1. Какого типа команда cd? Попробуйте объяснить, почему она именно такого типа; опишите ход своих мыслей, если считаете что она могла бы быть другого типа.

Это внутренняя команда в Bash. Является таковой для производительности скорей всего.

1. Какая альтернатива без pipe команде grep <some\_string> <some\_file> | wc -l? man grep поможет в ответе на этот вопрос. Ознакомьтесь с [документом](http://www.smallo.ruhr.de/award.html) о других подобных некорректных вариантах использования pipe.

Речь же скорее о бессмысленных конструкциях, порождающих лишние процессы и являющихся заведомо не продуктивными

vagrant@vagrant:~$ grep -c ls .bashrc

9

vagrant@vagrant:~$ grep ls .bashrc | wc -l

9

1. Какой процесс с PID 1 является родителем для всех процессов в вашей виртуальной машине Ubuntu 20.04?

systemd 1 root txt REG 253,0 1620224 536074 /usr/lib/systemd/systemd

1. Как будет выглядеть команда, которая перенаправит вывод stderr ls на другую сессию терминала?

Могу предположить ,что это будет ls 2 > /dev/tty2

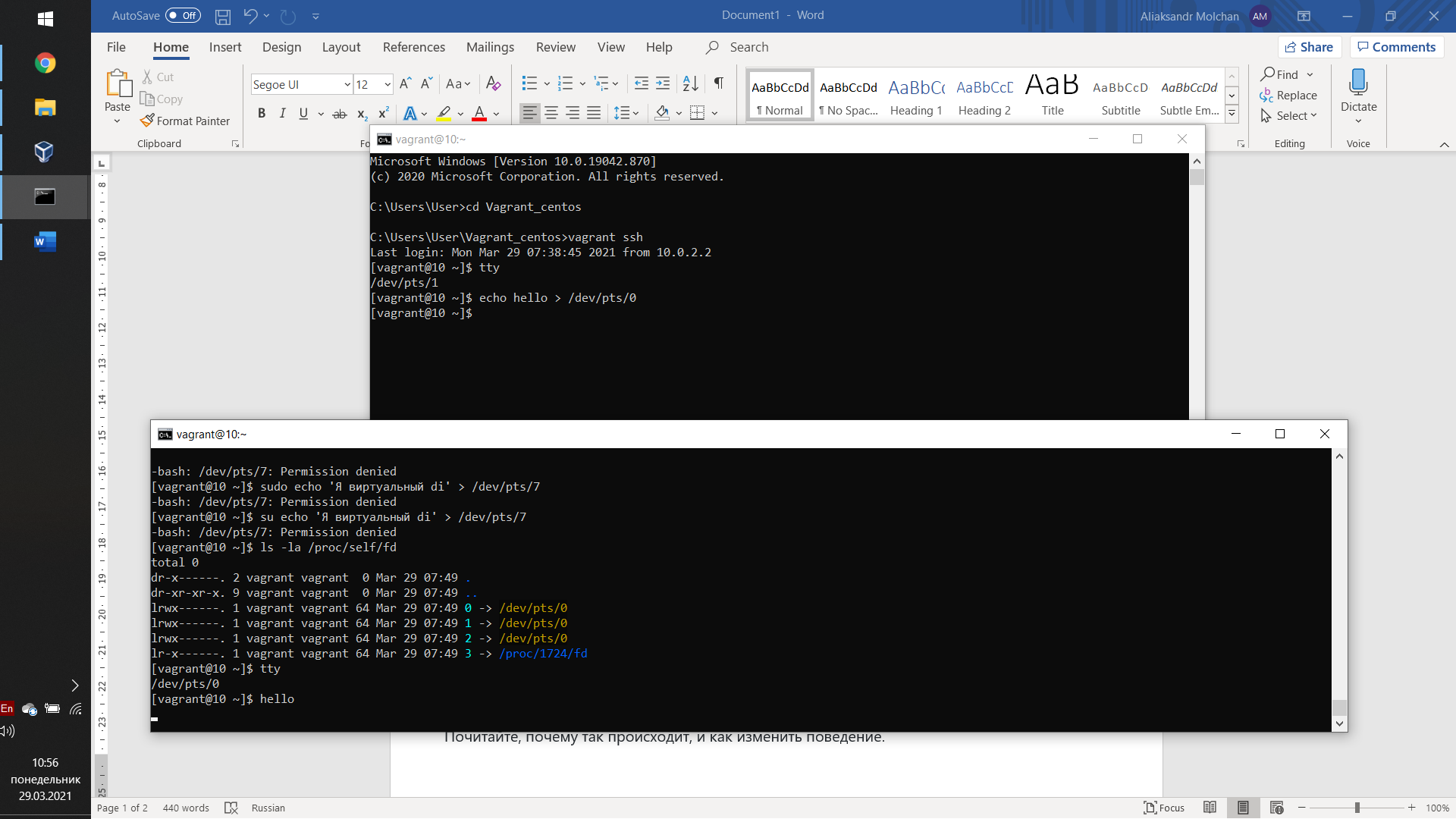
1. Получится ли одновременно передать команде файл на stdin и вывести ее stdout в другой файл? Приведите работающий пример.

