

# Лабораторна робота №2

з дисципліни «мережеві технології»

«Основи системного адміністрування ОС FreeBSD UNIX»

Виконав: студент групи ІП-73мп  
Олександр Ковальчук

## Контрольні запитання:

### 1. Типи файлів в Unix і їх призначення.

- Файл – зберігання даних
- Директорія – формування ієрархічної структури
- Пристрій (Device) – інтерфейси доступу до апаратних пристроїв
  - Блоковий пристрій (інтерфейс доступу до пристроїв, обмін даними з якими відбувається блоками. Наприклад, жорсткі диски)
  - Символьний пристрій (інтерфейс доступу до пристроїв, обмін даними з якими відбувається посимвольно (побайтово), наприклад, пристрої вводу-виводу)
- Сокет (обмін даними між процесами на різних фізичних пристроях)
- Канал (обмін даними між процесами на одному фізичному пристрої)
- Посилання
  - жорстке посилання (hardlink) – додаткове ім'я, яке вказує на вже існуючу inode з даними
  - символічне посилання (symlink) – файл, що містить шлях на файл-призначення.

### 2. Які функції виконує інтерпретатор команд?

- Взаємодія з користувачем (введення команд користувачем, історія команд, тощо)
- Обробка шаблонів імен (розгортання wildcard на кшталт "foo\*")
- Перенаправлення ІО команд
- Керування завданнями (bg/fg, kill, ...)

### 3. У яких файлах зберігається інформація про приналежність користувача групам?

/etc/group

**4. Яку інформацію зберігають файли /etc/passwd і /etc/master.passwd і в чому між ними різниця?**

Інформація про користувачів, така як: login, real name, password, login shell, login group, uid, тощо. Див детальніше у [.man 5 passwd](#)

У FreeBSD основна інформація (в т.ч. і паролі) зберігаються у /etc/master.passwd, який доступний тільки користувачеві root. Файл /etc/passwd генерується на основі /etc/master.passwd, доступний на читання усім та містить менше даних.

**5. Сутність демон во FreeBSD.**

Демон — сервіс Unix та Unix-подібних операційних систем, що працює у фоновому режимі без прямого спілкування з користувачем.

**6. В якому вигляді зберігається пароль користувача в системі?**

У зашифрованому. Згідно [man 5 passwd](#) — у зашифрованому. Втім, відповідно до [man 3 crypt](#) це може бути і хеш (на що особисто я сподіваюся)

**7. З якою метою може використовуватися жорсткий зв'язок в файлової системі FreeBSD?**

Це, якби, зв'язок імені файлу з його індексним дескриптором. Тому, впринципі, все, що ми бачимо як звичайні файли по-суті є жорсткими зв'язками. Жорсткий зв'язок дозволяє одному файлу мати декілька різних імен. Це може бути корисним, коли необхідно зберегти працездатність посилання при переміщенні цільового файлу.

**8. Структура каталогів FreeBSD.**

Див. [man 7 hier](#).

/ — корінь ФС

/bin — busybox програми, мінімальний набір програм для адміністрування та виправлення системи

/dev — тут знаходяться файли-інтерфейси до пристроїв

/etc – конфігураційні файли та скрипти

/lib – бібліотеки, які потрібні для запуску програм із /bin

/proc – procfs

/sbin – команди, необхідні для запуску системи, але не викликаються користувачем безпосередньо

/usr – тут лежать більшість застосувань, бібліотек, тощо

/var – логи, тимчасові файли

/tmp – тимчасові файли, які не зберігаються між перезапусками

/boot – завантажувачі системи та все з ними пов'язане

## 9. Які атрибути пов'язані з кожним процесом у FreeBSD?

Ідентифікатор процесу, ідентифікатор батьківського процесу, відносний пріоритет, користувач та група від імені яких процес запущено, термінал (tty).

## 10. Яким чином забезпечується розмежування прав доступу до файлів у FreeBSD?

За допомогою атрибутів доступу. Див. [man 1 chmod](#).

## 11. Яким чином здійснюється взаємодія з процесом демоном?

За допомогою сигналів.

# Створення користувачів і груп, встановлення прав доступу до директорій.

```
#!/bin/sh
```

```
# 1. Create groups city1 and city2
```

```
# =====
```

```
pw groupadd city1
```

```
pw groupadd city2
```

```
# 2. Create users and add them to groups
```

```
# =====
```

```
pw useradd -n london -c 'London, UK' -m -s /bin/sh -g city1 -G city2
```

```
pw useradd -n paris -c 'Paris, France' -m -s /bin/sh -g city1
```

```
pw useradd -n rome -c 'Rome, Italy' -m -s /bin/sh -g city1
```

```
pw useradd -n berlin -c 'Berlin, Germany' -m -s /bin/sh -g city1
```

```
# 3. Create directories and set permissions
```

```
# =====
```

```
mkdir -p /dir09
```

```
chmod 540 /dir09
```

```
chown paris:city1 /dir09
```

```
mkdir -p /dir10
```

```
chmod 054 /dir10
```

```
chown rome:city2 /dir10
```

```
root@monica:/ # cat /etc/passwd | grep -E '(london|berlin|paris|rome)'
london:*:1001:1001:London, UK:/home/london:/bin/sh
paris:*:1002:1001:Paris, France:/home/paris:/bin/sh
rome:*:1003:1001:Rome, Italy:/home/rome:/bin/sh
berlin:*:1004:1001:Berlin, Germany:/home/berlin:/bin/sh
root@monica:/ # cat /etc/group | grep -E '(london|berlin|paris|rome)'
city2:*:1002:london
root@monica:/ # ls -la / | grep 'dir'
dr-xr-----  2 paris  city1    512 Sep 17 23:15 dir09
d---r-xr--   2 rome   city2    512 Sep 17 23:19 dir10
root@monica:/ #
```

