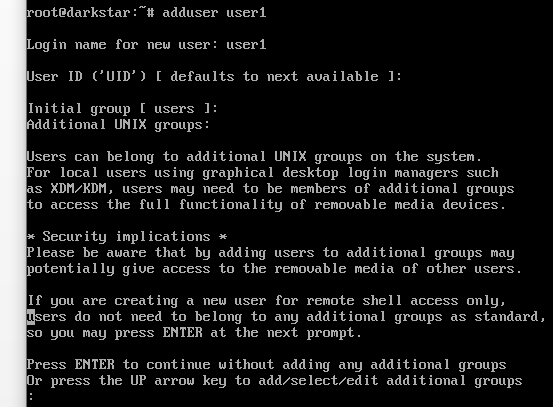


Рис. 1.1. Добавление пользователя

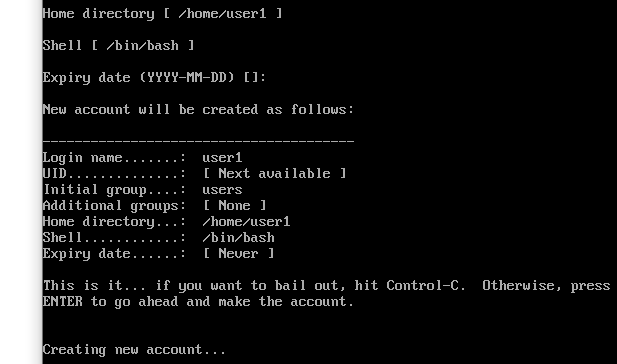


Рис. 1.2. Добавление пользователя

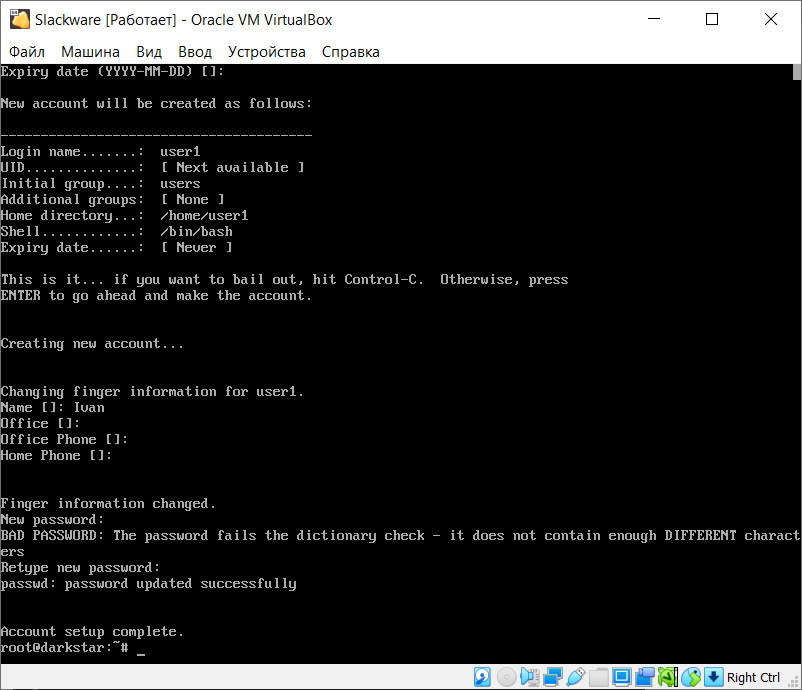


Рис. 1.3. Добавление пользователя

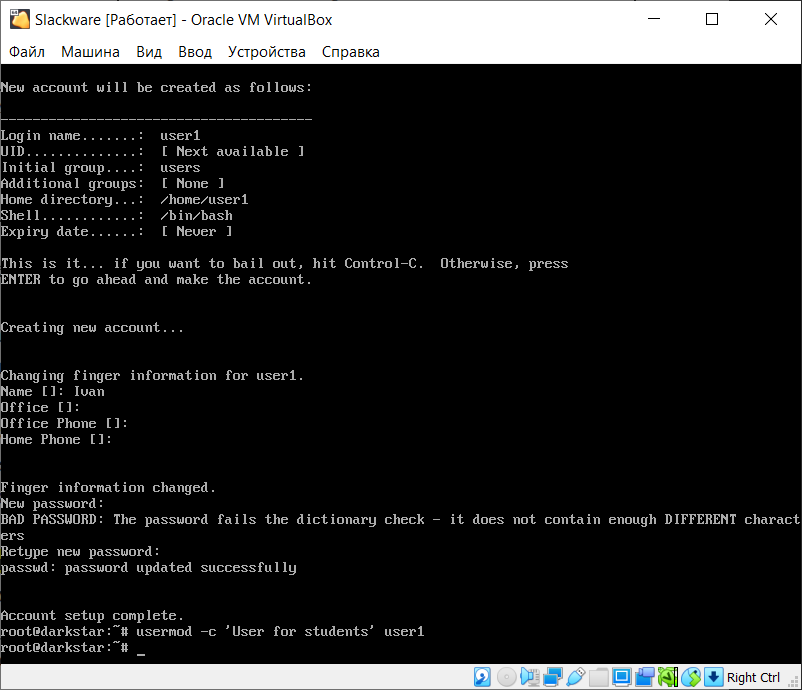


Рис. 2. Изменение информации о пользователе

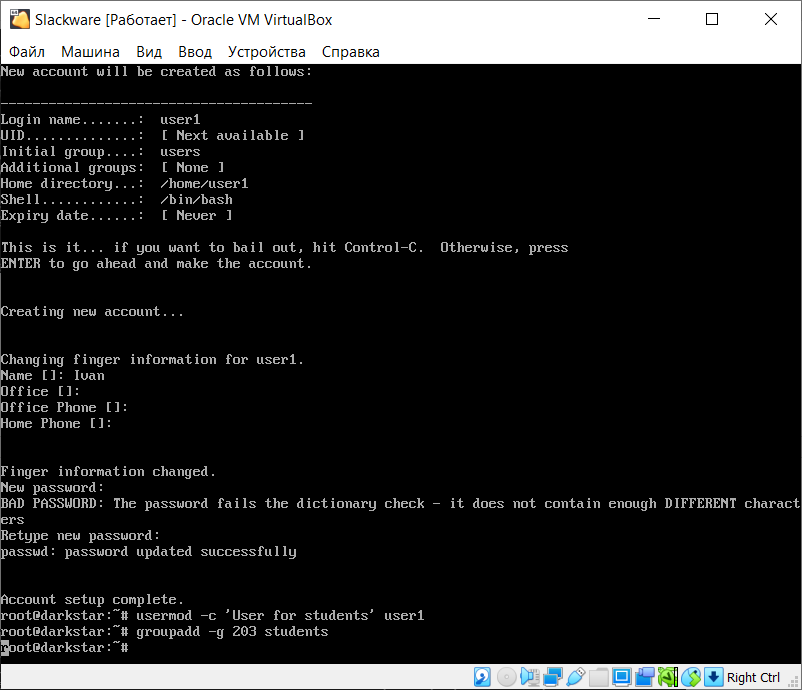


Рис. 3. Создание группы

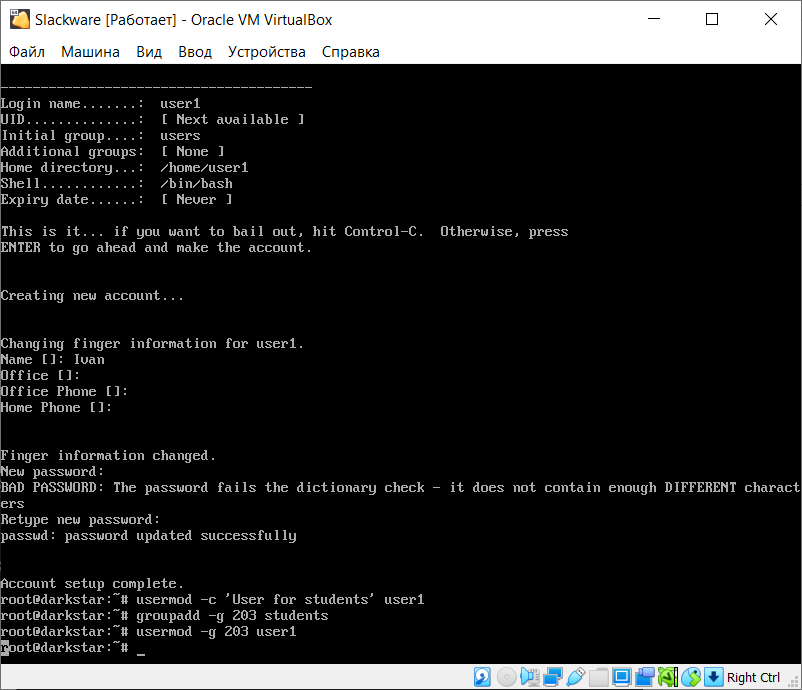


Рис. 4. Добавление в группу пользователя

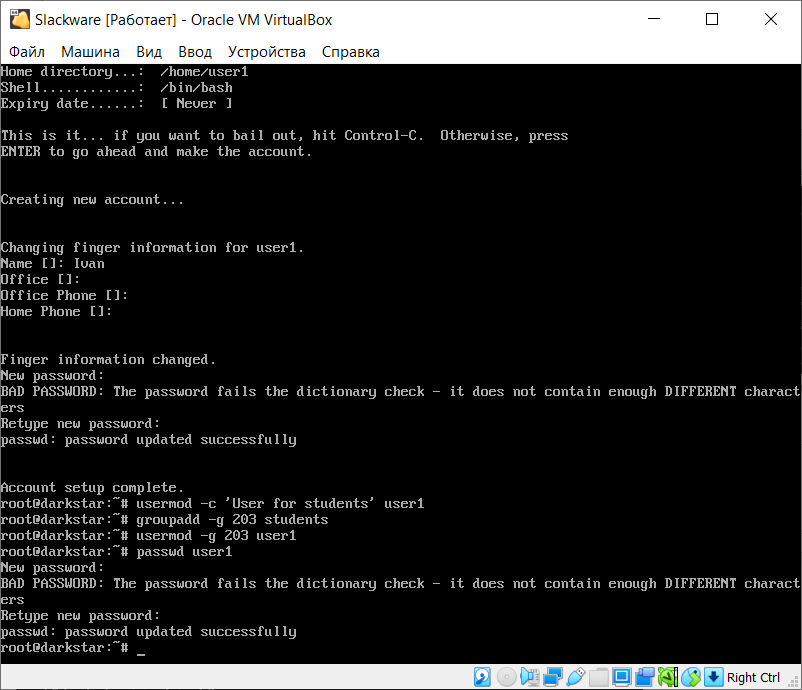


Рис. 5. Смена пароля

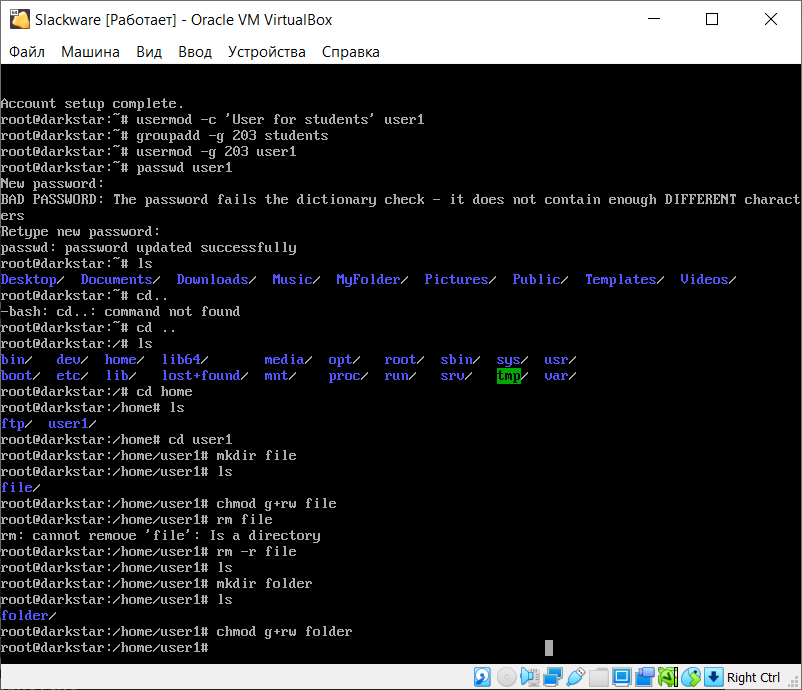


Рис. 6. Изменение прав доступа к папке

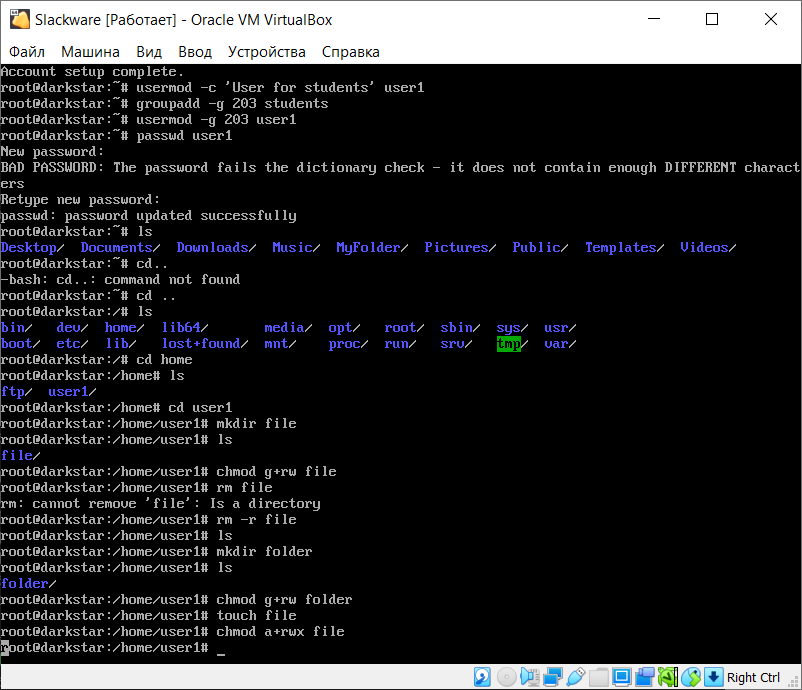


Рис. 7. Изменение прав доступа к файлу

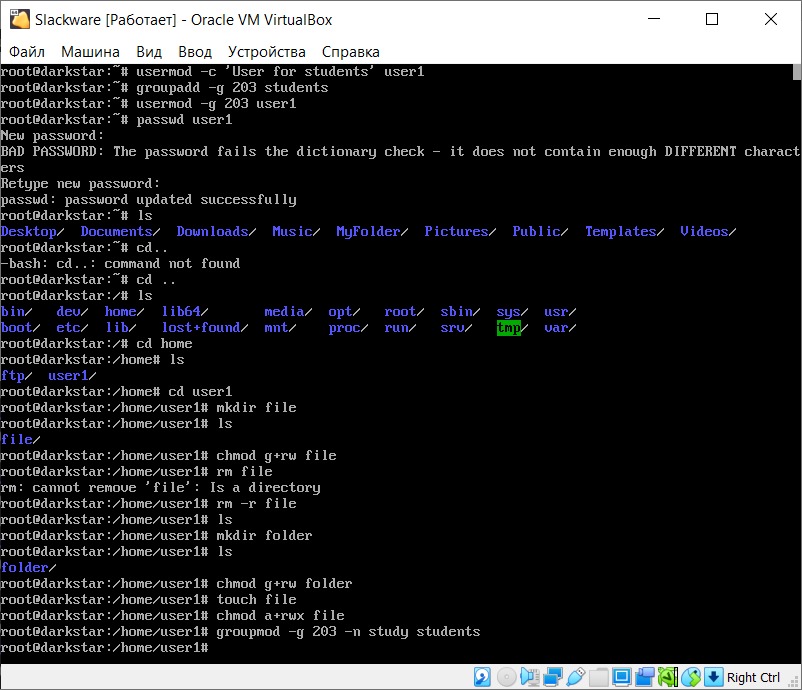


Рис. 8. Переименование группы

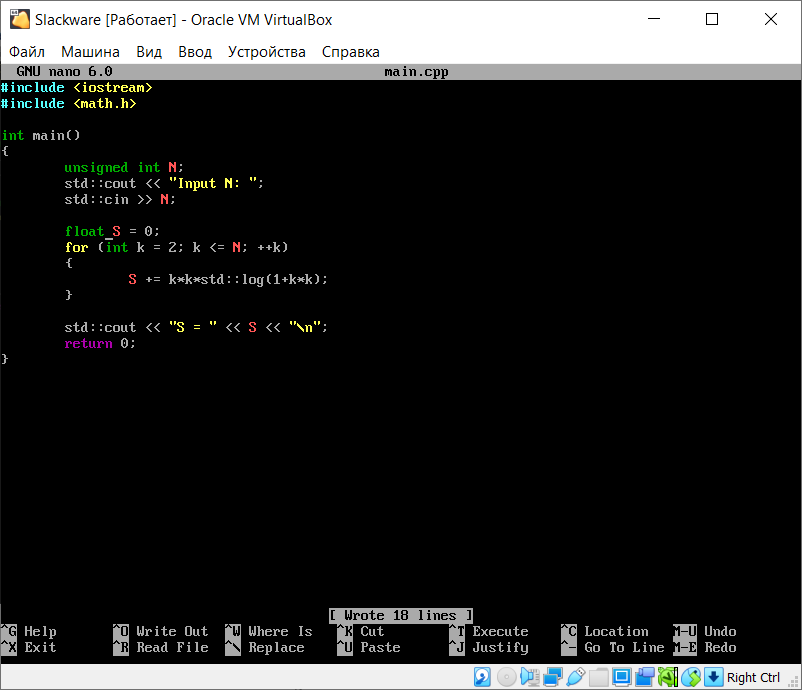


Рис. 9. Код программы