Cat -n test.file > stdout.fil

grep ls .bashrc > stdout.fil

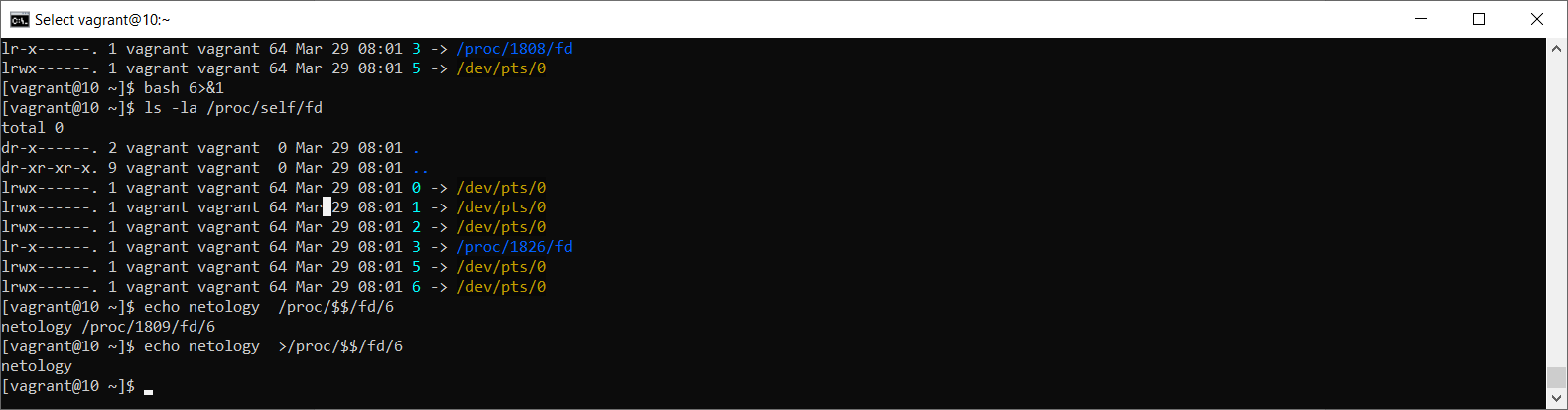
1. Получится ли вывести находясь в графическом режиме данные из PTY в какой-либо из эмуляторов TTY? Сможете ли вы наблюдать выводимые данные?

Да, просто перенаправив поток из PTY в любую из TTY . Данные можно будет увидеть, переключившись в эту консоль



1. Выполните команду bash 5>&1. К чему она приведет? Что будет, если вы выполните echo netology > /proc/$$/fd/5? Почему так происходит?

