

Студент: Герлах Глеб

## Задание 1. Systemd

### Решение:

- Создал bash скрипт, соответствующий systemd.service и запустил его

```
[ggerlakh@compute-vm-2-2-20-ssd-1759075627604:~]$ cat /usr/local/bin/homework_service.sh
#!/bin/bash
echo "My custom service has started."
while true; do
    echo "Service heartbeat: $(date)" >> /tmp/homework_service.log
    sleep 15
done
[ggerlakh@compute-vm-2-2-20-ssd-1759075627604:~]$ ls -l /usr/local/bin/homework_service.sh
-rwxr-xr-x 1 root root 146 Oct  8 18:14 /usr/local/bin/homework_service.sh
[ggerlakh@compute-vm-2-2-20-ssd-1759075627604:~]$ cat /etc/systemd/system/homework_service.service
[Unit]
Description=Simple homework service writing date

[Service]
Type=exec
ExecStart=/usr/local/bin/homework_service.sh
Restart=on-failure
[ggerlakh@compute-vm-2-2-20-ssd-1759075627604:~]$ sudo systemctl daemon-reload
[ggerlakh@compute-vm-2-2-20-ssd-1759075627604:~]$ sudo systemctl start homework_service.service
```

- Судя по status, сервис успешно работает

```
Oct 08 18:38:08 compute-vm-2-2-20-ssd-1759075627604 homework_service.sh[196371]: My custom service has started.
[ggerlakh@compute-vm-2-2-20-ssd-1759075627604:~]$ sudo systemctl status homework_service.service
● homework_service.service - Simple homework service writing date
   Loaded: loaded (/etc/systemd/system/homework_service.service; static)
   Active: active (running) since Wed 2025-10-08 18:38:08 UTC; 3s ago
     Main PID: 196371 (homework_service)
        Tasks: 2 (limit: 4654)
      Memory: 596.0K (peak: 844.0K)
         CPU: 6ms
    CGroup: /system.slice/homework_service.service
            └─196371 /bin/bash /usr/local/bin/homework_service.sh
                └─196373 sleep 15

Oct 08 18:38:08 compute-vm-2-2-20-ssd-1759075627604 systemd[1]: Starting homework_service.service - Simple homework service writing date...
Oct 08 18:38:08 compute-vm-2-2-20-ssd-1759075627604 systemd[1]: Started homework_service.service - Simple homework service writing date.
Oct 08 18:38:08 compute-vm-2-2-20-ssd-1759075627604 homework_service.sh[196371]: My custom service has started.
[ggerlakh@compute-vm-2-2-20-ssd-1759075627604:~]$ cat /tmp/homework_service.log
Service heartbeat: Wed Oct  8 18:38:01 UTC 2025
Service heartbeat: Wed Oct  8 18:38:08 UTC 2025
Service heartbeat: Wed Oct  8 18:38:23 UTC 2025
[ggerlakh@compute-vm-2-2-20-ssd-1759075627604:~]$
```

- Пробую запустить сервис снова

```
Service heartbeat: Wed Oct  8 18:45:00 UTC 2025
[ggerlakh@compute-vm-2-2-20-ssd-1759075627604:~]$ sudo systemctl status homework_service.service
● homework_service.service - Simple homework service writing date
   Loaded: loaded (/etc/systemd/system/homework_service.service; static)
   Active: active (running) since Wed 2025-10-08 18:45:51 UTC; 24s ago
     Main PID: 196908 (homework_service)
        Tasks: 2 (limit: 4654)
      Memory: 592.0K (peak: 1.0M)
         CPU: 6ms
    CGroup: /system.slice/homework_service.service
            └─196908 /bin/bash /usr/local/bin/homework_service.sh
                └─196924 sleep 15

Oct 08 18:45:51 compute-vm-2-2-20-ssd-1759075627604 systemd[1]: homework_service.service: Scheduled restart job, restart counter is at 1.
Oct 08 18:45:51 compute-vm-2-2-20-ssd-1759075627604 systemd[1]: Starting homework_service.service - Simple homework service writing date...
Oct 08 18:45:51 compute-vm-2-2-20-ssd-1759075627604 systemd[1]: Started homework_service.service - Simple homework service writing date.
Oct 08 18:45:51 compute-vm-2-2-20-ssd-1759075627604 homework_service.sh[196908]: My custom service has started.
```

