

Лабораторная работа №4.

Работа с systemd.

Выполнил: Михалев Никита Романович М3311.

Цель работы:

получить практические навыки по управлению ОС Linux с помощью systemd.

Часть 1. Получение информации о времени загрузки:

1. Выведите информацию о времени, затраченном на загрузку системы

```
vr61v@bad:~$ systemd-analyze time
```

```
Startup finished in 3.111s (kernel) + 4.513s (userspace) = 7.624s
graphical.target reached after 4.485s in userspace.
```

2. Выведите список всех запущенных при старте системы сервисов, в порядке уменьшения времени, затраченного на загрузку сервиса.

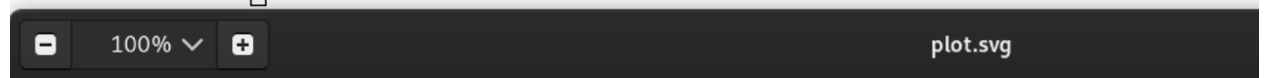
```
vr61v@bad:~/lab4$ systemd-analyze blame
2.136s plymouth-quit-wait.service
1.993s systemd-random-seed.service
951ms udisks2.service
889ms e2scrub_reap.service
825ms accounts-daemon.service
821ms avahi-daemon.service
820ms polkit.service
820ms power-profiles-daemon.service
818ms switcheroo-control.service
816ms systemd-logind.service
809ms dbus.service
797ms logrotate.service
625ms fstrim.service
239ms dev-sda2.device
161ms apparmor.service
133ms keyboard-setup.service
129ms systemd-udev-trigger.service
119ms networking.service
83ms NetworkManager-wait-online.service
68ms user@1000.service
67ms ModemManager.service
64ms systemd-timesyncd.service
60ms systemd-fsck@dev-disk-by\x2duuid-E295\x2d1AF5.service
59ms systemd-udevd.service
54ms systemd-tmpfiles-clean.service
47ms NetworkManager.service
44ms colord.service
40ms systemd-journald.service
28ms gdm.service
27ms geoclue.service
27ms systemd-tmpfiles-setup.service
25ms plymouth-start.service
23ms wpa_supplicant.service
23ms upower.service
23ms systemd-journal-flush.service
20ms cups.service
20ms modprobe@fuse.service
20ms systemd-tmpfiles-setup-dev.service
19ms sys-kernel-tracing.mount
18ms kmod-static-nodes.service
18ms systemd-remount-fs.service
18ms dev-mqueue.mount
18ms modprobe@configfs.service
18ms sys-kernel-debug.mount
18ms dev-hugepages.mount
17ms ifupdown-pre.service
17ms modprobe@drm.service
17ms systemd-binfmt.service
17ms systemd-sysusers.service
15ms systemd-modules-load.service
15ms systemd-sysctl.service
14ms dev-disk-by\x2duuid-3425604b\x2ded5b\x2d4325\x2dbba2\x2d4af014b5941b.swap
14ms console-setup.service
12ms plymouth-read-write.service
10ms proc-sys-fs-binfmt_misc.mount
10ms sys-kernel-config.mount
9ms boot-efi.mount
8ms sys-fs-fuse-connections.mount
7ms user-runtime-dir@1000.service
7ms systemd-user-sessions.service
6ms systemd-update-utmp.service
5ms alsa-restore.service
4ms modprobe@efi_pstore.service
4ms e2scrub_all.service
4ms modprobe@loop.service
3ms systemd-update-utmp-runlevel.service
2ms modprobe@dm_mod.service
2ms rtkit-daemon.service
```

3. Выведите список сервисов, запуск которых с необходимостью предшествовал запуску сервиса sshd.