Данной командой мы создаем ещё один поток stdout. Отправляя туда вывод команды echo получаем отображение её на экране



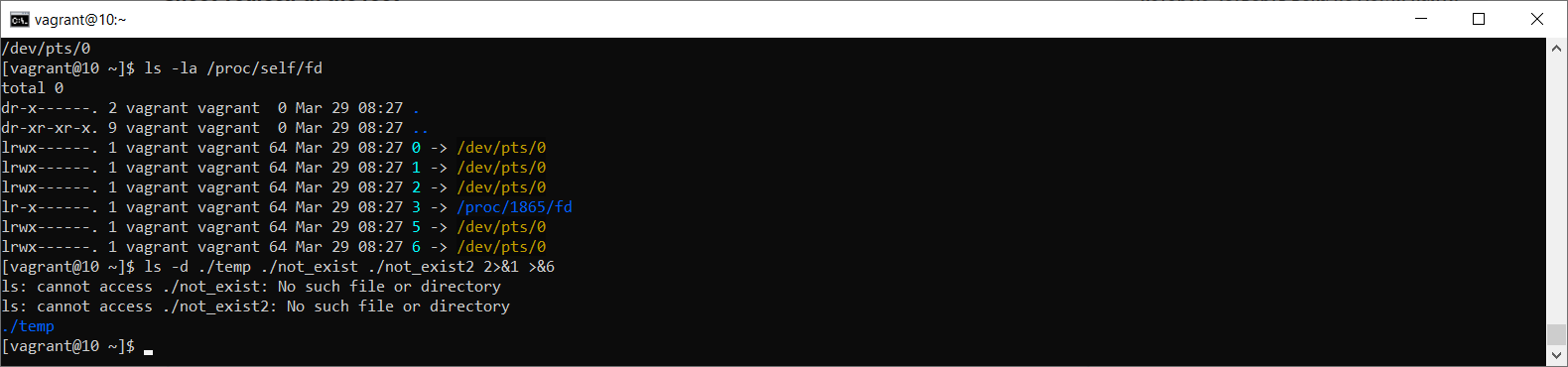
1. Получится ли в качестве входного потока для pipe использовать только stderr команды, не потеряв при этом отображение stdout на pty? Напоминаем: по умолчанию через pipe передается только stdout команды слева от | на stdin команды справа. Это можно сделать, поменяв стандартные потоки местами через промежуточный новый дескриптор, который вы научились создавать в предыдущем вопросе.

[vagrant@10 ~]$ ls -d ./temp ./not\_exist ./not\_exist2 2>&1 >&6

ls: cannot access ./not\_exist: No such file or directory

ls: cannot access ./not\_exist2: No such file or directory

./temp



1. Что выведет команда cat /proc/$$/environ? Как еще можно получить аналогичный по содержанию вывод?

Хранилище переменных окружения :

env

ps e -p $$

xargs --null --max-args=1 echo < /proc/$$/environ

1. Используя man, опишите что доступно по адресам /proc/<PID>/cmdline, /proc/<PID>/exe.

/proc/[pid]/cmdline

This holds the complete command line for the process, unless the process is a zombie. In the latter case, there is nothing in this file: that is, a read on this file will return

0 characters. The command-line arguments appear in this file as a set of strings separated by null bytes ('\0'), with a further null byte after the last string.

/proc/[pid]/exe

Under Linux 2.2 and later, this file is a symbolic link containing the actual pathname of the executed command. This symbolic link can be dereferenced normally; attempting to

open it will open the executable. You can even type /proc/[pid]/exe to run another copy of the same executable as is being run by process [pid]. In a multithreaded process, the

contents of this symbolic link are not available if the main thread has already terminated (typically by calling pthread\_exit(3)).