- В другом терминале пробую принудительно «убить» процесс, послав SIGKILL

```
kill: (196919): No such process
[ggerlakh@compute-vm-2-2-20-ssd-1759075627604:~]$ sudo kill -9 196908
[ggerlakh@compute-vm-2-2-20-ssd-1759075627604:~]$
```

- Судя по логам, сервис перезапустился

```
Oct 08 18:45:51 compute-vm-2-2-20-ssd-1759075627604 homework_service.sh[196908]: My custom service has started.
ggerlakh@compute-vm-2-2-20-ssd-1759075627604:~$ sudo systemctl status homework_service.service

● homework_service.service - Simple homework service writing date
   Loaded: loaded (/etc/systemd/system/homework_service.service; static)
   Active: active (running) since Wed 2025-10-08 18:46:28 UTC; 7s ago
     Main PID: 196949 (homework_service)
       Tasks: 2 (limit: 4654)
      Memory: 588.0K (peak: 1.0M)
         CPU: 4ms
    CGroup: /system.slice/homework_service.service
            └─196949 /bin/bash /usr/local/bin/homework_service.sh
            └─196951 sleep 15

Oct 08 18:46:28 compute-vm-2-2-20-ssd-1759075627604 systemd[1]: homework_service.service: Scheduled restart job, restart counter is at 2.
Oct 08 18:46:28 compute-vm-2-2-20-ssd-1759075627604 systemd[1]: Starting homework_service.service - Simple homework service writing date...
Oct 08 18:46:28 compute-vm-2-2-20-ssd-1759075627604 systemd[1]: Started homework_service.service - Simple homework service writing date.
Oct 08 18:46:28 compute-vm-2-2-20-ssd-1759075627604 homework_service.sh[196949]: My custom service has started.
ggerlakh@compute-vm-2-2-20-ssd-1759075627604:~$ cat /tmp/homework_service.log
Service heartbeat: Wed Oct 8 18:38:01 UTC 2025
Service heartbeat: Wed Oct 8 18:38:08 UTC 2025
Service heartbeat: Wed Oct 8 18:38:23 UTC 2025
Service heartbeat: Wed Oct 8 18:38:38 UTC 2025
Service heartbeat: Wed Oct 8 18:38:53 UTC 2025
Service heartbeat: Wed Oct 8 18:39:08 UTC 2025
Service heartbeat: Wed Oct 8 18:39:23 UTC 2025
Service heartbeat: Wed Oct 8 18:39:38 UTC 2025
Service heartbeat: Wed Oct 8 18:39:53 UTC 2025
Service heartbeat: Wed Oct 8 18:40:08 UTC 2025
Service heartbeat: Wed Oct 8 18:40:23 UTC 2025
Service heartbeat: Wed Oct 8 18:40:38 UTC 2025
Service heartbeat: Wed Oct 8 18:40:53 UTC 2025
Service heartbeat: Wed Oct 8 18:41:08 UTC 2025
Service heartbeat: Wed Oct 8 18:41:23 UTC 2025
Service heartbeat: Wed Oct 8 18:41:38 UTC 2025
Service heartbeat: Wed Oct 8 18:41:53 UTC 2025
Service heartbeat: Wed Oct 8 18:42:08 UTC 2025
Service heartbeat: Wed Oct 8 18:42:23 UTC 2025
Service heartbeat: Wed Oct 8 18:42:38 UTC 2025
Service heartbeat: Wed Oct 8 18:42:53 UTC 2025
Service heartbeat: Wed Oct 8 18:45:12 UTC 2025
Service heartbeat: Wed Oct 8 18:45:27 UTC 2025
Service heartbeat: Wed Oct 8 18:45:42 UTC 2025
Service heartbeat: Wed Oct 8 18:45:51 UTC 2025
Service heartbeat: Wed Oct 8 18:46:06 UTC 2025
Service heartbeat: Wed Oct 8 18:46:21 UTC 2025
Service heartbeat: Wed Oct 8 18:46:28 UTC 2025
ggerlakh@compute-vm-2-2-20-ssd-1759075627604:~$
```

