

# РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

## ОТЧЕТ

### ПО ЛАБОРАТОРНОЙ РАБОТЕ № 6

*дисциплина:* Архитектура компьютера

Студентка: Симбине Камила Шеймиле

Группа: НПИбд-03-23

МОСКВА

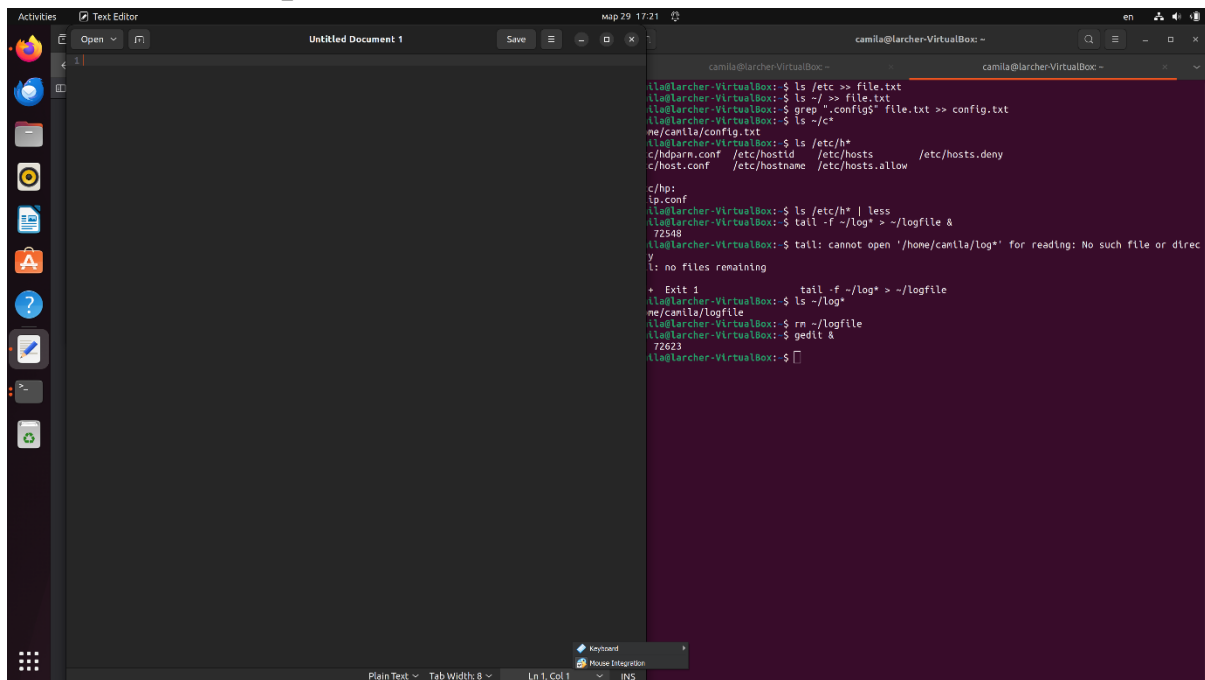
2024 г.

# Поиск файлов. Перенаправление ввода-вывода. Просмотр запущенных процессов

## Цель работы

Ознакомление с инструментами поиска файлов и фильтрации текстовых данных. Приобретение практических навыков: по управлению процессами (и заданиями), по проверке использования диска и обслуживанию файловых систем.

## Выполнение работы



```
camila@larcher-VirtualBox: ~  
ila@larcher-VirtualBox: $ ls /etc >> file.txt  
ila@larcher-VirtualBox: $ ls ~/ >> file.txt  
ila@larcher-VirtualBox: $ grep "config" file.txt >> config.txt  
ila@larcher-VirtualBox: $ ls ~/c*  
me/canila/config.txt  
ila@larcher-VirtualBox: $ ls /etc/h* | less  
c/hdparm.conf /etc/hostid /etc/hosts /etc/hosts.deny  
c/host.conf /etc/hostname /etc/hosts.allow  
c/hp:  
lp.conf  
ila@larcher-VirtualBox: $ ls /etc/h* | less  
ila@larcher-VirtualBox: $ tail -f ~/log* > ~/logfile &  
72548  
ila@larcher-VirtualBox: $ tail: cannot open '/home/camila/log*' for reading: No such file or direc  
y  
ls: no files remaining  
+ Exit 1  
ila@larcher-VirtualBox: $ tail -f ~/log* > ~/logfile  
ila@larcher-VirtualBox: $ ls ~/log*  
me/canila/logfile  
ila@larcher-VirtualBox: $ rm ~/logfile  
ila@larcher-VirtualBox: $ gedit &  
72623  
ila@larcher-VirtualBox: $
```

