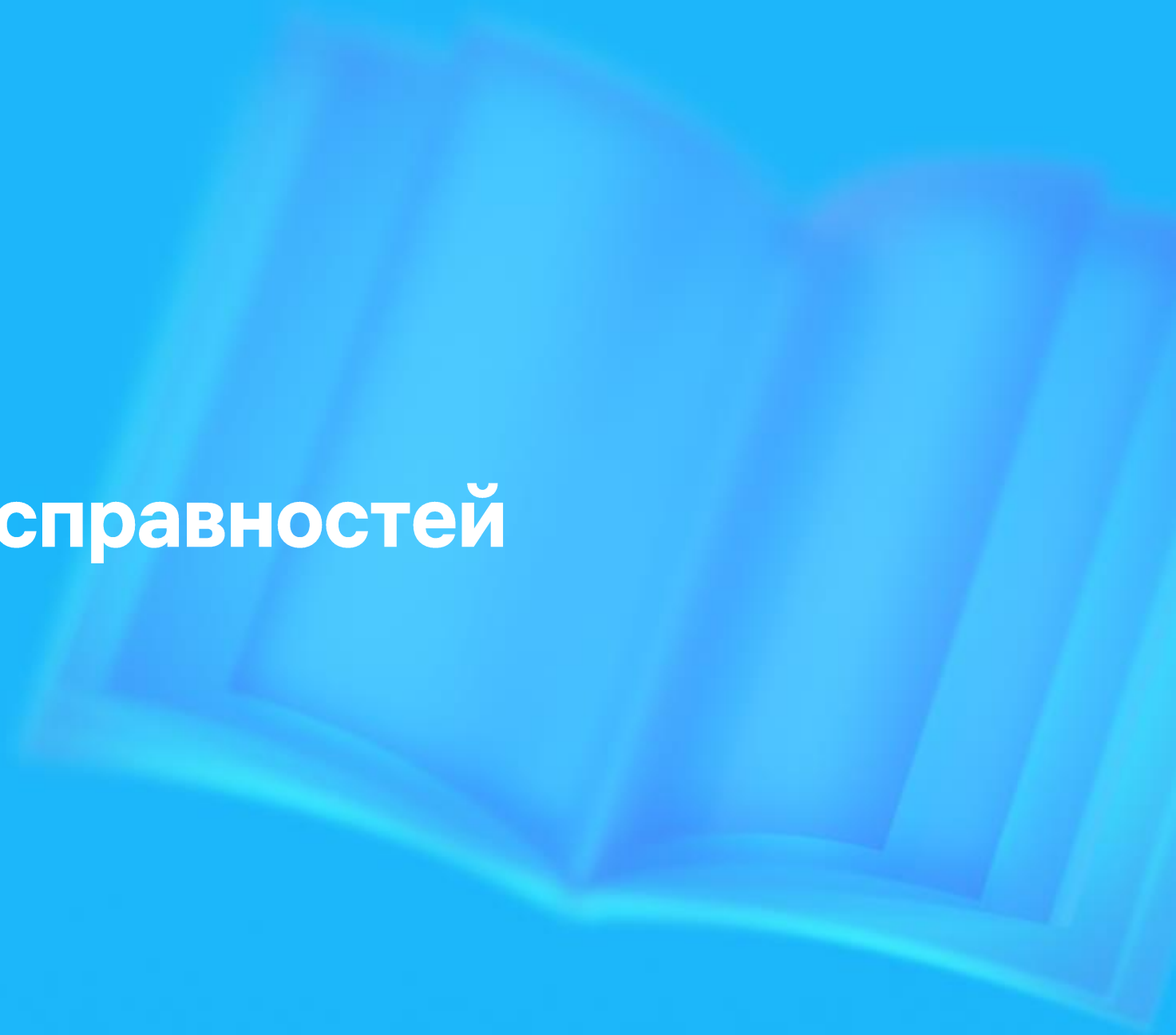


# Вебинар №4. Управление ПО и устранение неисправностей



# Что будет сегодня



Узнаете об управлении программным обеспечением



Познакомитесь с процессами ОС Astra Linux



Узнаете о процессе загрузки и выключения системы



Научитесь искать и устранять неисправности

# Управление программным обеспечением



## Компоненты системы управления ПО

Компоненты системы управления ПО в Linux включают следующие элементы:

