МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение высшего образования

«Казанский национальный исследовательский технический университет им. А.Н. Туполева-КАИ»

(КНИТУ-КАИ)

Институт компьютерных технологий и защиты информации

(наименование института (факультета), филиала)

Кафедра Прикладной Математики и Информатики

(наименование кафедры)

09.03.04 «Программная инженерия»

(шифр и наименование направления подготовки (специальности))

Лабораторная работа № 8 (Часть 2)

По дисциплине: Операционные системы

На тему: "Разработка простейших приложений "

Выполнил: студент группы 4310

**Гауиш М.Г**.

Проверил: доц. Урахчиский И. Н

Казань, 2023 год

Цель работы

1. Изучение пользовательского интерфейса текстовых процессоров Unix

2. Изучение средств разработки простейших программ и управления процессами

3. Получение практических навыков разработки программ

Краткие теоретические сведения

В системе Unix есть несколько текстовых редакторов. Самый мощный из них - vi (visual interpreter) Его вызов осуществляется по команде: vi имя-файла Если файл существует, то откроется его первая страница, иначе создастся новый файл и откроется пустая страница. Управление редактором осуществляется с помощью односимвольных команд и их комбинаций. В зависимости от команды редактор находится в одном из 4 режимов:

1) экранном - редактирование информации (вводимые команды редактирования не отображаются на экране)

2) строчном - вводятся команды работы с файлами и временного выхода в shell

3) ввода данных

4) команд shell

Основные команды vi :

h Перемещает курсор на один символ влево

j Перемещает курсор вниз на одну строку

k Перемещает курсор вверх на одну строку

l Перемещает курсор на один символ вправо

w Перемещает курсор к началу следующего слова

b Перемещает курсор к первому символу предыдущего слова

e Перемещает курсор на последний символ в следующем слове

nG Перемещает курсор на начало строки с номером n

{ Переместить курсор в начало текущего параграфа, который ограничивается пустой строкой

} Переместить курсор в начало следующего параграфа

( Перемещает курсор в начало текущего предложения

) Перемещает курсор в начало следующего предложения

% Перемещает курсор к парной скобке

H Переместить курсор к первой строке экрана

M Переместить курсор к средней строке экрана

L Переместить курсор к последней строке экрана

ctrl + u экран на одну страницу вверх

ctrl + d экран на одну страницу вниз

ctrl + f экран вперед

ctrl + b экран назад

$ Переместить курсор на последний символ в строке

0(ноль) Переместить курсор на первый символ в строке

a Создать текст после курсора

A Создать текст в конце текущей строки i Создать текст перед курсором

I Создать текст перед первым ненулевым символом в текущей строке o Создать текст с начала новой строки, расположенной под текущей строкой

O Создать текст с начала новой строки, расположенной над текущей строкой

r Заменить текущий символ (rx - заменить один символ символом "x")

R Заменить те символы, которые введены в режиме ESCAPE (RтекстESC)

x Удалить символ

dw Удалить слово над курсором и следующий за ним пробел, или знак пунктуации 3dw удалить три слова

dd Удалить текущую строку 3dd удалить три строки

u Отменить последнюю команду

cw Заменить слово или оставшиеся символы в слове на новый текст

c2w Заменить два слова /текстESC поиск "текст"

:wCR сохранить изменения в файле

:wqCR сохранить и выйти

:qCR выйти без сохранения

:w имя-файлаCR сохранить в указанном файле

:eCR начать редактирование заново

:e имя-файлаCR начать редактирование указанного файла

:! команда-shellCR выполнить команду shell

**Пример 1:**

Выполнить команду: vi convert.c

Ввести с помощью редактора vi простую команду на языке C(файл convert.c)

#include <stdio.h>

/\* Print a list of convetions from gallons to liters \*/

main(){

int low,high,incr; float gals,ltrs;

low = 1;

high = 20;

incr = 1;

gals = low;

while (gals <= high){

ltrs = (gals \* 3.785)

printf ("%4.0f %6.2f\n, gals, ltrs);

gals++;