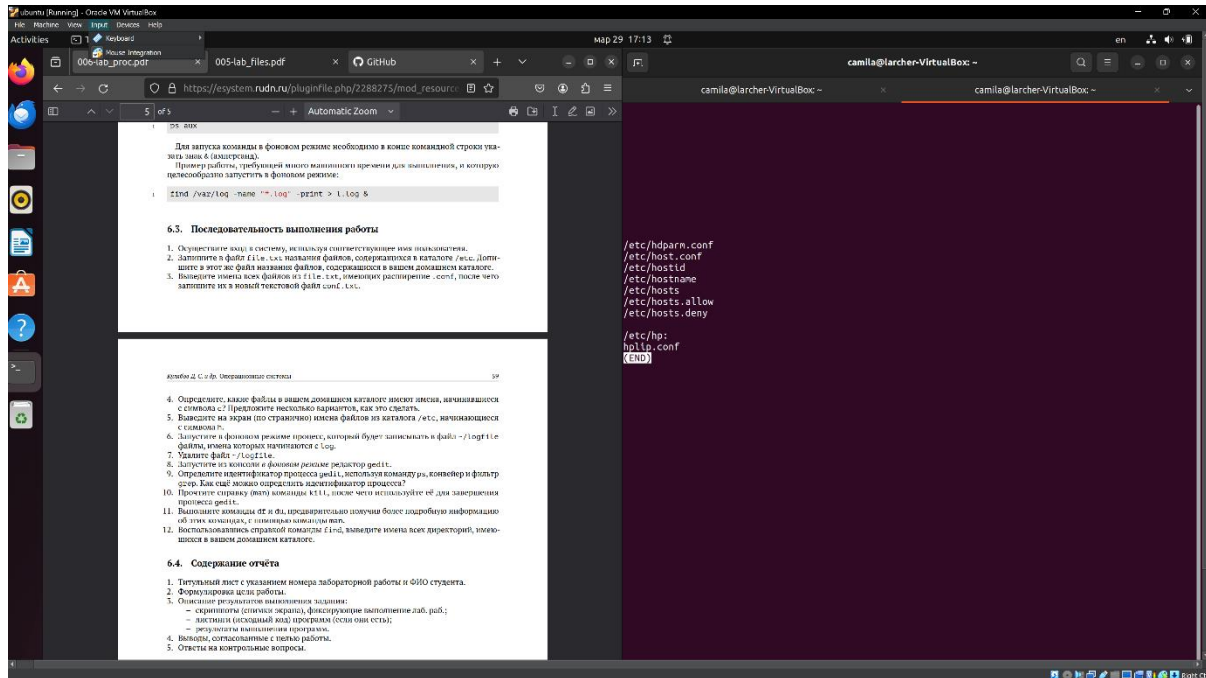
Мы записали имена файлов в каталоге /etc в файл с именем file.txt, и добавил имена файлов в ваш домашний каталог к тому же файлу. Затем мы перечислили файлы в каталоге "/etc" и добавили этот список в файл с именем "file.txt". Аналогично, файлы в домашнем каталоге ( ` ~ ` ) перечисляются и добавляются к одному и тому же файлу.

Затем мы приступили к отображению имен всех файлов из file.txt, которые имеют расширение .conf, а затем запишите их в новый текстовый файл с именем conf.txt\*\*\*! Здесь команда grep используется для поиска строк, содержащих шаблон ".conf\$" (файлы с расширением .conf) в файле file.txt-досье. Затем соответствующие строки записываются в новый файл с именем conf.txt.

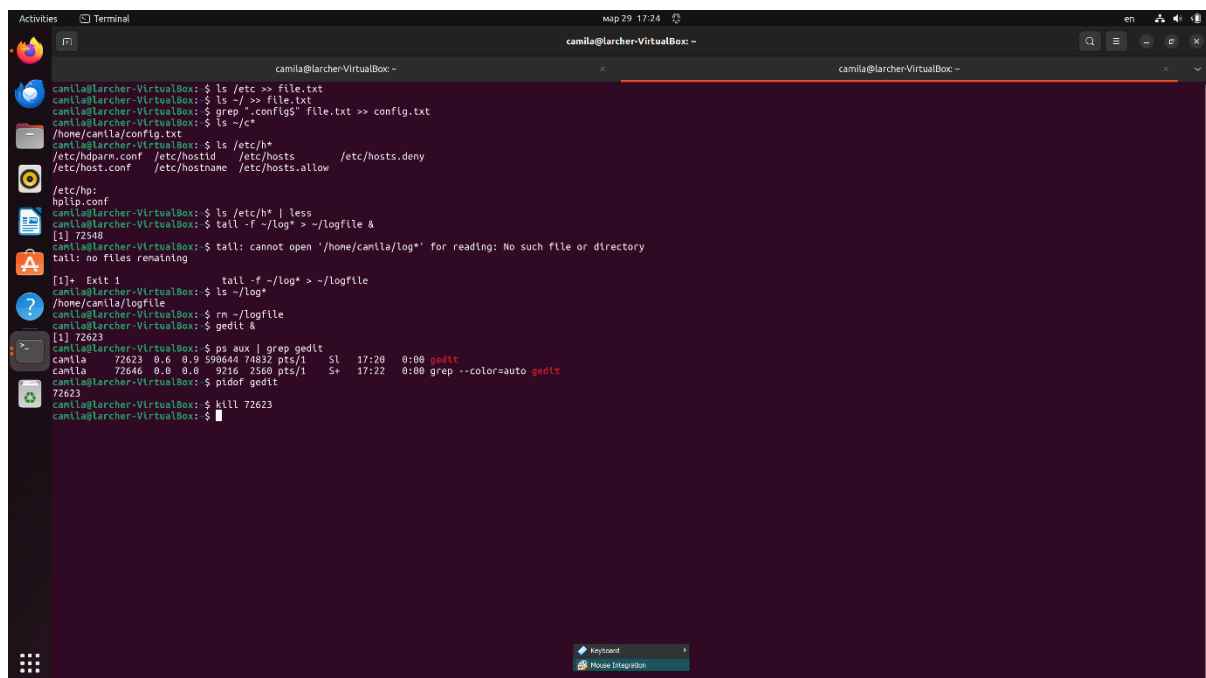
Затем мы перечисляем, чтобы определить, какие файлы в вашем домашнем каталоге имеют имена, начинающиеся с буквы "c". На этом шаге используется команда `ls` с подстановочным знаком ('c\*') для перечисления файлов в домашнем каталоге, начинающихся с буквы 'c'.