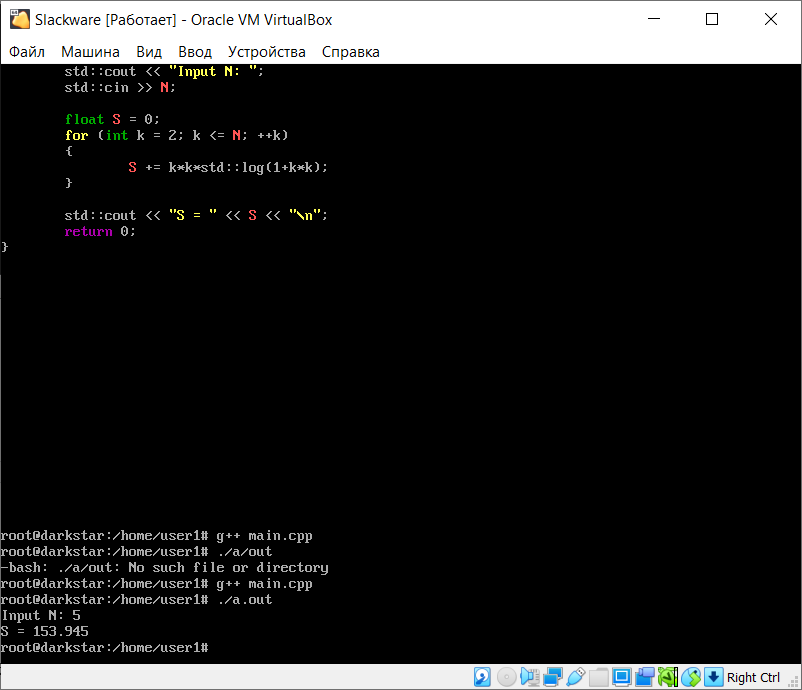


Рис. 10. Компиляция

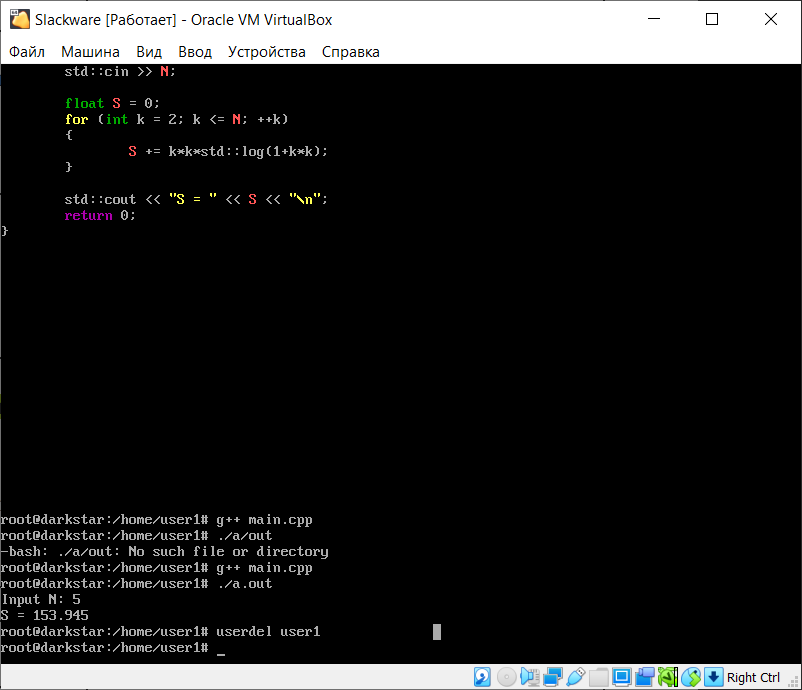


Рис. 11. Удаление пользователя

**Вывод:** в ходе выполнения лабораторной работы были получены практические навыки по работе с командами для работы с компиляторами для языков программирования С и С++, и для управления учетными записями пользователей, групп пользователей, правами доступа к файлам и каталогам в ОС Linux.

**Контрольные вопросы:**

1. **Раскройте понятие компиляции.**

Компиляция — сборка программы, включающая трансляцию всех модулей программы, написанных на одном или нескольких исходных языках программирования высокого уровня и/или языке ассемблера, в эквивалентные программные модули на низкоуровневом языке, близком машинному коду (абсолютный код, объектный модуль, иногда на язык ассемблера) или непосредственно