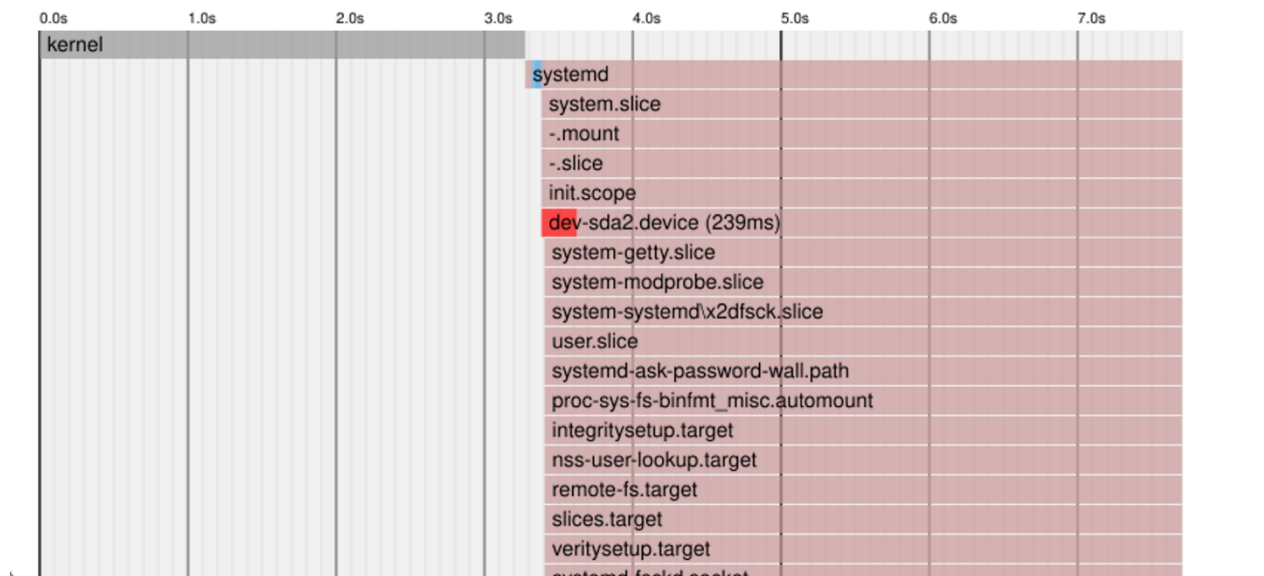
```
vr61v@bad:~/lab4$ systemctl list-dependencies --before sshd.service
sshd.service
```

4. Сформируйте изображение в формате svg с графиком загрузки системы, сохраните его в файл.

```
vr61v@bad:~/lab4$ cat open-plot.sh
#!/bin/bash
systemd-analyze plot > plot.svg
xdg-open "plot.svg"
vr61v@bad:~/lab4$ bash open-plot.sh
vr61v@bad:~/lab4$
```



Debian GNU/Linux 12 (bookworm) bad (Linux 6.1.0-30-arm64 #1 SMP Debian 6.1.124-1 (2025-01-12)) arm64 parallels
Startup finished in 3.274s (kernel) + 4.429s (userspace) = 7.704s graphical.target reached after 4.398s in userspace.



Часть 2. Управление юнитами:

1. Получите список всех запущенных юнитов сервисов

```
vr61v@bad:~/lab4$ systemctl list-units --type=service --state=running
```

UNIT	LOAD	ACTIVE	SUB	DESCRIPTION
accounts-daemon.service	loaded	active	running	Accounts Service
avahi-daemon.service	loaded	active	running	Avahi mDNS/DNS-SD Stack
colord.service	loaded	active	running	Manage, Install and Generate Color Profiles
cron.service	loaded	active	running	Regular background program processing daemon
cups-browsed.service	loaded	active	running	Make remote CUPS printers available locally
cups.service	loaded	active	running	CUPS Scheduler
dbus.service	loaded	active	running	D-Bus System Message Bus
gdm.service	loaded	active	running	GNOME Display Manager
geoclue.service	loaded	active	running	Location Lookup Service
low-memory-monitor.service	loaded	active	running	Low Memory Monitor
ModemManager.service	loaded	active	running	Modem Manager
NetworkManager.service	loaded	active	running	Network Manager
polkit.service	loaded	active	running	Authorization Manager
power-profiles-daemon.service	loaded	active	running	Power Profiles daemon
rtkit-daemon.service	loaded	active	running	RealtimeKit Scheduling Policy Service
switcheroo-control.service	loaded	active	running	Switcheroo Control Proxy service
systemd-journald.service	loaded	active	running	Journal Service
systemd-logind.service	loaded	active	running	User Login Management
systemd-timesyncd.service	loaded	active	running	Network Time Synchronization
systemd-udev.service	loaded	active	running	Rule-based Manager for Device Events and Files
udisks2.service	loaded	active	running	Disk Manager
upower.service	loaded	active	running	Daemon for power management
user@1000.service	loaded	active	running	User Manager for UID 1000
wpa_supplicant.service	loaded	active	running	WPA supplicant