Затем, наконец, мы выводим (страница за страницей) имена файлов в каталоге /etc,

начинающиеся с буквы "h".! Команда "ls" выводит список файлов в каталоге "/etc", начинающихся с буквы "h", а команда "less" используется для отображения выходных данных по одной странице за раз для удобства просмотра. Это видно именно на изображении ниже!



Затем, наконец, мы выводим (страница за страницей) имена файлов в каталоге /etc, начинающиеся с буквы "h".!



Мы запустили фоновый процесс, который будет непрерывно добавлять содержимое файлов, начинающихся с "log", в файл с именем ~ / logfile.! На этом этапе использовалась команда 'tail' с параметром '-f' для непрерывного вывода содержимого файлов, начинающихся с 'log' ('~/log\*'), в файл '~ / logfile'. Символ '&'

используется для выполнения команды в фоновом режиме. Чтобы подтвердить существование файла журнала, мы запустили `ls`.

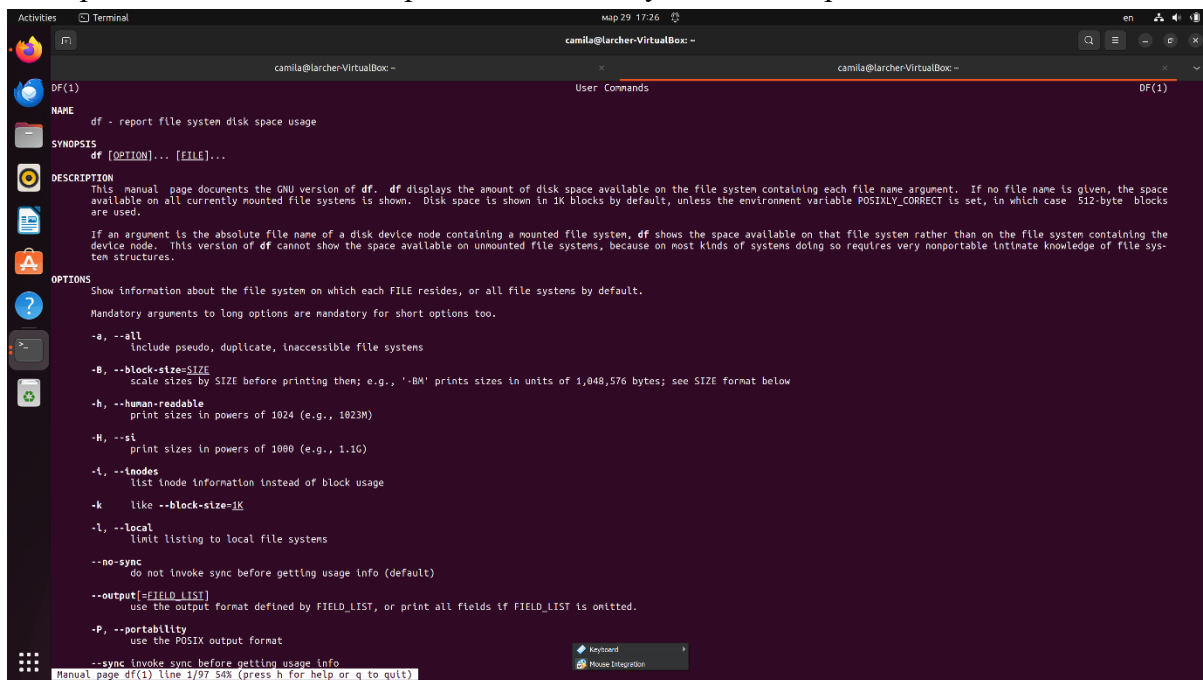
Затем мы удаляем файл `~/logfile`, используя команду `'rm'`

Мы запустили редактор `gedit` из консоли в фоновом режиме.

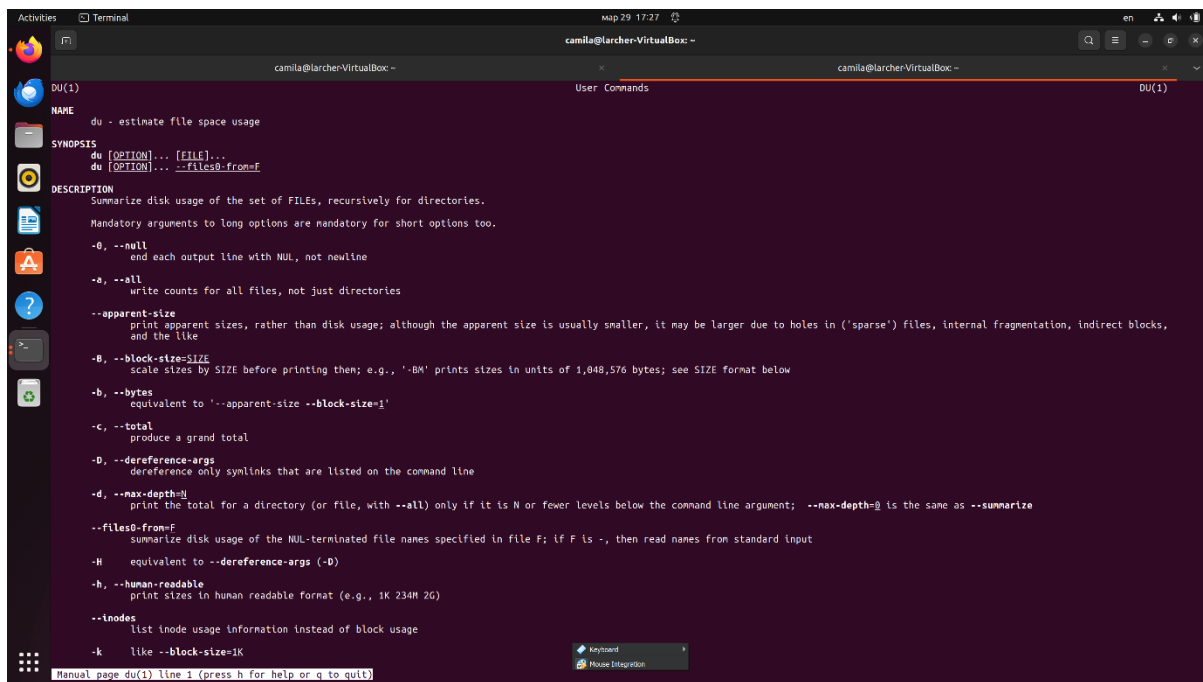
Символ `" & "` запускает команду в фоновом режиме, позволяя вам продолжать использовать терминал.

