

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
```

ОТВЕТ:

```
strace -e trace=openat file /dev/tty
```

```
openat(AT_FDCWD, "/usr/share/misc/magic.mgc",
O_RDONLY) = 3
```

база находится в `/usr/share/misc/magic.mgc`

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

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).

ОТВЕТ:

```
ping google.com > /tmp/test_ping &
```

```
/usr/bin/rm /tmp/test_ping
```

```
ps -o pid,cmd|grep ping
```

```
39729 ping google.com
```

```
sudo lsof -p 39729|grep test_ping
```

```
ping 39729 alexey 1w REG 8,1 44927 1585443 /tmp/test_ping
(deleted)
```

```
sudo bash -c 'echo > /proc/39729/fd/1'
```

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

ОТВЕТ:

Зомби не занимают ресурсы, но блокируют записи в таблице процессов, размер которой ограничен для каждого пользователя и системы в целом

5. В `iovisor BCC` есть утилита `opensnoop`:

```
root@vagrant:~$ dpkg -L bpfcc-tools | grep sbin/opensnoop
/usr/sbin/opensnoop-bpffcc
```

На какие файлы вы увидели вызовы группы `open` за первую секунду работы утилиты? Воспользуйтесь пакетом `bpffcc-tools` для Ubuntu 20.04. Дополнительные [сведения по установке](#).

ОТВЕТ:

```
root@vagrant:/home/vagrant# opensnoop-bpffcc -d 1
```

PID	COMM	FD	ERR	PATH
915	vminfo	4	0	/var/run/utmp
663	dbus-daemon	-1	2	/usr/local/share/dbus-1/system-services
663	dbus-daemon	19	0	/usr/share/dbus-1/system-services
663	dbus-daemon	-1	2	/lib/dbus-1/system-services
663	dbus-daemon	19	0	/var/lib/snapd/dbus-1/system-services/
915	vminfo	4	0	/var/run/utmp

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.

Ответ:

`man 2 uname`

Part of the utsname information is also accessible via `/proc/sys/kernel/{ostype, hostname, osrelease, version, domainname}`.

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

```
root@netology1:~# test -d /tmp/some_dir; echo Hi
Hi
root@netology1:~# test -d /tmp/some_dir && echo Hi
root@netology1:~#
```

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

Ответ:

Последовательность команд через `;` предполагает их последовательное исполнение независимо от результата предыдущей, через `&&` предполагает, что предыдущая команда закончилась с выходом 0.

Команду `ping -c 1 netology.ru && echo ok`

можно заменить командой с `set -e`

`bash -c 'set -e; ping -c 1 netology.ru; echo ok'`

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

Ответ:

`-e` выйти немедленно, но проверятся код ошибки последней команды в пайпе

`-u` проверяет инициализацию переменных и параметры в скрипте.

`-x` `bash` печатает в стандартный вывод все команды перед их исполнением.

`pipefail` все команды в пайпах завершились успешно

Указанные опции позволяют более тонко контролировать исполнение сценариев.

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).

Ответ:

```
ps -ax -o stat|cut -c1-1 |sort |uniq -c|sort -n
```

1 D

2 R

137 I

338 S - interruptible sleep (waiting for an event to complete)