LOAD = Reflects whether the unit definition was properly loaded.
ACTIVE = The high-level unit activation state, i.e. generalization of SUB.
SUB = The low-level unit activation state, values depend on unit type.
24 loaded units listed.

2. Выведите перечень всех юнитов сервисов, для которых назначена автозагрузка.

```
vr61v@bad:~/lab4$ systemctl list-unit-files --type=service --state=enabled
```

UNIT FILE	STATE	PRESET
accounts-daemon.service	enabled	enabled
anacron.service	enabled	enabled
apparmor.service	enabled	enabled
avahi-daemon.service	enabled	enabled
bluetooth.service	enabled	enabled
console-setup.service	enabled	enabled
cron.service	enabled	enabled
cups-browsed.service	enabled	enabled
cups.service	enabled	enabled
e2scrub_reap.service	enabled	enabled
getty@.service	enabled	enabled
keyboard-setup.service	enabled	enabled
low-memory-monitor.service	enabled	enabled
ModemManager.service	enabled	enabled
networking.service	enabled	enabled
NetworkManager-dispatcher.service	enabled	enabled
NetworkManager-wait-online.service	enabled	enabled
NetworkManager.service	enabled	enabled
power-profiles-daemon.service	enabled	enabled
switcheroo-control.service	enabled	enabled
systemd-pstore.service	enabled	enabled
systemd-timesyncd.service	enabled	enabled
udisks2.service	enabled	enabled
wpa_supplicant.service	enabled	enabled

24 unit files listed.

3. Определите от каких юнитов зависит сервис sshd.

```
vr61v@bad:~/lab4$ systemctl list-dependencies sshd.service
sshd.service
```

4. Определите запущен ли сервис cron, если нет, запустите его.

```
vr61v@bad:~/lab4$ systemctl is-active cron
```

active

```
vr61v@bad:~/lab4$ systemctl status cron
```

```
● cron.service - Regular background program processing daemon
   Loaded: loaded (/lib/systemd/system/cron.service; enabled; preset: enabled)
   Active: active (running) since Wed 2025-04-02 05:46:47 EDT; 49min ago
     Docs: man:cron(8)
  Main PID: 468 (cron)
    Tasks: 1 (limit: 2235)
   Memory: 1.3M
      CPU: 50ms
   CGroup: /system.slice/cron.service
           └─468 /usr/sbin/cron -f
```

Warning: some journal files were not opened due to insufficient permissions.

