

Выполнение домашнего задания "Процессы Linux"

Задание

При помощи `/proc/` `/sys/fs` собрать информацию:

Шаг 1

- О текущем процессе:

- Кол-во дескрипторов
- Потребление памяти
- Путь до исполняемого файла

- О текущей системе:

- Какой процессор (сколько)
- Какая память (сколько)

- Вывести все в Json

Шаг 2

Запустить приложение из шага 1 в изоляции при помощи например Docker

*Advanced level запустить руками в `cgroup` и `namespace`

Сравнить вывод Json шага 1 и шага 2

Решение

Сбор информации о текущем процессе

Для сбора создадим скрипт

```
nano collecting_information_about_the_current_process.sh
```

```
#!/bin/bash
```

```
# Получаем PID текущего процесса
```

```
PID=$$
```

```
# Получаем количество дескрипторов
```

```
FD_COUNT=$(ls /proc/$PID/fd/ | wc -l)
```

```
# Получаем потребление памяти
MEM_USAGE=$(awk '/VmRSS/ {print $2}' /proc/$PID/status)

# Получаем путь до исполняемого файла
EXEC_PATH=$(readlink /proc/$PID/exe)

# Формируем JSON-объект с собранной информацией
json='{ "process_info": {'
json+="\\"fd_count\": \"$FD_COUNT\", "
json+="\\"memory_usage\": \"$MEM_USAGE\", "
json+="\\"executable_path\": \"$EXEC_PATH\""
json+='}}'

echo $json > proc.json
```

Делаем файл исполняемым

```
chmod +x collecting_information_about_the_current_process.sh
```

Запускаем скрипт

```
./collecting_information_about_the_current_process.sh
```

Теперь выводим информацию из файла на экран

```
cat proc.json
```



Сбор информации о текущей системе

Для сбора создадим скрипт

```
nano collecting_information_about_the_current_system.sh
```

```
#!/bin/bash

# Получение информации о процессоре
CPU_INFO=$(lscpu | awk -F':' '{print "\"" $1 "\": \"" $2 "\"\","}'')
CPU_INFO=${CPU_INFO%?} # Удаление последней запятой

# Получение информации о памяти
MEM_INFO=$(free -h | awk 'NR==2{print "\"total_memory\": \""$2 "\"\","
\"used_memory\": \""$3 "\"\"," \"free_memory\": \""$4 "\"\"}')

# Получение информации о дисковом пространстве
DISK_INFO=$(df -h --output=size,used,avail,fstype | awk 'NR>1{print
\"\"filesystem\": \""$4 "\"\"," \"total_size\": \""$1 "\"\"," \"used\": \""$2 "\"\","
\"available\": \""$3 "\"\"}')
```

```
# Формирование JSON-объекта
echo "{\"system_info\": { $CPU_INFO, $MEM_INFO, $DISK_INFO } }" > sys.json
```

Делаем файл исполняемым

```
chmod +x collecting_information_about_the_current_system.sh
```

Запускаем скрипт

```
./collecting_information_about_the_current_system.sh
```

Теперь выводим информацию из файла на экран

```
cat sys.json
```



Запуск приложения из шага 1

В моем случае это оболочка **bash** Для ее запуска в докере потребуется создать **Dockerfile** с содержимым

```
FROM ubuntu:latest

# Установка необходимых пакетов
RUN apt-get update && apt-get install -y bash

# Копирование скрипта в рабочую директорию образа
COPY collecting_information_about_the_current_process.sh
/home/collecting_information_about_the_current_process.sh
COPY collecting_information_about_the_current_system.sh
/home/collecting_information_about_the_current_system.sh

# Установка разрешений для выполнения скрипта
RUN chmod +x /home/collecting_information_about_the_current_process.sh
RUN chmod +x /home/collecting_information_about_the_current_system.sh

# Запуск скрипта при старте контейнера
CMD ["/bin/bash",
"/home/collecting_information_about_the_current_process.sh"]
CMD ["/bin/bash",
"/home/collecting_information_about_the_current_system.sh"]
```

Теперь собираем приложение

```
docker build -t my-bash .
```

Запускаем и сразу попадаем в терминал контейнера

```
docker run -it my-bash /bin/bash
```

Проверяем скопировались ли скрипты

```
ls /home/
```

ВИДИМ ЧТО ОНИ ПОЯВИЛИСЬ



Запуск руками в cgroup

Выполняем команду с нужным вам PID

```
nsenter --target <PID> --uts --ipc --net --pid --mount --cgroup -- /bin/bash
```

Проверяем что вы в cgroup командой

```
lsns
```

From:

<https://wiki.fellk.ru/dokuwiki/> - **Игорь Fellk**

Permanent link:

https://wiki.fellk.ru/dokuwiki/doku.php/devops_training_in_yandex/proc_linux

Last update: **2023/11/13 14:10**

