

Задание 1.

hidden (1 балл)

Объяснить, где могут быть спрятаны ~11 МБ в «пустом» каталоге.

```
sobols@sobols-VirtualBox:~/unix/fs$ ls task1
sobols@sobols-VirtualBox:~/unix/fs$ du -sh task1
11M      task1
```

Добиться такого же эффекта на своей системе.

Решение.

Очевидно, что в каталоге присутствует скрытый файл. Чтобы его увидеть, надо использовать `ls -a/la`.

Добиться такого же эффекта на своей системе можно таким же образом, создав скрытый файл.

```
dd if = /dev/... of = .file.txt bs = 11MB count = 1
```

Здесь `bs` — количество записанных за 1 раз байт, `if = ...`, `of = ...` — источник, из которого копируем, и название файла, в который копируем, соответственно.

Задание 2.

mvslash (2 балла)

Проверьте на практике, в чём разница между командами

```
mv some_file foo
```

vs.

```
mv some_file foo/
```

1. Случай, когда `foo` не существует.
2. Случай, когда `foo` является файлом.
3. Случай, когда `foo` является каталогом.

Замечание. В WSL есть [bug](#), из-за которого поведение `mv` может отличаться от предполагаемого и привычного поведения на UNIX-системах.

Решение.

mv some_file foo :

1. если “foo” не существует, то команда меняет название файла на “foo”;
2. если “foo” является файлом, то тогда команда удаляет предыдущий файл, а новому дает название “foo”;
3. если “foo” является каталогом, то файл перемещается в директорию “foo/”.

mv some_file foo/ :

1. в первом случае команда выдаст ошибку, так как файла не существует, а значит нельзя переместить “some_file” по указанной директории;
2. во втором случае команда также выдаст ошибку;
3. в третьем случае результат аналогичен команде “mv some_file foo”.

Задание 3.

dupdirs (2 балла)

Команда

```
$ mkdir -p foo/bar/bar/{bar,baz} && touch  
foo/bar/bar/{1,2}.txt,bar/{3,4,5}.txt,baz/{6,7,8}.txt}
```

создаёт иерархию каталогов

```
foo  
├── bar  
│   └── bar  
│       ├── 1.txt  
│       ├── 2.txt  
│       └── bar  
│           ├── 3.txt  
│           ├── 4.txt  
│           └── 5.txt  
│       └── baz  
│           ├── 6.txt  
│           ├── 7.txt  
│           └── 8.txt
```

4 directories, 8 files

Нужно за не более чем три вызова каких-то утилит убрать лишний `bar`, чтобы получилось так:

```
foo
├── bar
│   ├── 1.txt
│   ├── 2.txt
│   └── bar
│       ├── 3.txt
│       ├── 4.txt
│       └── 5.txt
└── baz
    ├── 6.txt
    ├── 7.txt
    └── 8.txt

3 directories, 8 files
```

Замечание. Копировать содержимое файлов в процессе нельзя (представьте, что файлы большие, копирование занимает много времени и места на диске).

Решение.

Чтобы убрать лишний `bar`, необходимо вызвать следующие утилиты:

`mv bar newName` (даем директории `bar` новое название) →
`mv newName/bar` (перемещаем эту директорию) → `rmdir newName`
(удаляем `newName`)

Задание 4.

cannottouch (2 балла)

Каким образом можно добиться такой непонятной ошибки от команды `touch`?

```
sobols@sobols-VirtualBox:~/unix/fs/cannottouch/subdir$ ls -la
total 0

sobols@sobols-VirtualBox:~/unix/fs/cannottouch/subdir$ touch file.txt
touch: cannot touch 'file.txt': No such file or directory
```

Решение.

Добиться такой ошибки “поможет” использование двух терминалов.

```
mkdir newFolder - 1    →    cd newFolder - 1    →  
rm -r newFolder - 2    →    →    touch file.txt - 1
```

Задание 5.

tardir (2 балла)

Команда

```
$ mkdir -p foo/bar/{baz,quux} && touch  
foo/bar/{1,2}.txt,baz/{3,4}.txt,quux/{5,6}.txt
```

создаёт иерархию каталогов

```
foo  
├── bar  
│   ├── 1.txt  
│   ├── 2.txt  
│   └── baz  
│       ├── 3.txt  
│       └── 4.txt  
└── quux  
    ├── 5.txt  
    └── 6.txt
```

Представьте, что перед вами стоит задача — запаковать содержимое каталога `foo/bar` в архив.

Это можно сделать несколькими способами. Проанализируйте, в чём разница между следующими командами.

1. `tar cf res1.tar ./foo/bar`
2. `tar cf res2.tar foo/bar`
3. `tar cf res3.tar -C foo/bar .`
4. `cd foo/bar && tar cvf ../../res4.tar * && cd -`

В результате получается четыре разных архива! Чтобы это понять, научитесь печатать содержимое архива (список файлов в нём).

Решение.

`tar tf res-i.tar` — просмотр содержимого архива.

`tar cf res-1.tar ./foo/bar` - команда позволяет все хранить файлы внутри директории `./` таким же образом, как и в директории

tar cf res-2.tar foo/bar - все файлы хранятся таким же образом, как и в директории

tar cf res-3.tar -C foo/bar . - команда позволяет все хранить файлы внутри директории *./* без *foo/bar*

*cd foo/bar && tar cvf ../res4.tar * && cd* - команда позволяет все хранить файлы без *foo/bar*

c – создание нового архива

v – последовательный вывод названий файлов

f – использование следующего *tar* архива для эксплуатации.

Задание 6.

extractone (3 балла)

Скачайте архив по ссылке <https://acm.bsu.by/download/big2.tar.bz2>.

Не распаковывая все файлы на диск, извлеките содержимое файла с именем *secret.txt*.

Решение.

Первым делом необходимо скопировать полную директорию файла *secret.txt*. А затем следует выполнить команды перемещения файла и задания ему необходимых прав доступа.

```
tar -xvjf big2.tar.bz2 -wildcards " * secret.txt →  
→ mv enmfx4ki/2lvbefws/qp5ww8a0/dkwsklg4/kvwq8wg2/secret.txt →  
→ chmod 644 secret.txt
```