5. Выведите все параметры юнита cron, даже те, которые были назначены автоматически, и не были прописаны в файле юнита.

```
vr61v@bad:~/lab4$ systemctl show cron
Type=simple
ExitType=main
Restart=on-failure
NotifyAccess=none
RestartUSec=100ms
TimeoutStartUSec=1min 30s
TimeoutStopUSec=1min 30s
TimeoutAbortUSec=1min 30s
TimeoutStartFailureMode=terminate
TimeoutStopFailureMode=terminate
RuntimeMaxUSec=infinity
RuntimeRandomizedExtraUSec=0
WatchdogUSec=0
WatchdogTimestampMonotonic=0
RootDirectoryStartOnly=no
RemainAfterExit=no
GuessMainPID=yes
MainPID=468
ControlPID=0
FileDescriptorStoreMax=0
NFileDescriptorStore=0
StatusErrno=0
Result=succes
ReloadResult=succes
CleanResult=succes
UID=[not set]
GID=[not set]
NRestarts=0
OOMPolicy=stop
ExecMainStartTimestamp=Wed 2025-04-02 05:46:47 EDT
ExecMainStartTimestampMonotonic=4635670
ExecMainExitTimestampMonotonic=0
ExecMainPID=468
ExecMainCode=0
ExecMainStatus=0
ExecStart={ path=/usr/sbin/cron ; argv[]=/usr/sbin/cron -f $EXTRA_OPTS ; ignore_errors=no ; start_time=[Wed 2025-04-02 05:46:47 EDT] ; stop_time=[n/a] ; pid=468 ; code=(null) ; status=0/0 }
ExecStartEx={ path=/usr/sbin/cron ; argv[]=/usr/sbin/cron -f $EXTRA_OPTS ; flags= ; start_time=[Wed 2025-04-02 05:46:47 EDT] ; stop_time=[n/a] ; pid=468 ; code=(null) ; status=0/0 }
Slice=system.slice
ControlGroup=/system.slice/cron.service
ControlGroupID=2029
MemoryCurrent=1400832
MemoryAvailable=infinity
CPUUsageNSec=51593000
TasksCurrent=1
IPIngressBytes=[no data]
IPIngressPackets=[no data]
IPEgressBytes=[no data]
IPEgressPackets=[no data]
IOReadBytes=18446744073709551615
IOReadOperations=18446744073709551615
IOWriteBytes=18446744073709551615
IOWriteOperations=18446744073709551615
```

6. Запретите автозагрузку сервиса cron, но оставьте ему возможность запускаться по зависимостям.

```
root@bad:/home/vr61v/lab4# systemctl disable cron
Synchronizing state of cron.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install disable cron
root@bad:/home/vr61v/lab4# systemctl status cron
○ cron.service - Regular background program processing daemon
   Loaded: loaded (/lib/systemd/system/cron.service; disabled; preset: enabled)
   Active: inactive (dead)
     Docs: man:cron(8)

Apr 02 05:46:47 bad cron[468]: (CRON) INFO (pidfile fd = 3)
Apr 02 05:46:47 bad cron[468]: (CRON) INFO (Running @reboot jobs)
Apr 02 06:17:01 bad CRON[3281]: pam_unix(cron:session): session opened for user root(uid=0) by (uid=0)
Apr 02 06:17:01 bad CRON[3282]: (root) CMD (cd / && run-parts --report /etc/cron.hourly)
Apr 02 06:17:01 bad CRON[3281]: pam_unix(cron:session): session closed for user root
Apr 02 06:25:01 bad CRON[3382]: pam_unix(cron:session): session opened for user root(uid=0) by (uid=0)
Apr 02 06:25:01 bad CRON[3382]: pam_unix(cron:session): session closed for user root
Apr 02 06:43:42 bad systemd[1]: Stopping cron.service - Regular background program processing daemon...
Apr 02 06:43:42 bad systemd[1]: cron.service: Deactivated successfully.
Apr 02 06:43:42 bad systemd[1]: Stopped cron.service - Regular background program processing daemon.
```

Часть 3. Создание сервиса:

1. Создайте собственный сервис mymsg. Сервис mymsg должен:
 - a. при старте системы записывать в системный журнал дату и время
 - b. должен запускаться только если запущен сервис network.
2. Настройте автоматический запуск сервиса mymsg при старте системы.
3. Запустите сервис.

```

root@bad:/home/vr61v/lab4# nano /etc/systemd/system/mymsg.service
root@bad:/home/vr61v/lab4# systemd-analyze verify /etc/systemd/system/mymsg.service
root@bad:/home/vr61v/lab4# systemctl enable mymsg.service
Created symlink /etc/systemd/system/multi-user.target.wants/mymsg.service → /etc/systemd/system/mymsg.service.
root@bad:/home/vr61v/lab4# systemctl start mymsg.service
root@bad:/home/vr61v/lab4# systemctl status mymsg.service
• mymsg.service - mymsg_service
  Loaded: loaded (/etc/systemd/system/mymsg.service; enabled; preset: enabled)
  Active: active (exited) since Wed 2025-04-02 07:10:07 EDT; 6s ago
  Process: 5604 ExecStart=/bin/bash -c logger -t mymsg "mymsg service started" (code=exited, status=0/SUCCESS)
  Main PID: 5604 (code=exited, status=0/SUCCESS)
  CPU: 6ms

Apr 02 07:10:07 bad systemd[1]: Starting mymsg.service - mymsg_service...
Apr 02 07:10:07 bad mymsg[5604]: mymsg service started
Apr 02 07:10:07 bad systemd[1]: Finished mymsg.service - mymsg_service.
root@bad:/home/vr61v/lab4# cat /etc/systemd/system/mymsg.service
[Unit]
Description=mymsg_service
After=network.target
Requires=network.target
[Service]
Type=oneshot
ExecStart=/bin/bash -c 'logger -t mymsg "mymsg service started"'
RemainAfterExit=yes
[Install]
WantedBy=multi-user.target

root@bad:/home/vr61v/lab4# journalctl -t mymsg
Apr 02 07:10:07 bad mymsg[5604]: mymsg service started
root@bad:/home/vr61v/lab4# systemctl list-dependencies mymsg.service
mymsg.service
• └─system.slice
• └─network.target