Наконец, мы определили идентификатор процесса (PID) процесса `gedit` с помощью команды `ps`, конвейера и фильтра `grep`. Кроме того, предложите другие методы идентификации идентификатора процесса.

- Здесь команда `'ps'` перечисляет все запущенные процессы, а выходные данные передаются по конвейеру (`'|'`) в `'grep'` для фильтрации строк, содержащих `"gedit"`, что помогает идентифицировать PID процесса `gedit`. Команда `'pidof'` упоминается как альтернативный метод непосредственного получения PID процесса.

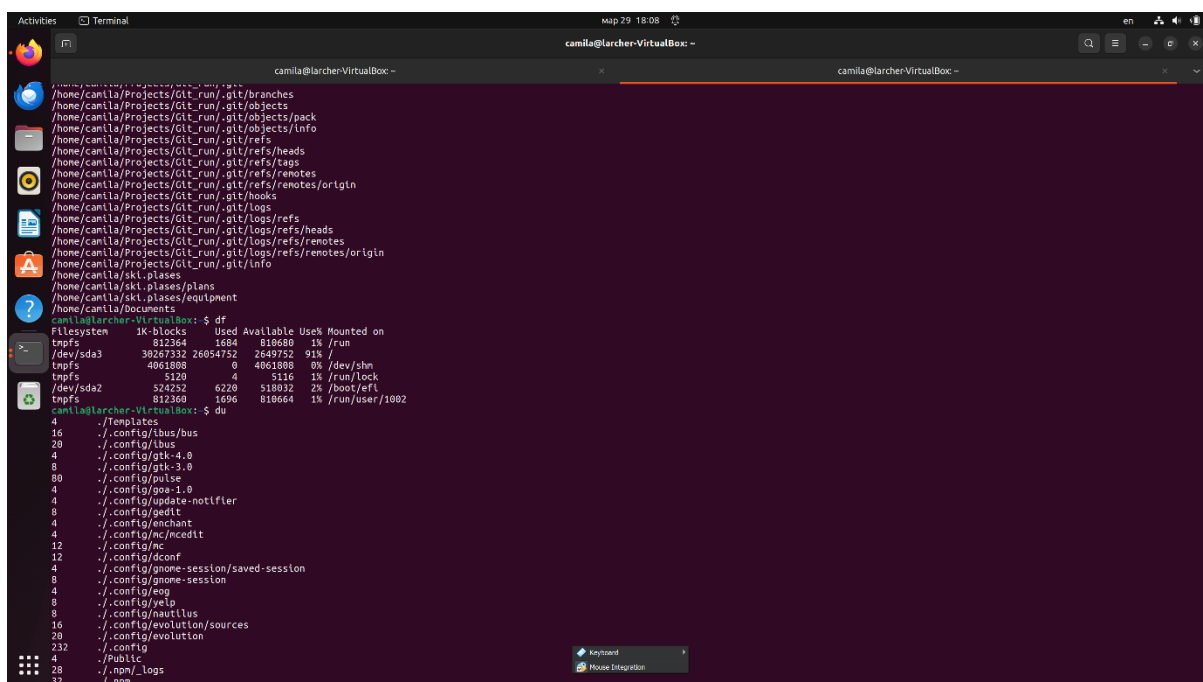


- Команда `" man "` используется для просмотра страниц руководства по командам `df` (Disk free)



