



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
КАФЕДРА

Информатика и системы управления
Информационная безопасность (ИУ8)

Безопасность Систем Баз Данных

Отчет по Лабораторной работе №2 “Знакомство с map-pages”

Выполнил:
Овсебян А.Н.,
студент группы ИУ8-63

Проверил:
Зенькович С. А.,
старший преподаватель
кафедры ИУ8

Оглавление

Цель работы	3
Ход работы.....	3
Теоретическая часть	3
man-pages.....	3
chmod	4
chown	4
fstab	4
proc	5
signal	5
sh	5
Практическая часть	6
stdio	6
stdin/stdout/stderr	6
pipe	7
dup	7
fork	8
exec.....	8
Вывод.....	10

Цель работы

Изучение идеологии man-pages

Ход работы

Теоретическая часть

man-pages

man-pages (от англ. manual — руководство) — справочные страницы, которые предоставляются почти всеми *nix-дистрибутивами, включая Arch Linux. Для их отображения служит команда `man`.

Структура руководства представима следующим образом:

- **NAME.** Содержит название и краткое описание программного продукта
- **SYNOPSIS.** Содержит описание объекта
- **DESCRIPTION.** Полное описание функционирования объекта
- **EXIT STATUS**
- **RETURN VALUE.** Возвращаемые объектом значения
- **OPTIONS.** Описывает ключи для утилиты, если программный продукт таковой является
- **FILES.** Ссылки на используемые конфигурационные файлы
- **ENVIRONMENT.** Используемые переменные окружения
- **VERSIONS**
- **CONFORMING TO**
- **NOTES**
- **AUTHORS**
- **SEE ALSO.** Описание других страниц руководства.
- **BUGS.** Известные неполадки в работе программного продукта
- **EXAMPLES.** Примеры использования

NAME

`man - an interface to the system reference manuals`

SYNOPSIS

```
man [man options] [[section] page ...] ...
man -k [apropos options] regexp ...
man -K [man options] [section] term ...
man -f [whatis options] page ...
man -l [man options] file ...
man -w|-W [man options] page ...
```

DESCRIPTION

chmod

SYNOPSIS

```
chmod [OPTION]... MODE[,MODE]... FILE...
chmod [OPTION]... OCTAL-MODE FILE...
chmod [OPTION]... --reference=RFILE FILE...
```

Согласно man-pages **chmod** изменяет права доступа каждого указанного файла в соответствии с режимом, который может быть представлен либо в символическом виде, либо восьмеричным числом.

chown

chown - изменяет права доступа пользователя и/или группы

SYNOPSIS

```
chown [OPTION]... [OWNER][:[GROUP]] FILE...
chown [OPTION]... --reference=RFILE FILE...
```

- Если указан только владелец (имя пользователя или числовой идентификатор пользователя), этот пользователь становится владельцем каждого данного файла, а группа файлов не изменяется.
- Если за владельцем следует “:” и имя группы (или числовой идентификатор группы) без пробелов между ними, групповое владение файлами также изменяется.
- Если за именем пользователя следует “:,” но не имя группы, этот пользователь становится владельцем файлов, а группа файлов изменяется на группу входа этого пользователя.
- Если даны “:” и группа, но не указан владелец, изменяется только группа файлов; в этом случае **chown** выполняет ту же функцию, что и **chgrp**.
- Если задано только “:” или весь операнд пуст, ни владелец, ни группа не меняются.

fstab

SYNOPSIS

```
/etc/fstab
```

Файл **fstab** содержит описательную информацию о файловых системах, которые система может монтировать.

Полный путь к файлу - **/etc/fstab**. Этот файл подлежит открытию в любом текстовом редакторе, но редактировать его возможно только от имени суперпользователя, т.к. файл является важной, неотъемлемой частью системы, без него система не загрузится.

Следующее является примером содержания файла **fstab**:

```
LABEL=t-home2    /home          ext4           defaults,auto_da_alloc    0    2
```

proc

/proc - это виртуальная файловая система, не занимающая место на диске. Большинство программ получают информацию из файлов в **/proc**, форматируют их своим способом, а затем отображают.

signal

Сигналы — асинхронное уведомление процесса о каком-либо событии, один из основных способов взаимодействия между процессами. Когда сигнал послан процессу, операционная система прерывает выполнение процесса, при этом, если процесс установил собственный обработчик сигнала, операционная система запускает этот обработчик, передав ему информацию о сигнале, если процесс не установил обработчик, то выполняется обработчик по умолчанию.

Узнать список посылаемых системных прерываний можно с помощью команды:
kill -l

sh

Команда **sh** позволяет запустить новый экземпляр оболочки **bash**, дав команду **bash** или **sh**.