1. Пакетный менеджер. Пакетные менеджеры: apt-get, yum, pacman, apt-get (наиболее популярен в deb-системах), aptitude и dpkg.
2. Репозиторий.
3. Зависимости.
4. Конфигурационные файлы.
5. Система контроля версий.
6. Сборочные инструменты.
7. Система автоматической сборки.

## Именованение и структура программных пакетов

Рассмотрим наименование и версионирование пакета.

example-package\_1.0-1\_amd64.deb

- example-package — название пакета;
- 1.0 — номер версии;
- 1 — номер ревизии (если в пакет были внесены изменения после его первоначального выпуска);
- amd64 — архитектура, для которой был собран пакет.

Структура deb-пакета обычно следующая (вы можете открыть любой пакет \*.deb для просмотра без распаковки с помощью утилиты `mc`: `mc` → выбираем пакет → Enter):

Левая панель				Правая панель			
Файл		Команда		Настройка		Правая панель	
...t UI deb-5.3.2.0-3.deb/deb://				[ ^ ]			
.и	Имя	Размер	Время правки	.и	Имя	Размер	Время правки
/..	-ВВЕРХ-	янв 18	21:01	/..	-ВВЕРХ-	июл 17	2019
/CONTENTS	0	июн 1	14:11	/.ansible	4096	мар 28	23:37
/DEBIAN	0	июн 1	14:11	/.bash-git-prompt	4096	янв 15	13:00
INFO	884	июн 1	14:11	/.cache	4096	мая 9	14:17
*INSTALL	424	июн 1	14:11	/.cert	4096	янв 4	12:25
				/.config	4096	июн 1	12:23
				/.gnupg	4096	июл 17	2019
				/.local	4096	июл 17	2019
				/.mozilla	4096	июн 17	2021
				/.npm	4096	янв 4	13:16
				/.nv	4096	авг 19	2019
				/.pki	4096	июн 17	2021
				/.ssh	4096	июн 1	14:09
				/.terraform.d	4096	мая 22	20:27
				/.vagrant.d	4096	фев 21	14:42
				/.vim	4096	янв 15	11:51
				/.vscode-oss	4096	янв 4	13:31
- ВВЕРХ -				- ВВЕРХ -			
				4560M / 68G (6%)			

Совет: Удаленный текст можно вернуть с помощью C-у.

[boskvd@arch-univ GP]\$

1Помощь 2Меню 3Про-отр 4Правка 5Копия 6Перенос 7НавТлог 8Удалить 9МенюМС 10Выход

## Структура репозитория программного обеспечения

Astra Linux использует структуру репозитория, подобную Debian.

Основные компоненты репозитория Astra Linux:

1. `dists/`: директория, которая содержит файлы `Packages.gz` и `Release`. Эти файлы содержат информацию обо всех доступных пакетах в репозитории и их версиях;
2. `pool/`: директория, которая содержит все пакеты, разделённые по категориям и включающие их разные версии;
3. `security/`: содержит обновления безопасности для уже установленных пакетов;
4. `updates/`: содержит исправления ошибок и новые функции для уже установленных пакетов;
5. `project/`: содержит дополнительные файлы, связанные с проектом.