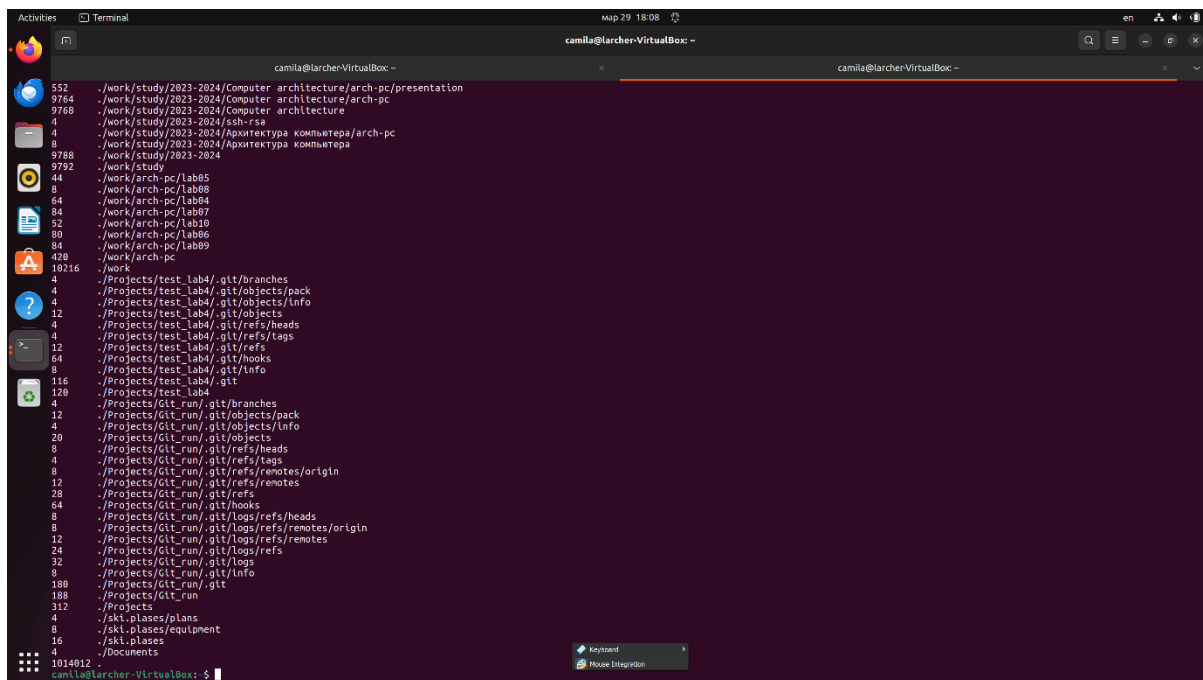
```
Activities Terminal map 29 17:27 en
camila@larcher-VirtualBox -
DU(1)
NAME
du - estimate file space usage
SYNOPSIS
du [OPTION]... [FILE]...
du [OPTION]... --files0-from=F
DESCRIPTION
Summarize disk usage of the set of FILES, recursively for directories.
Mandatory arguments to long options are mandatory for short options too.
-O, --null
end each output line with NUL, not newline
-a, --all
write counts for all files, not just directories
--apparent-size
print apparent sizes, rather than disk usage; although the apparent size is usually smaller, it may be larger due to holes in ('sparse') files, internal fragmentation, indirect blocks, and the like
-B, --block-size=SIZE
scale sizes by SIZE before printing them; e.g., '-BM' prints sizes in units of 1,048,576 bytes; see SIZE format below
-b, --bytes
equivalent to '--apparent-size --block-size=1'
-c, --total
produce a grand total
-D, --dereference-args
dereference only symlinks that are listed on the command line
-d, --max-depth=N
print the total for a directory (or file, with --all) only if it is N or fewer levels below the command line argument; --max-depth=0 is the same as --summarize
--files0-from=F
summarize disk usage of the NUL-terminated file names specified in file F; if F is -, then read names from standard input
-H
equivalent to --dereference-args (-D)
-h, --human-readable
print sizes in human readable format (e.g., 1K 234M 2G)
--inodes
list inode usage information instead of block usage
-k
like --block-size=1K
Manual page du(1) line 1 (press h for help or q to quit)
```

- Команда " man "используется для просмотра страниц руководства по командам " du " (disk usage) с подробной информацией об их использовании и параметрах



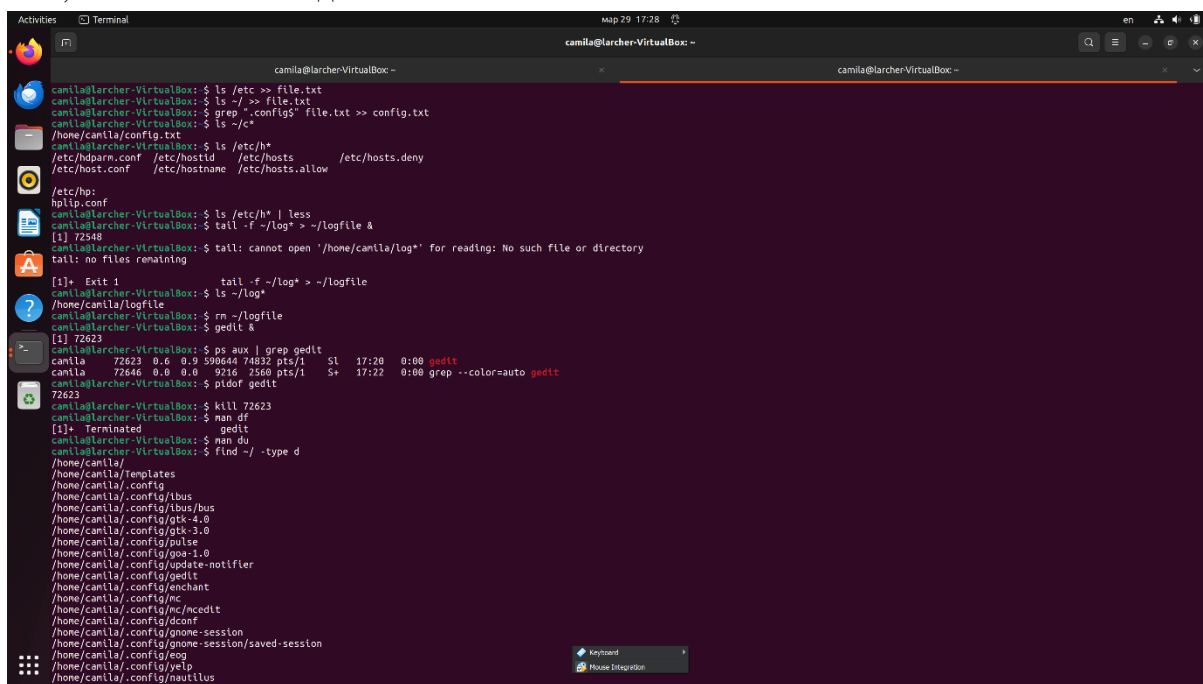
```
Activities Terminal map 29 18:06 en
camila@larcher-VirtualBox -
/home/cantila/Projects/git_run/.git/branches
/home/cantila/Projects/git_run/.git/objects
/home/cantila/Projects/git_run/.git/objects/pack
/home/cantila/Projects/git_run/.git/objects/info
/home/cantila/Projects/git_run/.git/refs
/home/cantila/Projects/git_run/.git/refs/heads
/home/cantila/Projects/git_run/.git/refs/tags
/home/cantila/Projects/git_run/.git/refs/remotes
/home/cantila/Projects/git_run/.git/refs/remotes/origin
/home/cantila/Projects/git_run/.git/hooks
/home/cantila/Projects/git_run/.git/logs
/home/cantila/Projects/git_run/.git/logs/refs
/home/cantila/Projects/git_run/.git/logs/refs/heads
/home/cantila/Projects/git_run/.git/logs/refs/remotes
/home/cantila/Projects/git_run/.git/logs/refs/remotes/origin
/home/cantila/Projects/git_run/.git/info
/home/cantila/ski_places
/home/cantila/ski_places/plans
/home/cantila/ski_places/equipment
/home/cantila/Documents
camila@larcher-VirtualBox:~$ df
Filesystem            1K-blocks    Used Available Use% Mounted on
tmpfs                  812360      1684    810676   1% /run
/dev/sda3             30267332 26054752  2649752   91% /
tmpfs                 4061808      0    4061808   0% /dev/shm
tmpfs                  5120         4      5116   1% /run/lock
/dev/sda2             524252      6220    518032   2% /boot/efi
tmpfs                 812360     1696    810664   1% /run/user/1002
camila@larcher-VirtualBox:~$ du
4  ./Templates
16 ./config/ibus/bus
20 ./config/ibus
4  ./config/gtk-4.0
8  ./config/gtk-3.0
80 ./config/pulse
4  ./config/gnome-1.0
4  ./config/update-notifier
8  ./config/gedit
4  ./config/enchant
4  ./config/mc/ncedit
12 ./config/mc
12 ./config/dconf
4  ./config/gnome-session/saved-session
4  ./config/gnome-session
4  ./config/eop
8  ./config/yelp
8  ./config/nautilus
16 ./config/evolution/sources
20 ./config/evolution
232 ./config
4  ./Public
28 ./npm/_logs
32 ./npm
```