# Скрипт зміни прав доступу до файлів

```
#!/bin/sh

# Exit on error
set -e

usage="$(basename "$0") [-h] PATH SIZE MODE

Sets given MODE to all files in PATH whose size is twice as big as SIZE.

Options:
  -h      print this help and exit
  PATH    path to the directory in which action should be performed
  SIZE    reference size. New permissions will be set to all files
          whose size is twice as big as SIZE.
  MODE    permissions bitmask, i.e. 777, 540, etc. See MODES section
          of man chmod (1) for more details

Examples:
  # Set 777 mode to all files in /root whose sizes are exactly 10 bytes
  $(basename "$0") /root 5c 777
  $(basename "$0") /root 5 777

  # Set 644 mode to all files in / whose sizes are exactly 20k
  $(basename "$0") / 10k 644
"

while getopts 'h' option; do
  case "$option" in
    h) echo "$usage"
        exit
        ;;
  esac
done

shift $((OPTIND - 1))

target_path="$1"
if [ -z "$target_path" ]; then
  echo "Missing PATH parameter" >&2
  echo "$usage" >&2
  exit 1
fi

reference_size="$2"
if [ -z "$reference_size" ]; then
  echo "Missing SIZE parameter" >&2
  echo "$usage" >&2
  exit 1
fi

mode="$3"
if [ -z "$mode" ]; then
  echo "Missing MODE parameter" >&2
  echo "$usage" >&2
```

```

    exit 1
fi

size_modifier=$(echo -n "$reference_size" | tail -c 1)
case "$size_modifier" in
    [ckMGTP])
        search_size=$(( ${reference_size%?} * 2 ))
        search_size="${search_size}${size_modifier}"
        ;;
    *)
        search_size=$(( ${reference_size} * 2 ))
        search_size="${search_size}c" # c is used for exact bytes
        ;;
esac
find "$target_path" \
    -maxdepth 1 -size "$search_size" \
    -exec chmod "$mode" {} \;

```

```

root@monica:~ # dd if=/dev/zero of=iwillchange bs=10 count=1
1+0 records in
1+0 records out
10 bytes transferred in 0.000069 secs (144913 bytes/sec)
root@monica:~ # dd if=/dev/zero of=iwillnotchange bs=11 count=1
1+0 records in
1+0 records out
11 bytes transferred in 0.000059 secs (186381 bytes/sec)
root@monica:~ # ls -la
total 64
drwxr-xr-x  3 root  wheel   512 Oct 22 17:08 .
drwxr-xr-x 21 root  wheel   512 Oct 22 16:05 ..
-rw-r--r--  2 root  wheel   957 Jul 21 02:11 .cshrc
-rw-----  1 root  wheel 16707 Sep 19 00:10 .history
-rw-r--r--  1 root  wheel   149 Jul 21 02:11 .k5login
-rw-----  1 root  wheel   122 Sep 18 00:40 .lessht
-rw-r--r--  1 root  wheel   295 Jul 21 02:11 .login
-rw-r--r--  2 root  wheel   249 Jul 21 02:11 .profile
drwx-----  2 root  wheel   512 Sep 17 21:29 .ssh
-rw-r--r--  1 root  wheel    10 Oct 22 17:08 iwillchange
-rw-r--r--  1 root  wheel    11 Oct 22 17:08 iwillnotchange
-rwxr-xr-x  1 root  wheel  1449 Sep 18 00:55 script.sh
root@monica:~ # ./script.sh . 5 777
root@monica:~ # ls -la
total 64
drwxr-xr-x  3 root  wheel   512 Oct 22 17:08 .
drwxr-xr-x 21 root  wheel   512 Oct 22 16:05 ..
-rw-r--r--  2 root  wheel   957 Jul 21 02:11 .cshrc
-rw-----  1 root  wheel 16707 Sep 19 00:10 .history
-rw-r--r--  1 root  wheel   149 Jul 21 02:11 .k5login
-rw-----  1 root  wheel   122 Sep 18 00:40 .lessht
-rw-r--r--  1 root  wheel   295 Jul 21 02:11 .login
-rw-r--r--  2 root  wheel   249 Jul 21 02:11 .profile
drwx-----  2 root  wheel   512 Sep 17 21:29 .ssh
-rwxrwxrwx  1 root  wheel    10 Oct 22 17:08 iwillchange
-rw-r--r--  1 root  wheel    11 Oct 22 17:08 iwillnotchange
-rwxr-xr-x  1 root  wheel  1449 Sep 18 00:55 script.sh
root@monica:~ #

```