- Используя systemd-analyze, вывел топ-5 unit'ов стартующих дольше всего

```
ggerlakh@compute-vm-2-2-20-ssd-1759075627604:~$ systemd-analyze blame | head -n 5
15.110s apt-daily.service
5.908s motd-news.service
2.512s cloud-init-local.service
2.453s cloud-init.service
2.277s cloud-config.service
ggerlakh@compute-vm-2-2-20-ssd-1759075627604:~$
```

**Решение:**

- Создал файл с программой на C, создающей шареную память, и запустил ее

```
[ggerlakh@compute-vm-2-2-20-ssd-1759075627604:~/hw3$ cat shm_creator.c
#include <stdio.h>
#include <stdlib.h>
#include <sys/ipc.h>
#include <sys/shm.h>
#include <unistd.h>

int main() {
    key_t key = ftok("homework_key", 65); // Generate a unique key
    int shmid = shmget(key, 1024, 0666|IPC_CREAT); // Create 1KB segment
    if (shmid == -1) {
        perror("shmget");
        exit(1);
    }
    printf("Shared memory segment created.\n");
    printf("ID: %d\nKey: 0x%x\n", shmid, key);
    printf("Run 'ipcs -m' to see it. Process will exit in 60 seconds...\n");
    sleep(60);
    shmctl(shmid, IPC_RMID, NULL); // Clean up
    printf("Shared memory segment removed.\n");
    return 0;
}

[ggerlakh@compute-vm-2-2-20-ssd-1759075627604:~/hw3$ gcc shm_creator.c -o shm_creator
[ggerlakh@compute-vm-2-2-20-ssd-1759075627604:~/hw3$ touch homework_key
[ggerlakh@compute-vm-2-2-20-ssd-1759075627604:~/hw3$ ./shm_creator
Shared memory segment created.
ID: 0
Key: 0x41016658
Run 'ipcs -m' to see it. Process will exit in 60 seconds...
Shared memory segment removed.
[ggerlakh@compute-vm-2-2-20-ssd-1759075627604:~/hw3$ ]
```

```
[ggerlakh@compute-vm-2-2-20-ssd-1759075627604:~/hw3$ ipcs -m

----- Shared Memory Segments -----
key      shmid  owner   perms   bytes   nattch   status
0x41016658 0       ggerlakh 666     1024    0
```

- В соседнем терминале запустил `ipcs -m`  
В котором показаны данные о созданном сегменте `shared_memory`, идентификатор сегмента, `key` — ключ созданного сегмента, права доступа, размер и количество присоединенных к сегменту процессов

### Задание 3. Анализ памяти процессов (VSZ vs RSS)

#### Решение:

- Запустил программу на Python в первом терминале, которая запрашивает 250 MiB памяти и держит ее 2 минуты

```
ggerlakh@compute-vm-2-2-28-ssd-1759875627684:~/hw3$ python3 -c "print('Allocating memory...'); a = 'X' * (250 * 1024 * 1024); import time; print('Memory allocated. Sleeping...'); time.sleep(120);"
Allocating memory...
Memory allocated. Sleeping...
```

- Во втором терминале нашел pid созданного процесса и вывел данные по используемой памяти RSS и VSZ

```
ggerlakh@compute-vm-2-2-28-ssd-1759875627684:~/hw3$ ps auxf | grep python3
root      849  0.0  0.5 118012 22784 ?        Ssl  Oct01   0:00 /usr/bin/python3 /usr/share/unattended-upgrades/unattended-upgrade-shutdown --wait-for-signal
ggerlakh 199735  0.0  0.8 46920 32640 pts/6    S+   19:49   0:00 |      \_ python3
ggerlakh 201225 1.9  6.6 274336 265984 pts/8    S+   20:11   0:00 |      \_ python3 -c print('Allocating memory...'); a = 'X' * (250 * 1024 * 1024); import time; print('Memory allocated. Sleeping...'); time.sleep(120);
ggerlakh 201228  0.0  0.0  7076 2176 pts/9      S+   20:11   0:00 |      \_ grep --color=auto python3
ggerlakh@compute-vm-2-2-28-ssd-1759875627684:~/hw3$ ps -o pid,user,mem,rss,vsz,comm -p 201225
  PID USER      MEM  RSS  VSZ COMMAND
 201225 ggerlakh  6.6 265984 274336 python3
```