на машинном языке или ином двоичнокодовом низкоуровневом командном языке и последующую сборку исполняемой машинной программы.

1. **Опишите структуру компилятора.**

Процесс компиляции состоит из следующих этапов:

1. трансляция программы — трансляция всех или только изменённых модулей исходной программы.
2. компоновка машинно-ориентированной программы.
3. **Назовите этапы трансляции программы.**

Трансляция программы как неотъемлемая составляющая компиляции включает в себя:

* Лексический анализ. На этом этапе последовательность символов исходного файла преобразуется в последовательность лексем.
* Синтаксический (грамматический) анализ. Последовательность лексем преобразуется в дерево разбора.
* Семантический анализ. Дерево разбора обрабатывается с целью установления его семантики (смысла) — например, привязка идентификаторов к их декларациям, типам, проверка совместимости, определение типов выражений и т. д. Результат обычно называется «промежуточным представлением/кодом», и может быть дополненным деревом разбора, новым деревом, абстрактным набором команд или чем-то ещё, удобным для дальнейшей обработки.
* Оптимизация. Выполняется удаление излишних конструкций и упрощение кода с сохранением его смысла. Оптимизация может быть на разных уровнях и этапах — например, над промежуточным кодом или над конечным машинным кодом.
* Генерация кода. Из промежуточного представления порождается код на целевом машинно-ориентированном языке.

1. **Опишите понятие конвейера.**

Конвейер представляет собой механизм межпроцессного взаимодействия с использованием передачи сообщений. Он позволяет объединить команды в цепочку и использовать вывод одной команды в качестве ввода другой.

