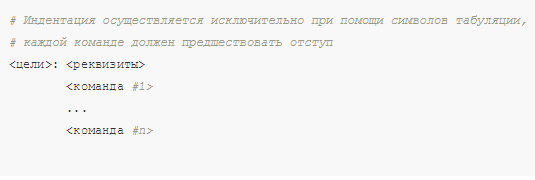
Практика 4.

Сборка программы с помощью GNU Make

Make — утилита предназначенная для автоматизации преобразования файлов из одной формы в другую. Правила преобразования задаются в скрипте с именем Makefile, который должен находиться в корне рабочей директории проекта.

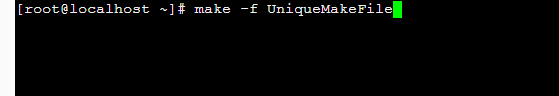
В общем виде, синтаксис файла можно представить так:



Запуск утилиты происходит посредством выполнения команды { make + имя\_файла } в консоли, в случае если имя файла задано стандартно,

(прим. `GNUmakefile', `makefile' и `Makefile' ). Файл make должен находится в корневой папке проекта.

Если имя make-файла задано нестандартно, то для его выполнения требуется добавить атрибут ‘ –f ’ или ‘- - file’.



Make открывает make-файл, считывает правила и выполняет команды, необходимые для создания указанной *цели*.

Стандартные цели для сборки дистрибутивов GNU:

all — выполнить сборку пакета;

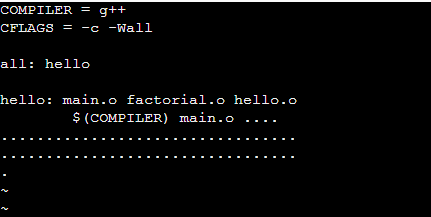
install — установить пакет из дистрибутива (производит копирование исполняемых файлов, библиотек и документации в системные каталоги);

uninstall — удалить пакет (производит удаление исполняемых файлов и библиотек из системных каталогов);

clean — очистить дистрибутив (удалить из дистрибутива объектные и исполняемые файлы, созданные в процессе компиляции);

distclean — очистить все созданные при компиляции файлы и все вспомогательные файлы, созданные утилитой ‘./configure’ в процессе настройки параметров компиляции дистрибутива.

По умолчанию make использует самую первую цель в make-файле. Так же, в файлах make имеется возможность использования переменных. Для их внедрения надо просто присвоить им значение до момента их использования. После этого можно подставлять их значение в нужное место вот таким способом: $(Имя переменной).



Значением может являться произвольная последовательность символов, включая пробелы и обращения к значениям других переменных. С учётом сказанного, можно модифицировать наш make-файл следующим образом.

Создание комментариев производится с помощью символов ‘ # ’, что сигнализирует терминалу об игнорировании данной строки. Последующие комментарии на следующих строках можно продолжить набрав ‘/’ в начале каждой строки.

*Пример:*

Имеется четыре файла :

1. main.cpp
2. hello.cpp
3. factorial.cpp
4. functions.h

Компиляция прямым способом: g++ main.cpp hello.cpp factorial.cpp -o hello

Компиляция с помощью makefile:



*Задание:*

1. С помощью утилиты Git, склонируйте проект с <https://github.com/arizorin/Practice4-MAKEFILE-.git> Разберите устройство каждого makefile’a.
2. Создайте свой многофайловый проект (Калькулятор и тд.) на удобном для вас языке программирования. Создайте к нему makefile и поместите на ваш репозиторий на GitHub’e/

Список литературы:

1. [Как создавать, собирать, устанавливать и использовать пакеты с программами и библиотеками для UNIX-подобных систем](https://habrahabr.ru/post/321468/)
2. [Managing Projects with GNU Make, Third Edition](http://shop.oreilly.com/product/9780596006105.do)
3. <http://rus-linux.net/nlib.php?name=/MyLDP/algol/gnu_make/gnu_make_3-79_russian_manual.html>