После получения более подробной информации об этих командах с помощью команды man, выполните команды df.



```
camila@larcher-VirtualBox: ~  
552 ./work/study/2023-2024/Computer architecture/arch-pc/presentation  
9764 ./work/study/2023-2024/Computer architecture/arch-pc  
9768 ./work/study/2023-2024/Computer architecture  
4 ./work/study/2023-2024/Архитектура компьютера/arch-pc  
8 ./work/study/2023-2024/Архитектура компьютера  
9788 ./work/study/2023-2024  
9792 ./work/study  
44 ./work/arch-pc/lab85  
8 ./work/arch-pc/lab88  
64 ./work/arch-pc/lab84  
84 ./work/arch-pc/lab87  
52 ./work/arch-pc/lab10  
80 ./work/arch-pc/lab86  
84 ./work/arch-pc/lab89  
428 ./work/arch-pc  
10216 ./work  
4 ./Projects/test_lab4/.git/branches  
4 ./Projects/test_lab4/.git/objects/pack  
4 ./Projects/test_lab4/.git/objects/info  
12 ./Projects/test_lab4/.git/objects  
4 ./Projects/test_lab4/.git/refs/heads  
4 ./Projects/test_lab4/.git/refs/tags  
12 ./Projects/test_lab4/.git/refs  
64 ./Projects/test_lab4/.git/hooks  
8 ./Projects/test_lab4/.git/info  
116 ./Projects/test_lab4/.git  
128 ./Projects/test_lab4  
4 ./Projects/Git_run/.git/branches  
12 ./Projects/Git_run/.git/objects/pack  
4 ./Projects/Git_run/.git/objects/info  
20 ./Projects/Git_run/.git/objects  
8 ./Projects/Git_run/.git/refs/heads  
4 ./Projects/Git_run/.git/refs/tags  
8 ./Projects/Git_run/.git/refs/remotes/origin  
12 ./Projects/Git_run/.git/refs/remotes  
28 ./Projects/Git_run/.git/refs  
64 ./Projects/Git_run/.git/hooks  
8 ./Projects/Git_run/.git/logs/refs/heads  
8 ./Projects/Git_run/.git/logs/refs/remotes/origin  
12 ./Projects/Git_run/.git/logs/refs/remotes  
24 ./Projects/Git_run/.git/logs/refs  
32 ./Projects/Git_run/.git/logs  
8 ./Projects/Git_run/.git/info  
188 ./Projects/Git_run/.git  
188 ./Projects/Git_run  
312 ./Projects  
4 ./ski.places/plans  
8 ./ski.places/equipment  
16 ./ski.places  
4 ./Documents  
1014012  
camila@larcher-VirtualBox: ~
```