При этом можно заставить новый экземпляр оболочки выполнить какой-то скрипт, если передать имя скрипта в виде аргумента команды **bash**

Пример:

```
sh somescript.sh
```

Практическая часть

stdio

stdio (от англ. standard input/output header — стандартный заголовочный файл ввода-вывода) заголовочный файл стандартной библиотеки языка Си, содержащий определения макросов, константы и объявления функций и типов, используемых для различных операций стандартного ввода и вывода.

stdin/stdout/stderr

Стандартные потоки ввода и вывода:

- **STDIN** стандартный ввод программы (то что приходит при вводе с терминала или от браузера клиента)
- **STDOUT** стандартный вывод - то, куда выводят данные команды echo/print, консоль или сокет отправляющий данные браузеру
- **STDERR** поток сообщений об ошибках

Пример:

```
#include <stdio>

int main ()
{
    FILE * f;
    f = fopen("../task.txt", "w");

    if (f != NULL)
    {
        fputs("Hi, my name is Ara", f);
        fclose (f);
    }

    return 0;
}
```

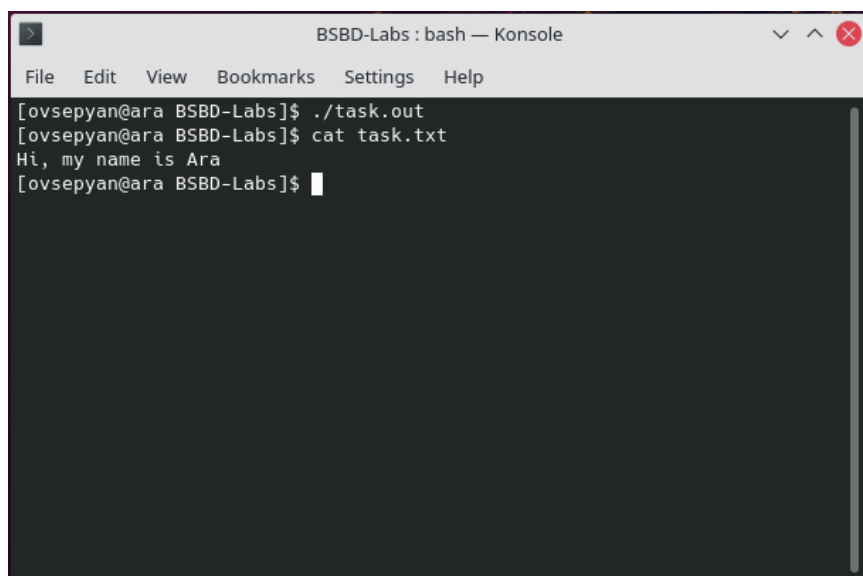


Рисунок 1 - Результат выполнения

pipe

Метод для создания неименованного канала передачи данных между процессами.

dup

Функция **dup()** возвращает новый дескриптор файла, который полностью описывает (т.е. дублирует) состояние файла, связанного с **handle**. В случае успеха возвращается неотрицательная величина, а в противном случае —1.

Функция **dup2()** дублирует **old_handle** как **new_handle**. Если имеется файл, который был связан с **new_handle** до вызова **dup2()**, то он будет закрыт. В случае успеха возвращается 0, а в случае ошибки —1.

Пример:

```
#include<stdio.h>
#include <unistd.h>
#include <fcntl.h>

int main()

{
    int file_desc = open("../task.txt", O_WRONLY | O_APPEND);
    int copy_desc = dup(file_desc);

    write(copy_desc, "This will be output to the file\n", 32);
    write(file_desc, "This will also be output to the file\n", 37);

    return 0;
}
```