## Менеджеры программных пакетов dpkg, apt, apt-команды, aptitude

Менеджеры программных пакетов dpkg, apt и aptitude могут быть использованы в различных сценариях:

1. Если нужно управлять пакетами низкого уровня и устанавливать их ручным способом.
2. Может автоматически решать зависимости и устанавливать необходимые пакеты, что делает процесс установки программного обеспечения гораздо более удобным и быстрым

## Подключение репозитория Astra Linux

Чтобы указать программам-установщикам пакетов репозитории, из которых будут загружаться устанавливаемые пакеты, следует для каждого репозитория сформировать строку вида:

```
deb <путь_к_корневому_каталогу_репозитория> <код_дистрибутива> <список_компонент>
```

1. Эти строки добавить в файл `/etc/apt/sources.list` или в файлы с произвольными именами и расширением `.list`, расположенные в каталоге `/etc/apt/sources.list.d`.
2. После каждого внесения изменений в файлы с описанием репозитория следует обновить списки пакетов:  

```
sudo apt update.
```



## Подключение стороннего репозитория

Подключение сторонних репозиториях, например, репозиториях Debian. Для этого:

1. Установить пакет `debian-archive-keyring`: `sudo apt install debian-archive-keyring`.
2. Добавить описание репозиториях, например, в файл `/etc/apt/sources.list.d/buster.list`:  
`deb https://deb.debian.org/debian/ buster main contrib non-free`  
`deb https://security.debian.org/debian-security/ buster/updates main contrib non-free`
3. Обновить кеш пакетов:  
`sudo apt update`
  - Суц:1 `http://dl.astralinux.ru/astra/stable/1.7_x86-64/repository-update 1.7_x86-64 InRelease`
  - Суц:2 `http://dl.astralinux.ru/astra/stable/1.7_x86-64/repository-base 1.7_x86-64 InRelease`
  - Суц:3 `http://dl.astralinux.ru/astra/stable/1.7_x86-64/repository-extended 1.7_x86-64 InRelease`
  - Суц:4 `https://security.debian.org/debian-security buster/updates InRelease`
  - Суц:5 `https://deb.debian.org/debian buster InRelease`

Сторонние репозитории Debian подключены.

## Установка обновлений

1. Подключить интернет-репозитории Astra Linux. Для этого в файле `/etc/apt/sources.list` добавляем строки:

```
deb http://dl.astralinux.ru/astra/frozen/1.7_x86-64/1.7.4/repository-main/ 1.7_x86-64 main contrib non-free
```

```
deb http://dl.astralinux.ru/astra/frozen/1.7_x86-64/1.7.4/repository-update/ 1.7_x86-64 main contrib non-free
```

2. Сохранить. Если в качестве источников пакетов указан оптический установочный диск, т. е. присутствует строка вида:

```
deb cdrom:[<наименование_установочного_диска>]/ 1.7_x86-64 contrib main non-free,
```

то эту строку необходимо удалить или закомментировать (установить символ `#` в начале строки).