```

Часть 4. Работа с системным журналом:

1. Выведите на консоль системный журнал. Убедитесь, что сервис mymsg отработал корректно.


```

root@bad:/home/vr61v# journalctl
Feb 07 10:00:50 bad kernel: Booting Linux on physical CPU 0x0000000000 [0x410f0000]
Feb 07 10:00:50 bad kernel: Linux version 6.1.0-30-arm64 (debian-kernel@lists.debian.org) (gcc-12 (Debian 12.2.0-14) 12.2.0, GNU ld (GNU Binutils for Debian) 2.40) #1 SMP Debian 6.1.124-1 (2025-01-12)
Feb 07 10:00:50 bad kernel: efi: EFI v2.70 by EDK II
Feb 07 10:00:50 bad kernel: efi: SMBIOS 3.0=0xb7b0000 ACPI 2.0=0xbce0000 MEMATTR=0xbe1d418 MOKvar=0xbbe0000 MEMRESERVE=0xbbe16098
Feb 07 10:00:50 bad kernel: secureboot: Secure boot disabled
Feb 07 10:00:50 bad kernel: ACPI: Early table checksum verification disabled
Feb 07 10:00:50 bad kernel: ACPI: RSDP 0x0000000000000000 000024 (v02 PRLS )
Feb 07 10:00:50 bad kernel: ACPI: XSDT 0x0000000000000000 000044 (v01 PRLS PRLS_OEM 00000002 01000013)
Feb 07 10:00:50 bad kernel: ACPI: FACP 0x0000000000000000 000114 (v06 PRLS PRLS_OEM 00000002 PRLS 00000002)
Feb 07 10:00:50 bad kernel: ACPI: DSDT 0x0000000000000000 0010D4 (v02 PRLS PRLS_OEM 00000003 INTL 20160527)
Feb 07 10:00:50 bad kernel: ACPI: GTDT 0x0000000000000000 000050 (v01 PRLS PRLS_OEM 00000002 PRLS 00000002)
Feb 07 10:00:50 bad kernel: ACPI: MCFG 0x0000000000000000 00003C (v01 00000001 00000000)
Feb 07 10:00:50 bad kernel: ACPI: APIC 0x0000000000000000 00010C (v04 PRLS PRLS_OEM 00000002 PRLS 00000002)
Feb 07 10:00:50 bad kernel: NUMA: Failed to initialise from firmware
Feb 07 10:00:50 bad kernel: NUMA: Faking a node at [mem 0x0000000040000000-0x00000000bfffffff]
Feb 07 10:00:50 bad kernel: NUMA: NODE_DATA [mem 0xbfbf380-0xbfbfafff]
Feb 07 10:00:50 bad kernel: Zone ranges:
Feb 07 10:00:50 bad kernel: DMA [mem 0x0000000040000000-0x00000000bfffffff]
Feb 07 10:00:50 bad kernel: DMA32 empty
Feb 07 10:00:50 bad kernel: Normal empty
Feb 07 10:00:50 bad kernel: Movable zone start for each node
Feb 07 10:00:50 bad kernel: Early memory node ranges
Feb 07 10:00:50 bad kernel: node 0: [mem 0x0000000040000000-0x00000000bfffffff]
Feb 07 10:00:50 bad kernel: node 0: [mem 0x00000000bbe00000-0x00000000bbe0ffff]