1. **Приведите пример использования конвейера.**

# команда1 | команда2

$ cp \*.c test | tar cf t.tar test | gzip t.tar | ls

1. **Опишите понятие псевдонимов (alias).**

Псевдоним (alias) представляет собой сокращение для ввода более длинной команды.

1. **Перечислите компоненты компилятора gcc.**

* препроцессор
* компилятор
* ассемблер
* редактор связей

1. **Приведите пример команды для компиляции программы.**

$ gcc main.с io.c -о bookrecs

1. **Перечислите и опишите опции компилятора gcc.**

|  |  |
| --- | --- |
| **Опция** | **Назначение** |
| -S | Выводит исключительно код на языке ассемблера. Версии кода на языке ассемблера для скомпилированных файлов имеют расширение .s. В примере генерируется файл greet.s |
| -P | Выводит результаты работы препроцессора |
| -c | Создает исключительно файл объектного кода.  Версии скомпилированных файлов объектного кода имеют расширение .о |
| -g | Осуществляет подготовку скомпилированной программы для использования совместно с символическим отладчиком |
| -o  имя\_файла | Задает имя исполняемого файла, *имя\_файла.* По умолчанию используется a.out |
| -O | Выполняет оптимизированную компиляцию |
| -l имя\_файла | Обеспечивает применение для компоновки программы системной библиотеки с указанным именем файла. Имя файла содержит префикс lib и расширение  .a. В командной строке gcc эта опция не указывается. Опции -l должны всегда располагаться после исходного кода и имен файлов объектного кода в командной строке |
| -Idir | Задает каталоги, в которых производится поиск файлов для включения, таких как файлы заголовков (.h) |
| -Ldir | Задает каталоги, в которых производится поиск библиотек |