Under Linux 2.0 and earlier /proc/[pid]/exe is a pointer to the binary which was executed, and appears as a symbolic link. A readlink(2) call on this file under Linux 2.0

returns a string in the format:

[device]:inode

For example, [0301]:1502 would be inode 1502 on device major 03 (IDE, MFM, etc. drives) minor 01 (first partition on the first drive).

find(1) with the -inum option can be used to locate the file.

1. Узнайте, какую наиболее старшую версию набора инструкций SSE поддерживает ваш процессор с помощью /proc/cpuinfo.

[vagrant@10 ~]$ cat /proc/cpuinfo | grep "sse"

flags : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 ht syscall nx rdtscp lm constant\_tsc rep\_good nopl xtopology nonstop\_tsc eagerfpu pni pclmulqdq ssse3 cx16 pcid sse4\_1 sse4\_2 x2apic movbe popcnt aes xsave avx rdrand hypervisor lahf\_lm abm 3dnowprefetch invpcid\_single fsgsbase avx2 invpcid rdseed clflushopt md\_clear flush\_l1d

flags : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 ht syscall nx rdtscp lm constant\_tsc rep\_good nopl xtopology nonstop\_tsc eagerfpu pni pclmulqdq ssse3 cx16 pcid sse4\_1 sse4\_2 x2apic movbe popcnt aes xsave avx rdrand hypervisor lahf\_lm abm 3dnowprefetch invpcid\_single fsgsbase avx2 invpcid rdseed clflushopt md\_clear flush\_l1d

Видимо, sse4\_2

1. При открытии нового окна терминала и vagrant ssh создается новая сессия и выделяется pty. Это можно подтвердить командой tty, которая упоминалась в лекции 3.2. Однако:

vagrant@netology1:~$ ssh localhost 'tty'

not a tty

Почитайте, почему так происходит, и как изменить поведение.

[vagrant@10 ~]$ ssh localhost 'tty'

The authenticity of host 'localhost (::1)' can't be established.

ECDSA key fingerprint is SHA256:+CsZ6C0ERAmopvD7kJxT1XO5u/vDKF1+SDVgtffY8tY.

ECDSA key fingerprint is MD5:aa:b0:3d:5b:c4:37:b0:a5:11:e5:af:cb:3e:b6:2c:18.

Are you sure you want to continue connecting (yes/no)? yes

Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.

Permission denied (publickey,gssapi-keyex,gssapi-with-mic).

Это странно, но у меня сразу вот такой вывод (((

1. Бывает, что есть необходимость переместить запущенный процесс из одной сессии в другую. Попробуйте сделать это, воспользовавшись reptyr. Например, так можно перенести в screen процесс, который вы запустили по ошибке в обычной SSH-сессии.

Почему-то несмотря на все усилия получал либо:

vagrant@vagrant:~$ sudo reptyr 2390

[-] Unable to open the tty in the child.

Unable to attach to pid 2390: Permission denied

Либо:

vagrant@vagrant:~$ sudo reptyr -T 2208

[-] Unable to find the fd for the pty!

Unable to attach to pid 2208: No such process

Либо:

vagrant@vagrant:~$ reptyr 2189

Unable to attach to pid 2189: Operation not permitted

The kernel denied permission while attaching. If your uid matches

the target's, check the value of /proc/sys/kernel/yama/ptrace\_scope.

For more information, see /etc/sysctl.d/10-ptrace.conf

1. sudo echo string > /root/new\_file не даст выполнить перенаправление под обычным пользователем, так как перенаправлением занимается процесс shell'а, который запущен без sudo под вашим пользователем. Для решения данной проблемы можно использовать конструкцию echo string | sudo tee /root/new\_file. Узнайте что делает команда tee и почему в отличие от sudo echo команда с sudo tee будет работать.

Это не работает так как перенаправление выполняется оболочкой а не командой, которой оно применяется. А оболочка работает от имени рут. Команда tee позволяет использовать стандатный вывод, а не рутовый.