Feb 07 10:00:50 bad kernel: node 0: [mem 0x00000000bbe10000-0x00000000bbeeffff]
Feb 07 10:00:50 bad kernel: node 0: [mem 0x00000000bbef0000-0x00000000bbef8fff]
Feb 07 10:00:50 bad kernel: node 0: [mem 0x00000000bbf90000-0x00000000bbfbffff]
Feb 07 10:00:50 bad kernel: node 0: [mem 0x00000000bbfc0000-0x00000000bbfbffff]
Feb 07 10:00:50 bad kernel: node 0: [mem 0x00000000bc000000-0x00000000bc03ffff]
Feb 07 10:00:50 bad kernel: node 0: [mem 0x00000000bc040000-0x00000000bc0cffff]
Feb 07 10:00:50 bad kernel: node 0: [mem 0x00000000bc0d0000-0x00000000bc0fbfff]
Feb 07 10:00:50 bad kernel: node 0: [mem 0x00000000bc0f5c0000-0x00000000bc0f99ffff]
Feb 07 10:00:50 bad kernel: node 0: [mem 0x00000000bc0f9a0000-0x00000000bfffffff]
Feb 07 10:00:50 bad kernel: Initmem setup node 0 [mem 0x0000000040000000-0x00000000bfffffff]
Feb 07 10:00:50 bad kernel: cma: Reserved 64 MiB at 0x00000000b5e00000
Feb 07 10:00:50 bad kernel: psci: probing for conduit method from ACPI.
Feb 07 10:00:50 bad kernel: psci: PSCIv1.0 detected in firmware.
Feb 07 10:00:50 bad kernel: psci: Using standard PSCI v0.2 function IDs
Feb 07 10:00:50 bad kernel: psci: Trusted OS migration not required
Feb 07 10:00:50 bad kernel: psci: SMC Calling Convention v1.0
Feb 07 10:00:50 bad kernel: percpu: Embedded 31 pages/cpu s86632 r8192 d32152 u126976
Feb 07 10:00:50 bad kernel: pcpu-alloc: s86632 r8192 d32152 u126976 alloc=31*4096
Feb 07 10:00:50 bad kernel: pcpu-alloc: [0] 0 [0] 1
Feb 07 10:00:50 bad kernel: Detected PIPT I-cache on CPU0
Feb 07 10:00:50 bad kernel: CPU features: detected: Address authentication (IMP DEF algorithm)
Feb 07 10:00:50 bad kernel: CPU features: detected: Spectre-v4
Feb 07 10:00:50 bad kernel: alternatives: applying boot alternatives
Feb 07 10:00:50 bad kernel: fallback order for Node 0: 0
Feb 07 10:00:50 bad kernel: Built 1 zonelists, mobility grouping on. Total pages: 516096
Feb 07 10:00:50 bad kernel: Policy zone: DMA
Feb 07 10:00:50 bad kernel: Kernel command line: BOOT_IMAGE=/boot/vmlinuz-6.1.0-30-arm64 root=UUID=f18d2a0e-8ee5-42a9-8ec8-da6a163c9784 ro quiet
Feb 07 10:00:50 bad kernel: Unknown kernel command line parameters "BOOT_IMAGE=/boot/vmlinuz-6.1.0-30-arm64", will be passed to user space.

