УО «Белорусский Государственный Технологический Университет»

**ФИТ**

Отчет по Лабораторной работе №11 на тему «Сигналы»

Выполнил студент

8-й группы:

Куницкий Н. Д.

Проверила:

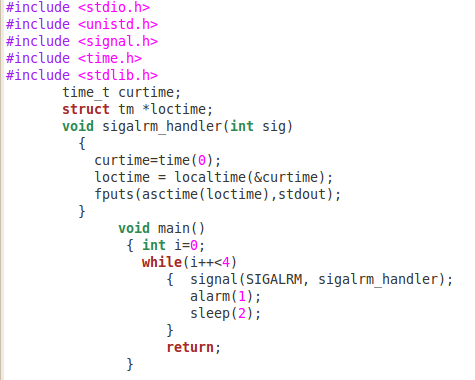
Герман Ю. О

Минск 2016

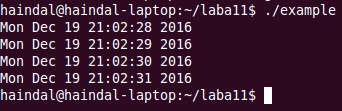
**Цель**. Закрепить на практике команды и типы сигналов Linux.

**Краткие теоретические сведения.**

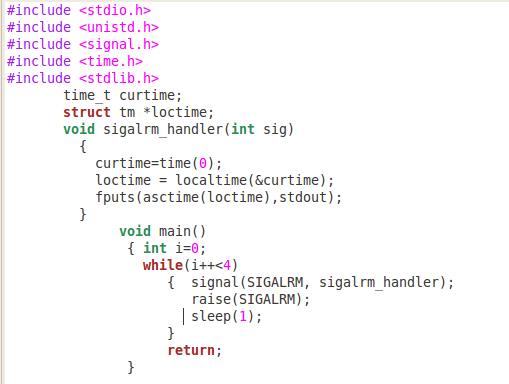
Вывод времени каждую секунду:



Результат работы программы демонстрирует следующий скриншот:



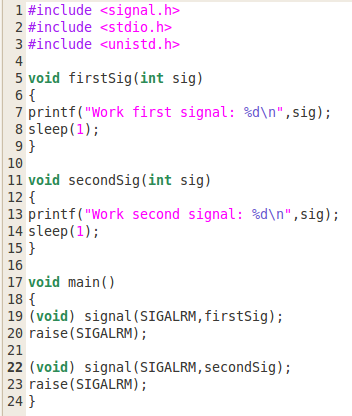
Укажем более предпочтительный вариант программы (более универсальный, подходящий под различные типы сигналов).



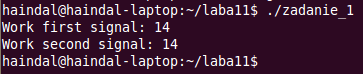
Здесь использована функция raise(). Эта функция «зажигает сигнал». Отметим, что функция signal просто устанавливает связь с обработчиком сигнала, но не активизирует его (легко проверить закомментировав вызов raise).

**ЗАДАНИЕ №1**. Создайте два обработчика сигнала SIGALRM.

**ЛИСТИНГ:**



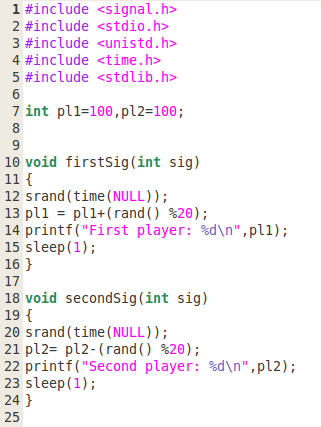
**ВЫПОЛНЕНИЕ:**

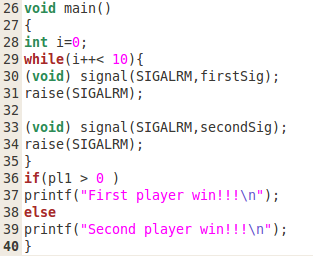
****

**ЗАДАНИЕ №2**. Измените последнюю программу так, чтобы она выдавала не время и дату, а счет в игре двух человек. Первый обработчик играет за первого человека (условно говоря). Второй обработчик играет за второго человека. Первый обработчик добавляет случайное число к общей переменной, второй – отнимает. Играют до 10 раз. Если в конце игры на счету будет число больше нуля – то выиграл первый игрок, иначе выиграл второй игрок. Случайное число разыгрывается таким образом:

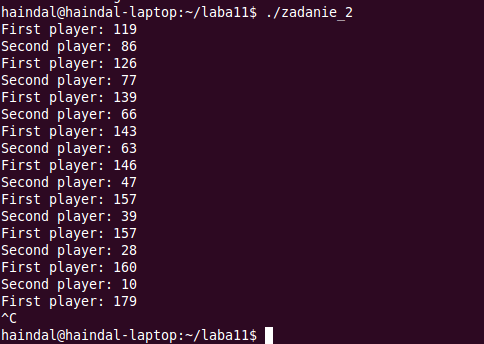
Для получения случайных чисел можно использовать устройство-генератор случайных чисел Linux или библиотечные функции. Последняя возможность иллюстрируется следующей программой:

**ЛИСТИНГ:**

****

****

**ВЫПОЛНЕНИЕ:**

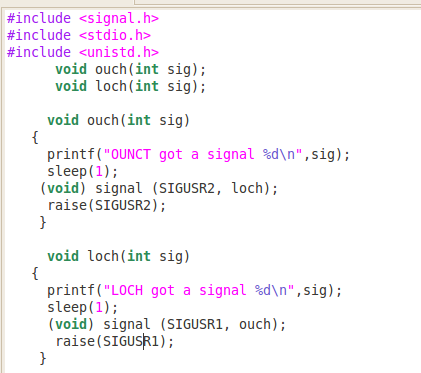
****

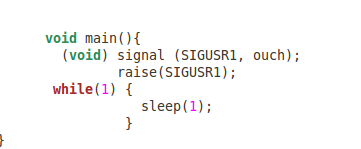
Функция srand устанавливает начальное состояние генератора случайных чисел. Собственно случайные числа в диапазоне от 1 до 100 получаются в команде player[i] =rand() % 100;

Функция возвращает случайное число от 1 до 100. Просто rand() возвращает достаточно большое случайное число, ограниченно системной константой.

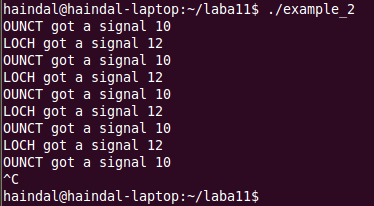
Теперь познакомимся с сигналами пользователя – SIGUSR1 и SIGUSR2.

Рассмотрим следующий пример





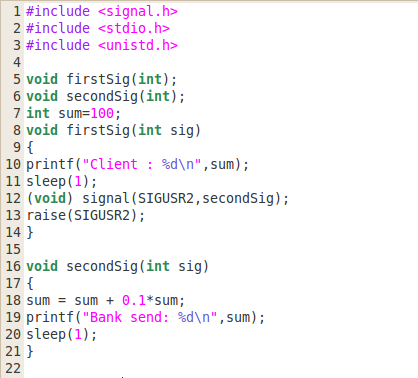
Отметим, что команда (void) signal (…) не посылает сигнал, а связывает сигнал с обработчиком. Сигнал зажигается по команде raise(…). В нашем примере использованы два сигнала пользователя и два обработчика. Работу их легко понять из порождаемого скриншота:

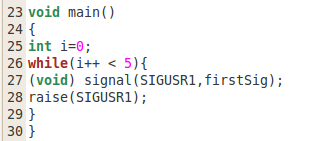


Происходит обмен сигналами между функциями с поочередным выводом на экран.

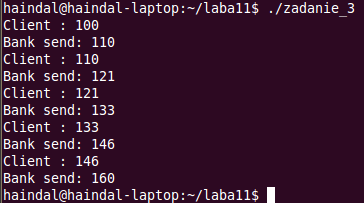
**ЗАДАНИЕ №3**. Представим, что есть клиент банка, который справляется о состоянии своего счета. Клиент посылает запрос (сигнал), банк возвращает ответ (сумма). Допустим, что сумма постепенно растет с каждым вызовом по формуле X= X+0.1\*X.

**ЛИСТИНГ:**

****

****

**ВЫПОЛНЕНИЕ:**

****

**Вывод :** Закрепил на практике команды и типы сигналов Linux.