РГПУ имени А.И. Герцена

Институт Компьютерных Наук и Технологического Образования

Информатика и вычислительная техника

Работу выполнил Н.Ю. Бахурев

**Лабораторная работа №2.**

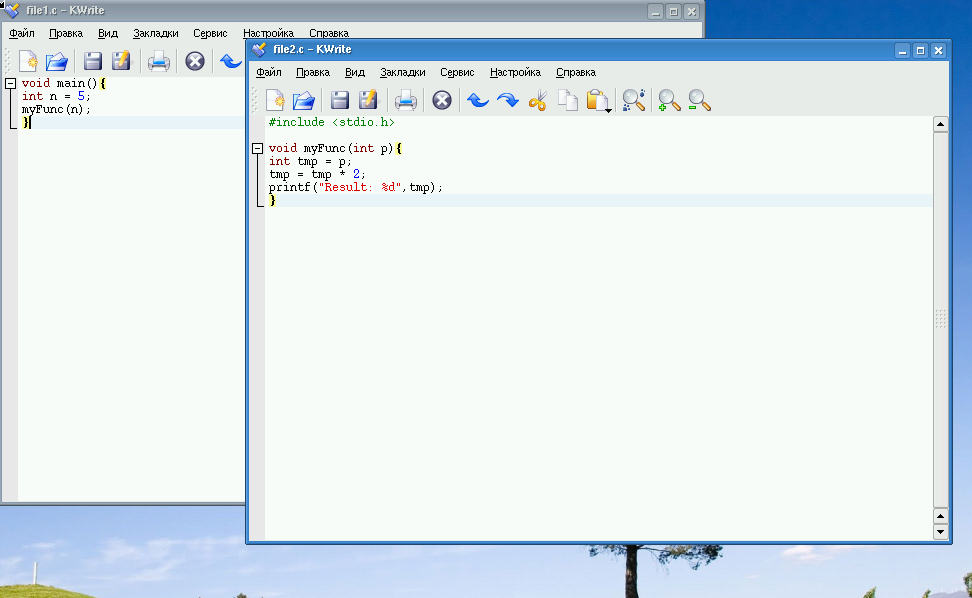
**Практическое знакомство со стандартной утилитой GNU make для построения проектов в ОС UNIX**

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

**Задание:** Изучить особенности работы с утилитой make при создании проекта на языке С (С++) в ОС Unix, а также получить практические навыки использования утилиты GNU make при создании и сборке проекта.

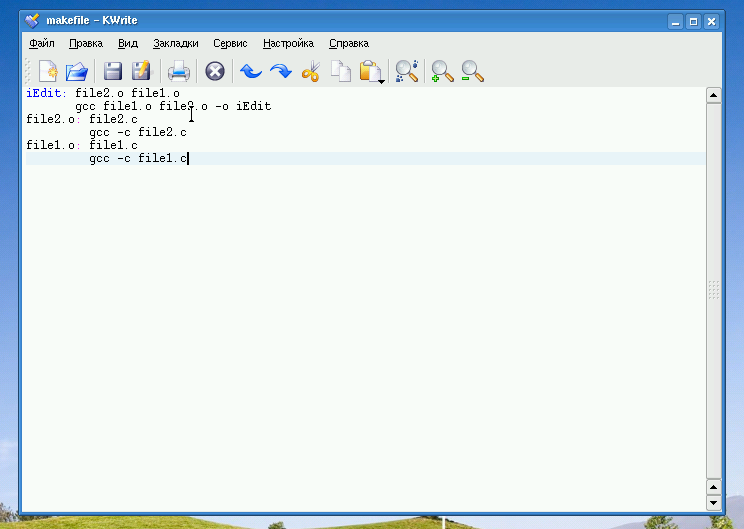
**Задание 1.**

Используя любой текстовый редактор, создать простейшую программу на языке C (C++) с использованием, как минимум, двух исходных файлов (с программным кодом).



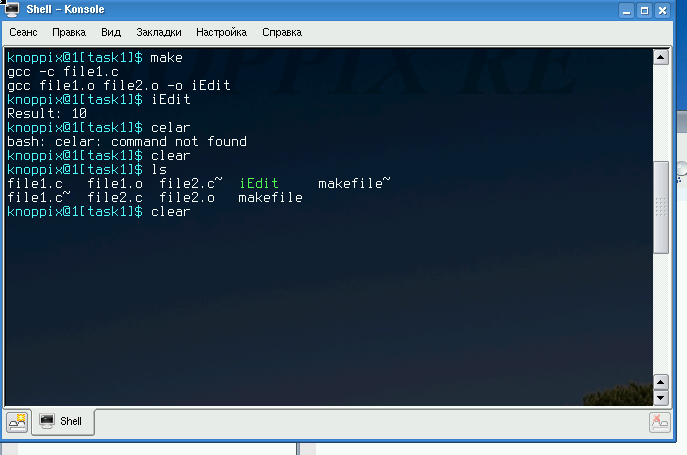
**Задание 2.**

Для автоматизации сборки проекта утилитой make создать make-файл.