1. **Назовите форматы, которые могут использоваться при создании двоичных файлов.**

ELF (Executable and Linking Format - формат компоновки и исполнения) и a.out.

1. **Назовите особенность компилятора gcc при использовании его для языка C++.**

Утилита gcc выполняет также функции компилятора языка C++. Она может считывать и компилировать любую программу на языке C++. Однако она не позволяет выполнять автоматическую компоновку с вызовом библиотеки классов C++. Эта библиотека должна вызываться отдельно, с использованием режима командной строки. Для этого можно также воспользоваться командой g++, в результате чего вызывается компилятор gcc вместе с библиотекой классов C++.

1. **Перечислите данные о пользователе, которые хранятся в системе.**

* имя пользователя — регистрационное имя пользователя;
* пароль — зашифрованный пароль учетной записи пользователя;
* идентификатор пользователя — уникальный номер, назначенный системой;
* идентификатор группы — номер, служащий для обозначения группы, к которой относится пользователь;
* комментарий — информация о пользователе, например его имя и фамилия;
* начальный каталог — начальный каталог пользователя;
* регистрационный shell — shell, запускаемый при регистрации пользователя в системе обычно /bin/bash.

1. **Опишите понятие системного имени (user name).**

Регистрационное имя пользователя.

1. **Опишите понятие идентификатора пользователя (UID).**

Уникальный номер, назначенный системой.

1. **Опишите понятие идентификатора группы (GID).**

Номер, служащий для обозначения группы, к которой относится пользователь.

1. **Опишите понятие полного имени (full name).**

Полное имя пользователя.

1. **Опишите понятие домашнего каталога (home directory).**

Начальный каталог пользователя.

1. **Опишите понятие начальной оболочки (login shell).**

Начальная оболочка (интерпретатор командной строки) (login shell) запускается при входе пользователя в систему в текстовом режиме (например, на виртуальной консоли). Она подтягивает опции и переменные окружения из конфигурационных файлов, все последующие процессы запускаются в контексте этого шелла.

1. **Перечислите права для каталогов.**

* право на чтение — просмотр содержимого каталога;
* право на запись — создание и удаление файлов данного каталога;
* право на выполнение — возможность входа в каталог и поиска файлов в нем.

1. **Назовите команду, позволяющую посмотреть, какие пользователи работают в данный момент в системе.**

who (и можно воспользоваться опцией -u для получения дополнительной информации о пользователе: с какого терминала подключен, как долго неактивен)

1. **Предложите вариант команды для добавления пользователя.**

useradd рег\_имя

1. **Приведите пример записи, хранящейся в файле /etc/passwd.**

root:х:0:0:test,w,1,w:/root:/bin/bash

richlp:YOTPd3Pyy9hAc:500:500:CalderaDesktopUser:/home/richp:/bin/bash

mark:\*:501:501:CalderaDesktopUser:/home/mark:/bin/bash

1. **Опишите структуру записи, хранящейся в файле /etc/passwd.**

* Имя пользователя — регистрационное имя пользователя;
* пароль — зашифрованный пароль учетной записи пользователя;
* идентификатор пользователя — уникальный номер, назначенный системой;
* идентификатор группы — номер, служащий для обозначения группы, к которой относится пользователь;
* комментарий — информация о пользователе, например его имя и фамилия;
* начальный каталог— начальный каталог пользователя;
* регистрационный shell — shell, запускаемый при регистрации пользователя в системе обычно /bin/bash

1. **Назовите команды для работы с учетными записями.**

useradd, usermod и userdel.

1. **Назовите команды для работы с группами пользователей.**

groupadd, groupmod и groupdel.

1. **Предложите вариант команды, предоставляющей право на выполнение и отменяющей право на запись для файла file.**

$ chmod +x-w file