1. Какой системный вызов делает команда cd? В прошлом ДЗ мы выяснили, что cd не является самостоятельной программой, это shell builtin, поэтому запустить strace непосредственно на cd не получится. Тем не менее, вы можете запустить strace на /bin/bash -c 'cd /tmp'. В этом случае вы увидите полный список системных вызовов, которые делает сам bash при старте. Вам нужно найти тот единственный, который относится именно к cd.

Ответ: chdir(“ /tmp”)

2. Попробуйте использовать команду file на объекты разных типов на файловой системе. Например:

vagrant@netology1:~$ file /dev/tty

/dev/tty: character special (5/0)

vagrant@netology1:~$ file /dev/sda

/dev/sda: block special (8/0)

vagrant@netology1:~$ file /bin/bash

/bin/bash: ELF 64-bit LSB shared object, x86-64

Используя strace выясните, где находится база данных file на основании которой она делает свои догадки.

Ответ: /usr/share/misc/magic.mcg

3. Предположим, приложение пишет лог в текстовый файл. Этот файл оказался удален (deleted в lsof), однако возможности сигналом сказать приложению переоткрыть файлы или просто перезапустить приложение – нет. Так как приложение продолжает писать в удаленный файл, место на диске постепенно заканчивается. Основываясь на знаниях о перенаправлении потоков предложите способ обнуления открытого удаленного файла (чтобы освободить место на файловой системе).

Перенаправить поток в dev/null

kill -9 pid процесса.

4. Занимают ли зомби-процессы какие-то ресурсы в ОС (CPU, RAM, IO)?

Ответ: Это просто записи в таблице процессов ядра. Это единственный ресурс, который они потребляют. Они не потребляют ни CPU, ни RAM. Единственная опасность наличия зомби - это нехватка места в таблице процессов

5. В iovisor BCC есть утилита opensnoop:

root@vagrant:~# dpkg -L bpfcc-tools | grep sbin/opensnoop

/usr/sbin/opensnoop-bpfcc

На какие файлы вы увидели вызовы группы open за первую секунду работы утилиты? Воспользуйтесь пакетом bpfcc-tools для Ubuntu 20.04. Дополнительные [сведения по установке](https://github.com/iovisor/bcc/blob/master/INSTALL.md).

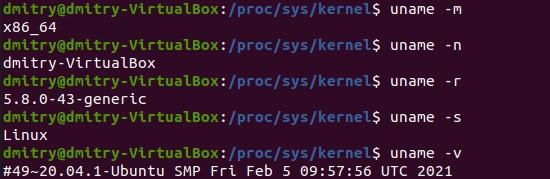
**Ответ: sudo /usr/share/bcc/tools/opensnoop**

6. Какой системный вызов использует uname -a? Приведите цитату из man по этому системному вызову, где описывается альтернативное местоположение в /proc, где можно узнать версию ядра и релиз ОС.

6.1. Какой системный вызов использует uname -a? Ответ: uname использует системный вызов uname(2)для получения информации, относящейся к ядру.

6.2

|  |  |  |  |
| --- | --- | --- | --- |
| uname | Обозначение | Вывод | альтернатива /proc |
| -m | Тип компьютера | X86\_64 /  в domainname вывод (none) | /proc/sys/kernel/domainname |
| -n | Узел сети | Dmitry-VirtualBox | /proc/sys/kernel/hostname |
| -r | Номер редакции | 5.8.0.43-generic | /proc/sys/kernel/osrelease |
| -s | Название операционной системы | Linux | /proc/sys/kernel/ostype |
| -v | Номер версии операционной системы | #49~20.04.1-Ubuntu SMP Fri Feb 5 09:57:56 UTC 2021 | /proc/sys/kernel/version |



7.1. Чем отличается последовательность команд через ; и через && в bash? Например:

1. root@netology1:~# test -d /tmp/some\_dir; echo Hi
2. Hi
3. root@netology1:~# test -d /tmp/some\_dir && echo Hi

root@netology1:~#

; (точка с запятой). При таком подходе вы можете ввести несколько команд в одной строке. При этом, в отличие от обычного ввода команд, для выполнения следующей команды не надо ждать завершения предыдущей.

разделитель &&. Этот разделитель позволяет гарантировать то, что следующая команда будет выполнена лишь в том случае, если предыдущая отработает успешно.

7.2 Есть ли смысл использовать в bash &&, если применить set -e?

-e Exit immediately if a command exits with a non-zero status.

-e Немедленный выход, если команда завершается с ненулевым статусом.

This is not done in situations, where the exit code is already checked (if, while, until,||, &&)

Этого не происходит(не делается) в ситуациях, когда код выхода уже проверен (если, while, until, ||, &&)

8. Из каких опций состоит режим bash set -euxo pipefail и почему его хорошо было бы использовать в сценариях?

set -e Немедленный выход, если команда завершается с ненулевым статусом.

set -u ссылка на любую переменную, которую ранее не определяли будет являться ошибкой.

set -x включает режим оболочки, в котором все выполняемые команды выводятся на терминал.

set -o pipefail Этот параметр предотвращает маскировку ошибок в конвейере. Если какая-либо команда в конвейере не работает, этот код возврата будет использоваться как код возврата всего конвейера. По умолчанию код возврата конвейера - это код последней команды, даже если она выполнена успешно.

9. Используя -o stat для ps, определите, какой наиболее часто встречающийся статус у процессов в системе. В man ps ознакомьтесь (/PROCESS STATE CODES) что значат дополнительные к основной заглавной буквы статуса процессов. Его можно не учитывать при расчете (считать S, Ss или Ssl равнозначными).

9.1.Используя -o stat для ps, определите, какой наиболее часто встречающийся статус у процессов в системе.

Ответ: наиболее часто встречающийся статус у процессов в системе: T – остановлен сигналом управления задачами