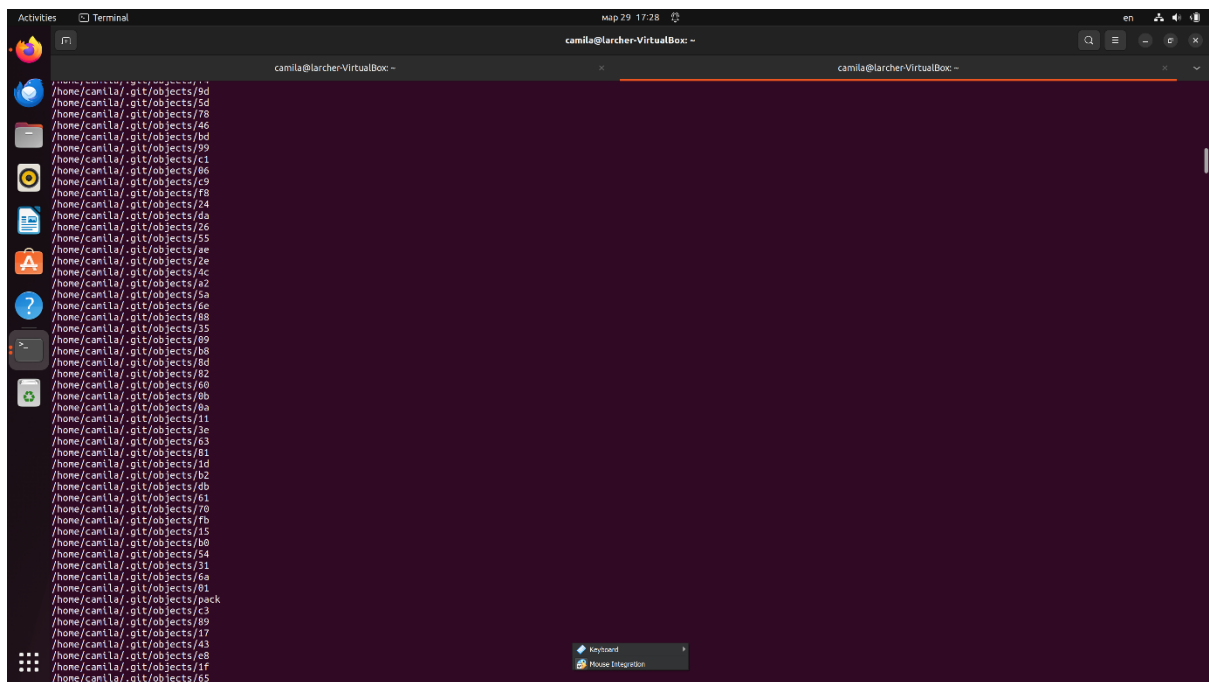
После получения более подробной информации об этих командах с помощью команды `man`, выполните команды `du`.



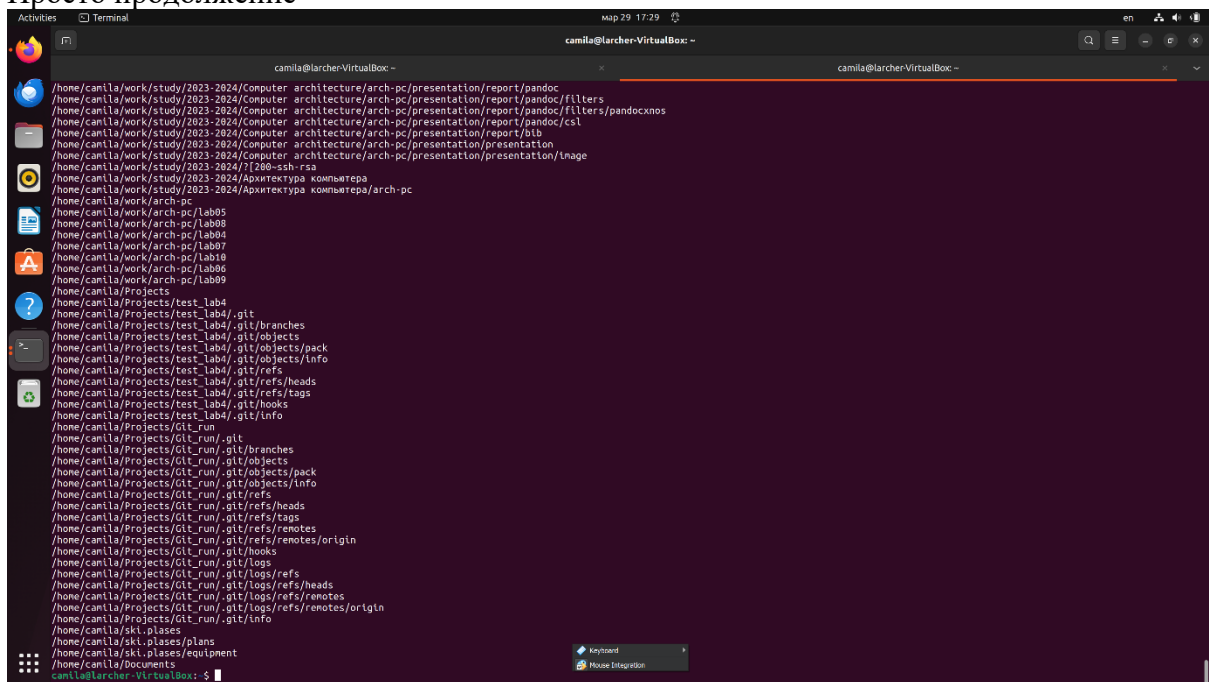
```
camila@larcher-VirtualBox: ~  
camila@larcher-VirtualBox: ~$ ls /etc >> file.txt  
camila@larcher-VirtualBox: ~$ ./ >> file.txt  
camila@larcher-VirtualBox: ~$ grep "config" file.txt >> config.txt  
camila@larcher-VirtualBox: ~$ ls -lc*  
/home/camila/config.txt  
camila@larcher-VirtualBox: ~$ ls /etc/h*  
/etc/hdparm.conf /etc/hostid /etc/hosts /etc/hosts.deny  
/etc/host.conf /etc/hostname /etc/hosts.allow  
/etc/hp:  
hplip.conf  
camila@larcher-VirtualBox: ~$ ls /etc/h* | less  
camila@larcher-VirtualBox: ~$ tail -f ~/log* > ~/logfile &  
[1] 72548  
camila@larcher-VirtualBox: ~$ tail: cannot open '/home/camila/log*' for reading: No such file or directory  
tail: no files remaining  
[1]+ Exit 1  
camila@larcher-VirtualBox: ~$ tail -f ~/log* > ~/logfile  
camila@larcher-VirtualBox: ~$ ls ~/log*  
/home/camila/logfile  
camila@larcher-VirtualBox: ~$ rm ~/logfile  
camila@larcher-VirtualBox: ~$ gedit &  
[1] 72623  
camila@larcher-VirtualBox: ~$ ps aux | grep gedit  
camila 72623 0.6 0.9 590644 74832 pts/1 Sl 17:20 0:00 gedit  
camila 72646 0.0 0.0 9216 2560 pts/1 S+ 17:22 0:00 grep --color=auto gedit  
camila@larcher-VirtualBox: ~$ pidof gedit  
72623  
camila@larcher-VirtualBox: ~$ kill 72623  
camila@larcher-VirtualBox: ~$ man df  
[1]+ Terminated  
camila@larcher-VirtualBox: ~$ man du  
camila@larcher-VirtualBox: ~$ find ~/ -type d  
/home/camila/  
/home/camila/.templates  
/home/camila/.config  
/home/camila/.config/ibus  
/home/camila/.config/ibus/bus  
/home/camila/.config/gtk-4.0  
/home/camila/.config/gtk-3.0  
/home/camila/.config/pulse  
/home/camila/.config/gss-1.0  
/home/camila/.config/update-notifier  
/home/camila/.config/gedit  
/home/camila/.config/enchant  
/home/camila/.config/nc  
/home/camila/.config/nc/mcedit  
/home/camila/.config/dconf  
/home/camila/.config/gnome-session  
/home/camila/.config/gnome-session/saved-session  
/home/camila/.config/yelp  
/home/camila/.config/nautilus
```