```

```

root@bad:/home/vr61v# journalctl -t mymsg
Apr 02 07:10:07 bad mymsg[5604]: mymsg service started
-- Boot 96bd697d39ef41899bab5d272c040025 --
Apr 02 07:13:25 bad mymsg[548]: mymsg service started

```

2. Выведите на консоль все сообщения системного журнала, касающиеся сервиса mymsg.

```

root@bad:/home/vr61v# journalctl -u mymsg.service
Apr 02 07:10:07 bad systemd[1]: Starting mymsg.service - mymsg_service...
Apr 02 07:10:07 bad mymsg[5604]: mymsg service started
Apr 02 07:10:07 bad systemd[1]: Finished mymsg.service - mymsg_service.
Apr 02 07:13:13 bad systemd[1]: mymsg.service: Deactivated successfully.
Apr 02 07:13:13 bad systemd[1]: Stopped mymsg.service - mymsg_service.
-- Boot 96bd697d39ef41899bab5d272c040025 --
Apr 02 07:13:25 bad systemd[1]: Starting mymsg.service - mymsg_service...
Apr 02 07:13:25 bad mymsg[548]: mymsg service started
Apr 02 07:13:25 bad systemd[1]: Finished mymsg.service - mymsg_service.

```

3. Выведите на экран все сообщения об ошибках в журнале.

```

root@bad:/home/vr61v# journalctl -p err
Feb 07 10:01:02 bad gdm-password[1050]: gkr-pam: unable to locate daemon control file
Feb 07 10:01:04 bad gdm-launch-environment[594]: GLib-GObject: g_object_unref: assertion 'G_IS_OBJECT (object)' failed
-- Boot 8d0558837857453e8ddc5493af33cf67 --
Feb 09 07:29:50 bad gdm-password[1034]: gkr-pam: unable to locate daemon control file
Feb 09 07:29:53 bad gdm-launch-environment[605]: GLib-GObject: g_object_unref: assertion 'G_IS_OBJECT (object)' failed
-- Boot 40f2fac4624f46698cd44b598acde94f --
Apr 02 05:53:41 bad gdm-password[1071]: gkr-pam: unable to locate daemon control file
Apr 02 05:53:45 bad gdm-launch-environment[580]: GLib-GObject: g_object_unref: assertion 'G_IS_OBJECT (object)' failed
Apr 02 06:48:32 bad systemd[1]: Failed to start fwupd-refresh.service - Refresh fwupd metadata and update motd.
-- Boot 96bd697d39ef41899bab5d272c040025 --
Apr 02 07:13:34 bad gdm-password[1033]: gkr-pam: unable to locate daemon control file
Apr 02 07:13:34 bad systemd[1047]: Failed to start app-gnome-xdg\x2duser\x2ddirs-1237.scope - Application launched by gnome-session-binary.
Apr 02 07:13:37 bad gdm-launch-environment[602]: GLib-GObject: g_object_unref: assertion 'G_IS_OBJECT (object)' failed

```

4. Определите размер журнала.

```

root@bad:/home/vr61v# journalctl --disk-usage
Archived and active journals take up 48.0M in the file system.

```

Часть 5. Создание и настройка .mount юнита:

1. Подготовьте файловую систему.

- Создайте новый раздел на диске или используйте существующий.
- отформатируйте его в файловую систему ext4.
- Создайте директорию для монтирования /mnt/mydata

```
root@bad:/home/vr61v# /sbin/fdisk -l
Disk /dev/sda: 64 GiB, 68719476736 bytes, 134217728 sectors
Disk model: Debian 12.9.0-0
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
Disklabel type: gpt
Disk identifier: F784E11A-015A-4F84-855D-2D3C39433B6B

Device            Start       End   Sectors  Size Type
/dev/sda1         2048     1050623   1048576  512M EFI System
/dev/sda2       1050624 132216831 131166208 62.5G Linux filesystem
/dev/sda3       132216832 134215679   1998848  976M Linux swap
```

```
Disk /dev/sdb: 1 GiB, 1073741824 bytes, 2097152 sectors
Disk model: Debian 12.9.0 -
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
root@bad:/home/vr61v# /sbin/fdisk /dev/sdb

Welcome to fdisk (util-linux 2.38.1).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.
```

```
Device does not contain a recognized partition table.
Created a new DOS (MBR) disklabel with disk identifier 0xb89864d1.

Command (m for help): n
Partition type
   p   primary (0 primary, 0 extended, 4 free)
   e   extended (container for logical partitions)
Select (default p):

Using default response p.
Partition number (1-4, default 1):
First sector (2048-2097151, default 2048):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-2097151, default 2097151):

Created a new partition 1 of type 'Linux' and of size 1023 MiB.

Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.
```

```
root@bad:/home/vr61v# mkdir -p /mnt/mydata
```