}

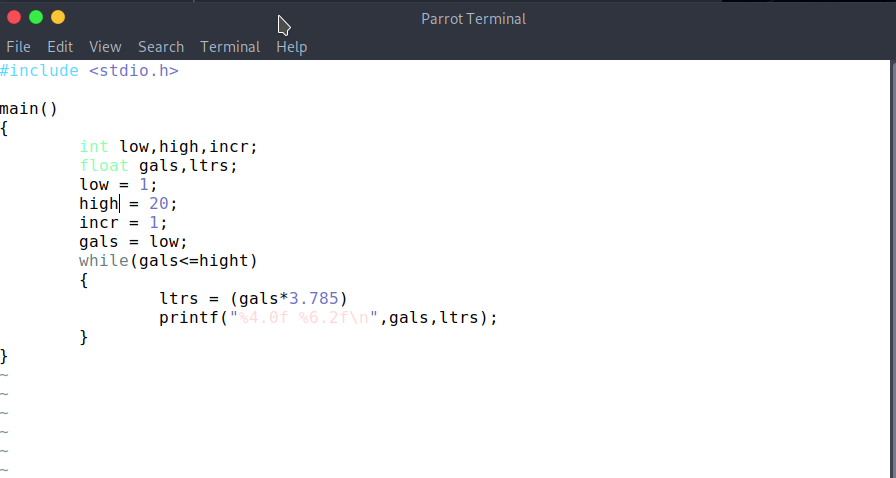
}

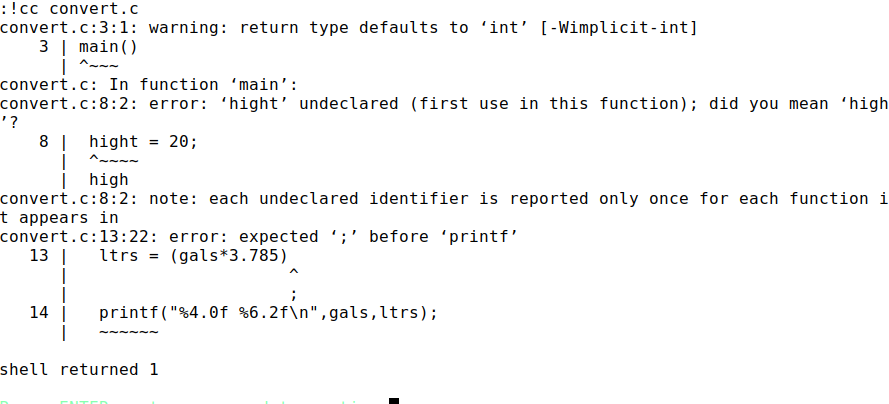
По завершению сохраняем файл (командой :w!)

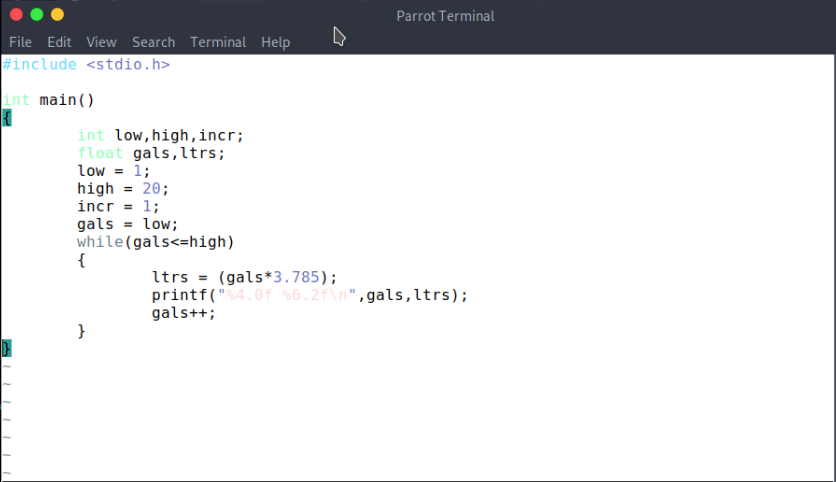
Обязательно сделать ошибки при наборе. Не выходя из редактора, вызвать

компилятор C:

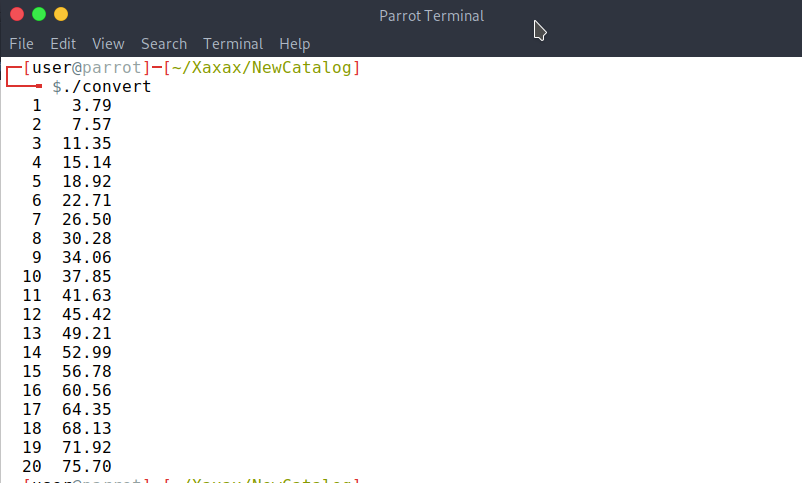
:!cc convert.c

**Результат:**



Исправить допущенные ошибки.

По умолчанию компилятор помещает созданный выполнимый модуль в файл a.out. Для создания другого выполнимого модуля в файле с заданным именем в командной строке вызова компилятора необходимо задать режим -o и имя файла :!cc -o convert convert.c



**Пример 2:**

Набрать бесконечную программу :

#include <stdio.h>

main(){

int num, sum;

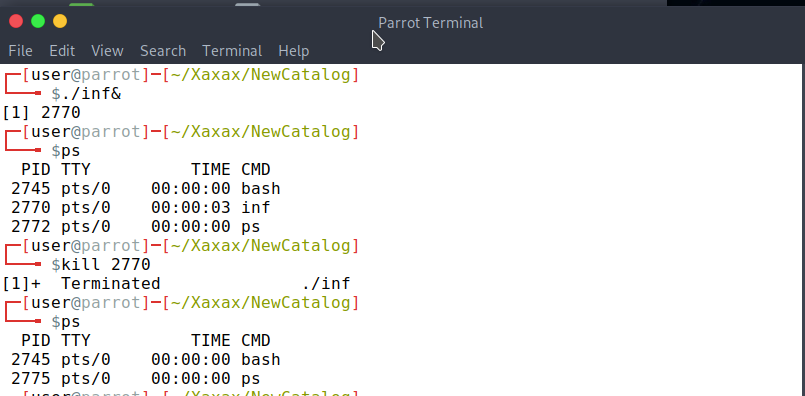
num = 1;

while (num<5){

sum = num +1;

}

}

Выйдя из редактора, откомпилировать и запустить программу в фоновом режиме: $имя-модуля& После запуска программы shell выдаст на экран номер запущенного процесса (PID). Для управления процессами в Unix используются команды ps (process status) для вывода списка активных процессов и kill для уничтожения процесса по его PID. Набрать команды: $ps $kill n $ps

**Вариант задания:**