Мы использовали команду `'kill'` для отправки сигнала на завершение (`'kill'`) процесса `gedit`, указав его PID, который мы получили ранее в данном случае 72623

Мы используем руководство команды `find` для отображения имен всех каталогов в вашем домашнем каталоге. Команда "найти" используется для поиска файлов и каталогов. На этом этапе он используется для поиска (`'find'`) всех каталогов (`- type d`) в домашнем каталоге (`~/`).



## Просто продолжение



## Просто продолжение

# Контрольные вопросы для самопроверки

## 1. Какие потоки ввода вывода вы знаете?

- В UNIX-подобных системах существует три стандартных потока ввода-вывода: stdin (стандартный поток ввода), stdout (стандартный поток вывода), и stderr (стандартный поток ошибок).

## 2. Объясните разницу между операцией > и >>.

- Операция ``>`` используется для перенаправления вывода команды в файл, при этом существующий файл будет перезаписан. Операция ``>>`` также перенаправляет вывод команды в файл, но при этом данные будут добавлены в конец файла, не перезаписывая его содержимое.

### **3. Что такое конвейер?**

- Конвейер (pipeline) это механизм в UNIX-подобных операционных системах, который позволяет объединять вывод одной команды с вводом другой, создавая последовательность команд, где выход одной команды является входом для другой.

### **4. Что такое процесс? Чем это понятие отличается от программы?**

- Процесс - это экземпляр программы, запущенный в операционной системе. Программа - это набор инструкций и данных, которые могут быть выполнены компьютером, тогда как процесс - это активное выполнение программы в определенный момент времени.

### **5. Что такое PID и GID?**

- PID (Process ID) - это уникальный идентификатор процесса, который используется операционной системой для идентификации каждого процесса. GID (Group ID) - это идентификатор группы, который определяет принадлежность пользователя к определенной группе пользователей.

### **6. Что такое задачи и какая команда позволяет ими управлять?**

- Задачи - это активные процессы или команды, выполняемые в операционной системе. Команда ``ps`` позволяет просматривать информацию о процессах, а команда ``kill`` - завершать процессы.

### **7. Найдите информацию об утилитах `top` и `htop`. Каковы их функции?**

- ``top`` и ``htop`` - это утилиты мониторинга процессов в UNIX-подобных системах. Они отображают информацию о процессах, загрузке системы, использовании ресурсов и позволяют управлять процессами.

### **8. Назовите и дайте характеристику команде поиска файлов. Приведите примеры использования этой команды.**

- Команда поиска файлов в UNIX-подобных системах называется ``find``. Она позволяет искать файлы и директории по различным критериям. Например:

- ``find /home/user -name "*.txt"`` - ищет все файлы с расширением `.txt` в домашнем каталоге пользователя.



- `find /var/log -type f -mtime +7` - ищет все файлы в каталоге `/var/log`, которые были изменены более 7 дней назад.

**9. Можно ли по контексту (содержанию) найти файл? Если да, то как?**

- Да, можно использовать команду `grep` для поиска файлов по содержимому. Например, `grep "search_term" /path/to/files/*` ищет все файлы в указанном пути, содержащие указанный поисковый термин.

**10. Как определить объем свободной памяти на жёстком диске?**

- Для этого можно использовать команду `df -h`, которая выводит информацию о доступном месте на файловых системах.

**11. Как определить объем вашего домашнего каталога?**

- Это можно сделать с помощью команды `du -sh ~`, которая покажет размер вашего домашнего каталога в удобно читаемом формате.

**12. Как удалить зависший процесс?**

- Для завершения зависшего процесса можно использовать команду `kill` с соответствующим PID зависшего процесса. Например, `kill PID`. Если процесс не реагирует на сигнал завершения, можно использовать команду `kill -9 PID`, чтобы принудительно завершить его выполнение.