2. Создание .mount юнита

- Создайте файл .mount юнита в /etc/systemd/system/mnt-mydata.mount.
- Настройте юнит следующим образом:
 - Добавьте описание юнита в секцию [Unit].
 - В секции [Mount] укажите устройство, точку монтирования, тип файловой системы и опции.
 - В секции [Install] укажите, что юнит должен быть активирован при достижении multi-user.target.
- Сохраните файл и выйдите из редактора.


```

root@bad:/home/vr61v/lab4# cat /etc/systemd/system/mnt-mydata.mount
[Unit]
Description=description
[Mount]
What=/dev/sdb1
Where=/mnt/mydata
Type=ext4
Options=defaults
[Install]
WantedBy=multi-user.target

```

3. Запуск и проверка .mount юнита
 - a. Включите и запустите юнит.
 - b. Проверьте статус юнита.
 - c. Убедитесь, что раздел смонтирован.

```

root@bad:/home/vr61v/lab4# systemctl enable mnt-mydata.mount
root@bad:/home/vr61v/lab4# systemctl status mnt-mydata.mount
○ mnt-mydata.mount - description
   Loaded: loaded (/etc/systemd/system/mnt-mydata.mount; enabled; preset: enabled)
   Active: inactive (dead)
 TriggeredBy: ● mnt-mydata.automount
    Where: /mnt/mydata
    What: /dev/sdb1
root@bad:/home/vr61v/lab4# mount | grep mydata
systemd-1 on /mnt/mydata type autofs (rw,relatime,fd=46,pgrp=1,timeout=300,minproto=5,maxproto=5,direct,pipe_ino=14227)

```

Часть 6. Использование .automount для отложенного монтирования:

1. Подготовьте соответствующий .mount-юнит
 - a. После выполнения Части 5 у вас должен был остаться юнит для монтирования /mnt/mydata
 - b. Убедитесь, что при остановке раздел отмонтируется, а монтируется обратно только при запуске юнита или перезагрузке системы
2. Создание .automount юнита
 - a. Создайте файл .automount юнита в /etc/systemd/system/mntmydata.automount.
 - b. Настройте юнит:
 - i. В секции [Unit] добавьте описание.
 - ii. В секции [Automount] укажите точку монтирования и время до размонтирования (TimeoutIdleSec).
 - iii. В секции [Install] укажите, что юнит должен быть активирован при достижении multi-user.target.
 - c. Сохраните файл и выйдите из редактора.
3. Запуск и проверка .automount юнита
 - a. Включите и запустите .automount юнит.
 - b. Проверьте статус юнита и убедитесь, что раздел монтируется при обращении к точке монтирования.
 - c. Убедитесь, что раздел размонтируется после завершения работы.

```
root@bad:/home/vr61v# nano /etc/systemd/system/mnt-mydata.automount
root@bad:/home/vr61v# cat /etc/systemd/system/mnt-mydata.automount
[Unit]
Description=description
[Automount]
Where=/mnt/mydata
TimeoutIdleSec=300
[Install]
WantedBy=multi-user.target
root@bad:/home/vr61v# systemctl disable --now mnt-mydata.mount
Removed "/etc/systemd/system/multi-user.target.wants/mnt-mydata.mount".
root@bad:/home/vr61v# systemctl enable --now mnt-mydata.automount
Created symlink /etc/systemd/system/multi-user.target.wants/mnt-mydata.automount → /etc/systemd/system/mnt-mydata.automount.
root@bad:/home/vr61v# systemctl status mnt-mydata.automount
● mnt-mydata.automount - description
   Loaded: loaded (/etc/systemd/system/mnt-mydata.automount; enabled; preset: enabled)
   Active: active (waiting) since Wed 2025-04-02 08:25:14 EDT; 9s ago
     Triggers: ● mnt-mydata.mount
        Where: /mnt/mydata
```

Apr 02 08:25:14 bad systemd[1]: Set up automount mnt-mydata.automount - description.