|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ **Информатика и системы управления**

КАФЕДРА **«Программное обеспечение ЭВМ и информационные технологии» (ИУ7)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.03.04 Программная инженерия**

**Отчет**

|  |  |
| --- | --- |
| **по лабораторной работе №** | 1 |

**Название:** Знакомство с Unix

**Дисциплина:** Операционные системы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ7-52Б |  |  | Е.В. Брянская |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | Н.Ю. Рязанова |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

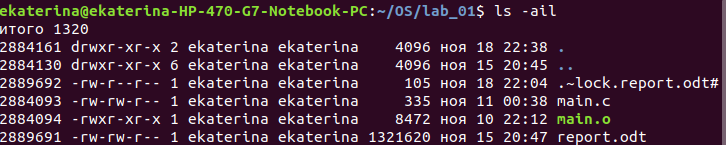
Москва, 2020

**Задание 1.**

Сделать hard link.

Жёсткая ссылка (hard link) — другое название файла.

До создания:



1 столбик — номера inode

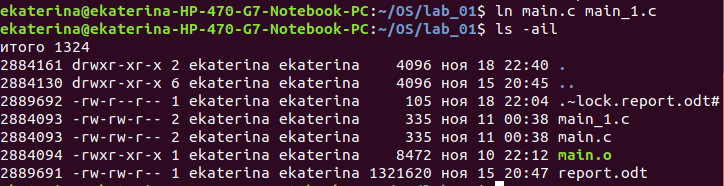
2 столбик —

а) 1 буква — тип файла (d — директория, «-» - обычный файл)

б) права доступа (первые три для user, следующие три для group, остальные три для others); r — read, w — write, x — execute

3 столбик — количество жёстких ссылок (hard links) на файл

После:



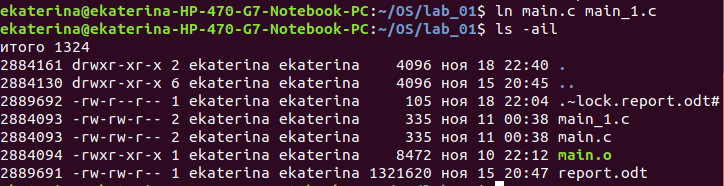
У main\_1.c (hard link) такой же номер inode, что и у main.c. Также увеличилось количество жёстких ссылок на него, как и должно было.

**Задание 2.**

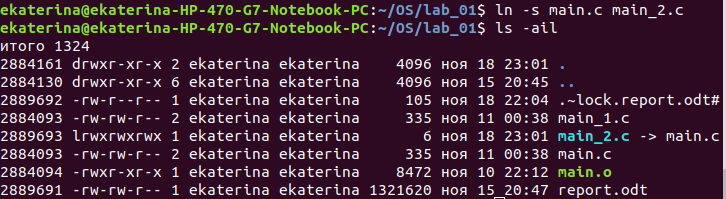
Сделать soft link.

Soft link (мягкая ссылка, символьная строка) хранит путь к файлу.

До создания:



После:



main\_2.c — soft link на main.c. Указан тип файла «l» во втором столбике.

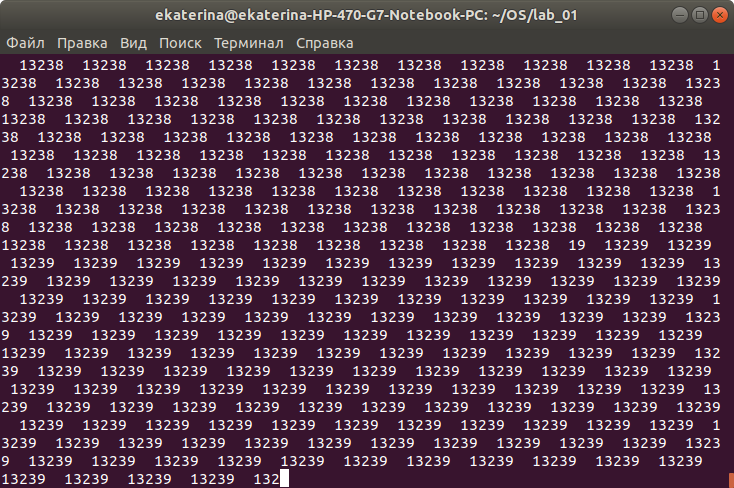
**Задание 3.**

Создать сироту или зомби.