- Память RSS не 0, так как в программе была выделена память под строку, но в ней также еще и учитывается исходная память процесса-интерпретатора python  
Память VSZ > RSS, так как еще есть overhead python, связанный с подключаемыми библиотеками и резервом выделяемым под Python heap

### Задание 4. NUMA и cgroups (35 баллов)

## Решение:

- При помощи команды `numactl`, я вывел количество нод на сервере и количество памяти для каждой NUMA ноды

```
ggerlakh@compute-vm-2-2-20-ssd-1759075627604:~$ numactl --hardware
available: 1 nodes (0)
node 0 cpus: 0 1
node 0 size: 3915 MB
node 0 free: 1239 MB
node distances:
node 0
0: 10
ggerlakh@compute-vm-2-2-20-ssd-1759075627604:~$
```

- Запустил команду `systemd-run`

```
ggerlakh@compute-vm-2-2-20-ssd-1759075627604:~/hw3$ sudo systemd-run --unit=homework.service --slice=testing.slice --property="MemoryMax=150M" --property="CPUWeight=100" --cpu 1 --vm 1 --vm-bytes 300M --timeout 90s ./stress
systemd-run: unrecognized option '--cpu'
ggerlakh@compute-vm-2-2-20-ssd-1759075627604:~/hw3$ sudo systemd-run --unit=highload-stress-test --slice=testing.slice --property="MemoryMax=150M" --property="CPUWeight=100" ./stress
Running as unit: highload-stress-test.service; invocation ID: e86d28c37a2442ba7c3880f8cff16e4
ggerlakh@compute-vm-2-2-20-ssd-1759075627604:~/hw3$ sudo systemctl status highload-stress-test
● highload-stress-test.service - /home/ggerlakh/hw3/./stress
   Loaded: loaded (/run/systemd/transient/highload-stress-test.service; transient)
   Transient: yes
   Active: active (running) since Wed 2025-10-08 20:50:54 UTC; 37s ago
     Main PID: 203206 (stress)
       Tasks: 1 (limit: 4654)
      Memory: 176.0K (max: 150.0M available: 149.8M peak: 332.0K)
         CPU: 1ms
        CGroup: /testing.slice/highload-stress-test.service
                └─203206 /home/ggerlakh/hw3/./stress

Oct 08 20:50:54 compute-vm-2-2-20-ssd-1759075627604 systemd[1]: Started highload-stress-test.service - /home/ggerlakh/hw3/./stress.
```

- Будет ли работать тест если мы запрашиваем 300M оперативной памяти, а ограничиваем 150M?  
Не будет, так как из-за ограничений `MemoryMax` при превышении лимита, процесс будет убит OOM killer
- Запустил команду `echo q | systemd-cgtop -p testing.slice` для мониторинга используемых ресурсов

CGroup	Tasks	%CPU	Memory	Input/s	Output/s
testing.slice	1	-	196.0K	-	-
testing.slice/highload-stress-test.service	1	-	184.0K	-	-

```
ggerlakh@compute-vm-2-2-20-ssd-1759075627604:~$
```

- Опишите что делает и для чего можно использовать `MemoryMax` and `CPUWeight`.  
`MemoryMax` позволяет задать «жесткий» лимит по памяти для процесса, используя механизм cgroups. При превышении этого лимита, процесс будет убит OOM killer.  
`CPUWeight` можно использовать для пропорциональным распределением времени CPU между конкурирующими systemd сервисами.