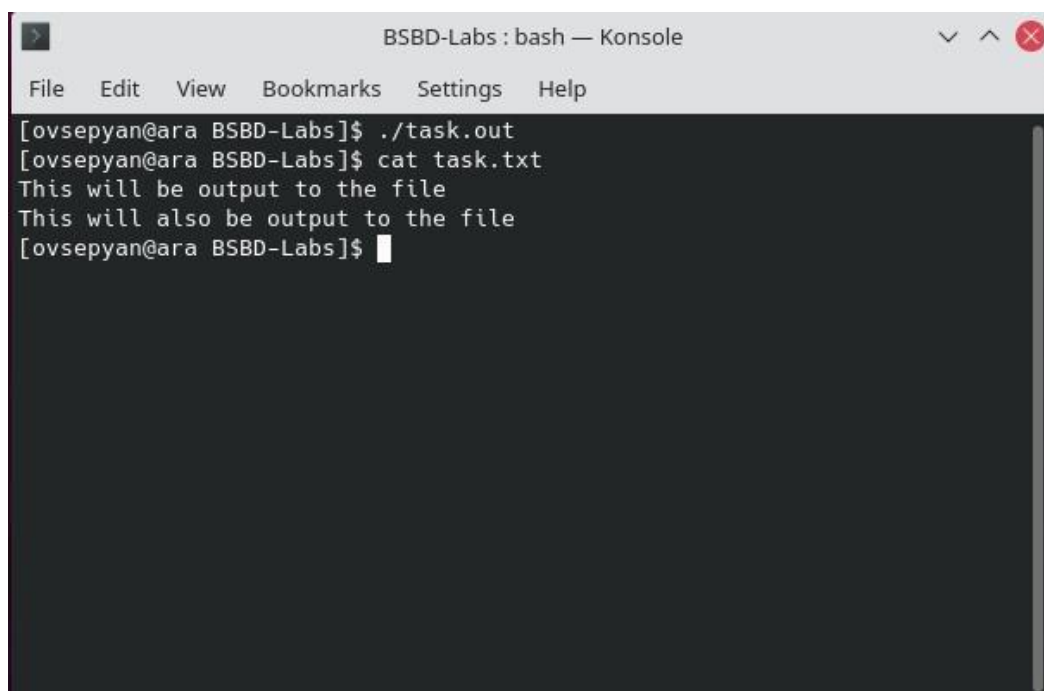


Рисунок 2 - Результат выполнения

fork

fork создает процесс-потомок-двойник, который отличается от родительского только значениями PID (идентификатор процесса) и PPID (идентификатор родительского процесса).

Пример:

```
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>
int main()
{
    fork();

    printf("Hi!\n");
    return 0;
}
```

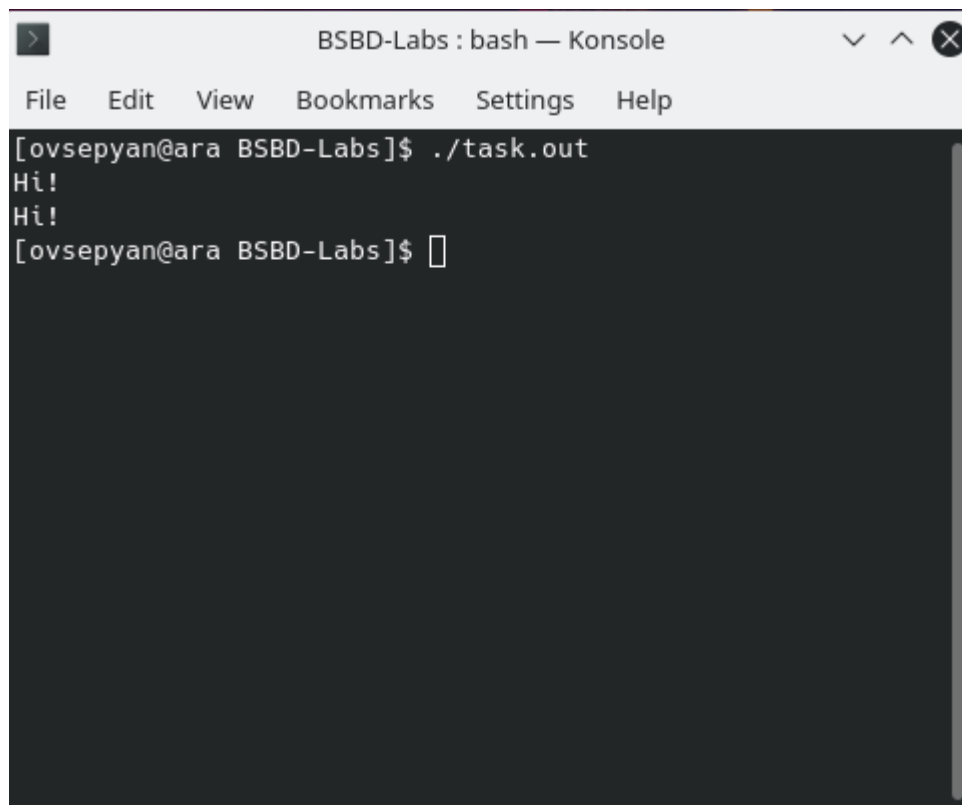


Рисунок 3 - Результат выполнения

exec

Функция **exec()** (execute) загружает и запускает другую программу. Таким образом, новая программа полностью замещает текущий процесс. Новая программа начинает свое выполнение с функции **main**. Все файлы вызывающей программы остаются открытыми. Они также являются доступными новой программе.

Суффиксы **l**, **v**, **p**, **e** в именах функций определяют формат и объем аргументов, а также каталоги, в которых нужно искать загружаемую программу:

1. **l** (список). Аргументы командной строки передаются в форме списка **arg0, arg1 argn, NULL**. Эту форму используют, если количество аргументов известно;
2. **v** (vector). Аргументы командной строки передаются в форме вектора **argv[]**. Отдельные аргументы адресуются через **argv [0], argv [1]... argv [n]**. Последний аргумент должен быть указателем **NULL**;
3. **p** (path). Обозначенный по имени файл ищется не только в текущем каталоге, но и в каталогах, определенных переменной среды **PATH**;
4. **e** (среда). Функция ожидает список переменных среды в виде вектора и не использует текущей среды.

Вывод

В ходе выполнения лабораторной работы была изучена идеология map-pages