Создание зомби

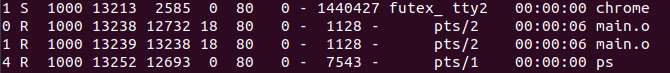
Зомби — процесс, у которого отобраны все ресурсы, кроме последнего, а именно, как говорят, строки в таблице процессов. Нужен для того, чтобы процесс-предок, вызывающий системный вызов wait(), не был заблокирован на нём навсегда, а смог получить статус завершения процесса-потомка.

Запущенная программа:



До создания:

\* \* \*



1 столбик — флаги

а) 0 — был и fork(), и exec(); нормальное выполнение;

б) 1 — был fork(), но не было exec();

в) 4 — процесс выполняется с правами суперпользователя.

2 столбик — статус

а) R — running, процесс выполняется;

б) S — interruptable sleep, прерываемый сон;

в) Z — зомби;

и т. д.

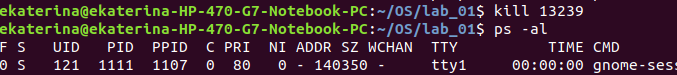
3 столбик — идентификатор пользователя

4 столбик — идентификатор процесса

5 столбик — идентификатор процесса-родителя

У процесса-родителя выставлен флаг 0, а у процесса-потомка 1.

Создание зомби:

\* \* \*

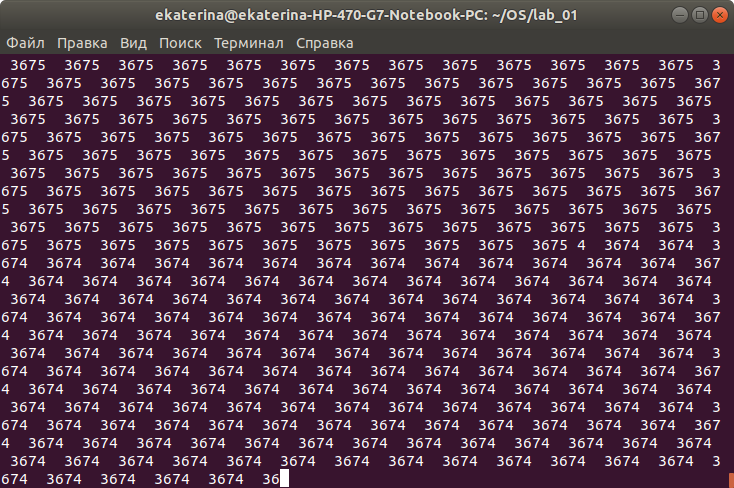


Статус «Z» указывает, что процесс с идентификатором 13239 — зомби.

Создание сироты

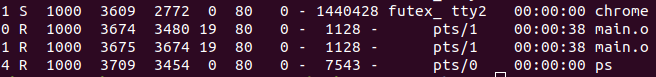
Сирота — процесс-потомок, предок которого завершился раньше него самого. В таком случае, указатель на предка становится недействительным, и запускается процесс усыновления. В результате процесс-потомок получает указатель на нового предка.

Запущенная программа:

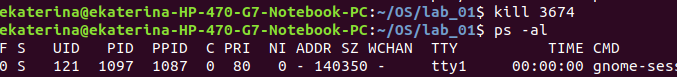


До создания:

\* \* \*



Создание сироты:

\* \* \*



Так как используется Ubuntu, то потомка усыновляет процесс-посредник (в данном случае, его идентификатор 1457).

**Задание 4.**

Передать сообщение через именованный программный канал.

Создание именованного канала типа fifo:



Передача сообщения: (терминал 1)



Получение сообщения: (терминал 2)