**Задание 3.**

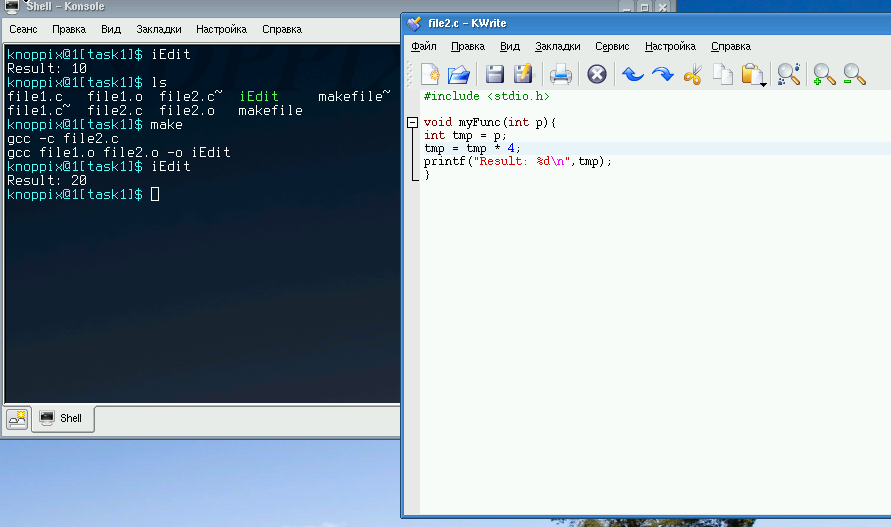
Выполнить программу (скомпилировать, при необходимости отладить).



**Задание 4.**

Показать, что при изменении одного исходного файла и последующем вызове make будут исполнены только необходимые команды компиляции.

При изменении одного файла и последующей сборки всей программы, происходит перекомпиляция только одного файла, в нашем случае file2.c

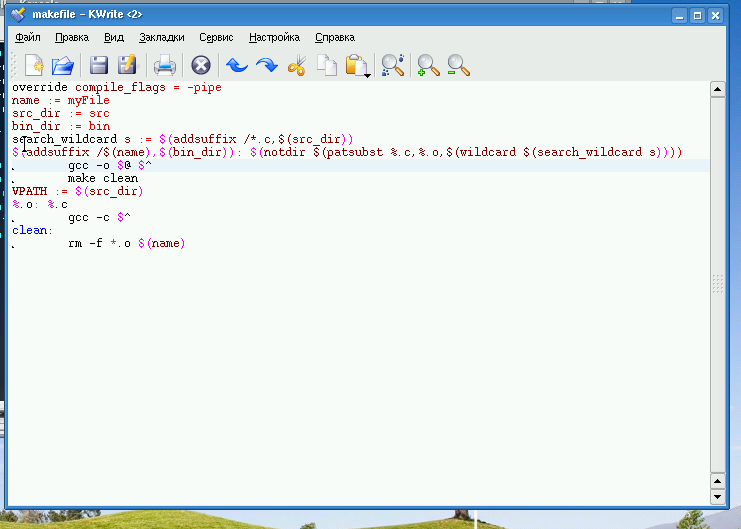


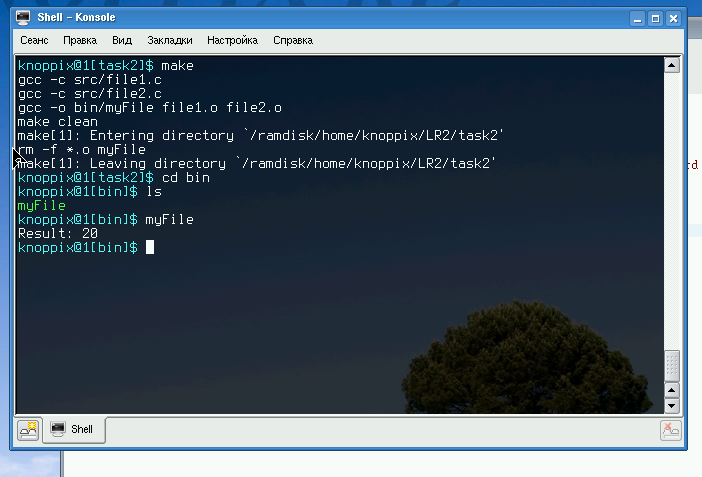
**Задание 5.**

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 (С++) и цели в make-файле должны формироваться динамически;

наличие цели clean, удалающей временные файлы





Make-файл был значительно автоматизирован и стал более универсален. Здесь использованы функция wildcard, которая получает список файлов с заданным шаблоном в выбранном каталоге, и функция patsubst, которая заменяет заданную подстроку в заданной строке. Эти функции позволяют автоматически построить список объектных файлов программы. Еще в данном Make-файле автоматизирован поиск исходных файлов по нескольким директорием с использованием переменных search \_wildcard s и VPATH.

**Выводы**

В ходе данной лабораторной работы была изучена утилита Make, которая позволяет собирать программу из множества разрозненных файлов. Данная утилита имеет большое количество возможностей по автоматизации сборки проекта и позволяет создать такой Make-файл, который подойдет ко многим программам и при сборке каждой из них потребует незначительное количество изменений. Была использована сборка knoppix и Oracle VirtualBox.

2017