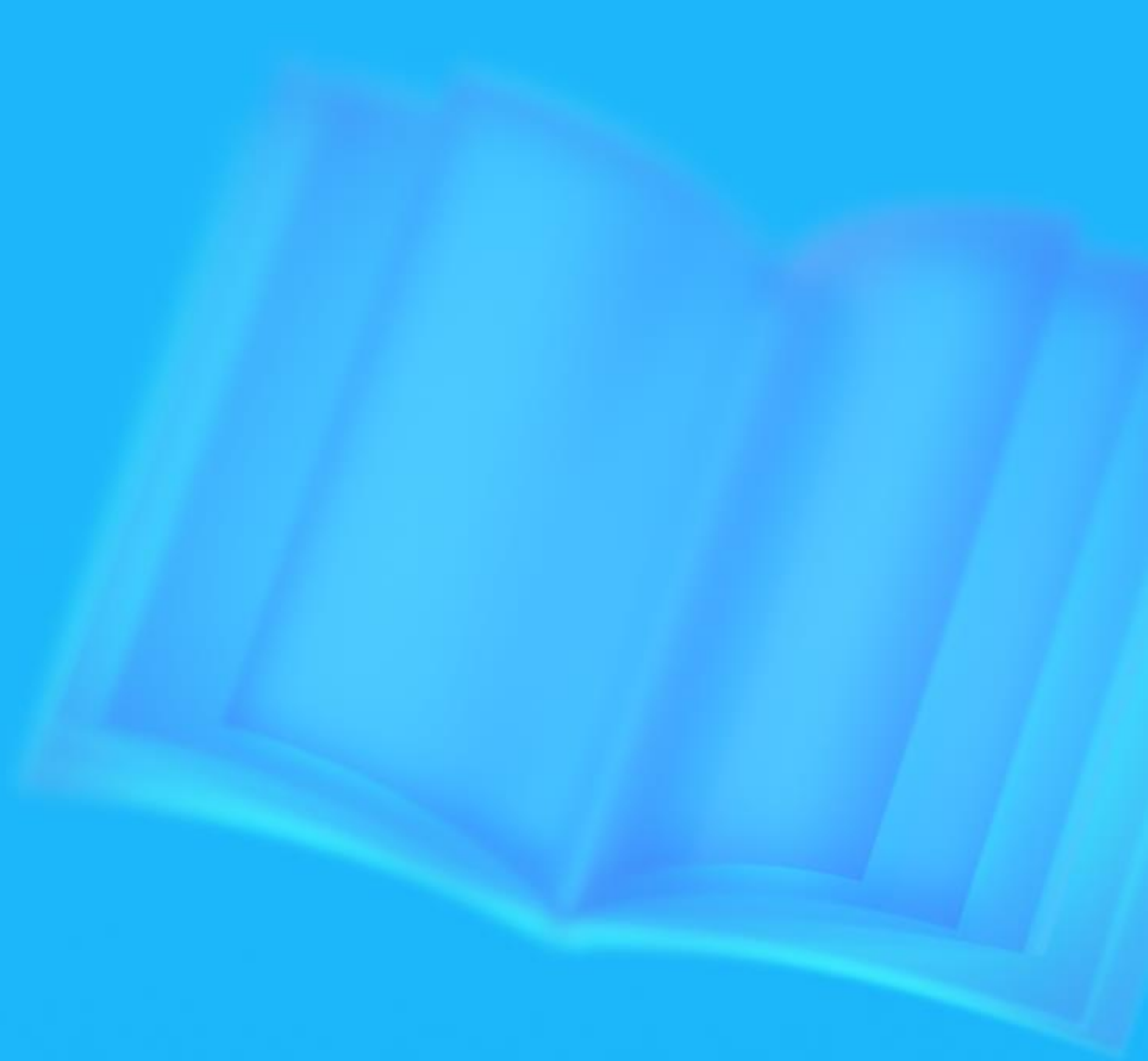
3. Выполнить повторную синхронизацию файлов описаний пакетов с их источником. Для этого выполняем команду:

```
sudo apt update
```

4. Установить обновление, выполнив команду:

```
sudo astra-update -A -r
```

# Процессы в Linux



## Общие понятия о программах, процессах и потоках выполнения

Программа в Linux — это исполняемый файл, который может быть запущен в отдельном процессе.

Процесс в Linux — это экземпляр запущенной программы, который включает исполняемый код программы, данные и ресурсы, такие как файловые дескрипторы, сегменты памяти и т. д.

Поток выполнения (поток) — это единица параллельной обработки внутри процесса.

## Жизненный цикл процесса



В Linux существуют также системные вызовы, которые позволяют родительскому процессу мониторить и управлять его дочерними процессами, такие как `wait()` и `kill()`.

## Настройка доступа к общим библиотекам

Для настройки доступа к общим системным библиотекам в Astra Linux можно использовать команды `chmod` и `chown`, которые позволяют изменять права доступа и владельца файлов.

Для обновления кеша библиотек и добавления новых библиотек в список общих библиотек можете использовать команду `ldconfig`. Это может быть необходимо, если вы установили новую библиотеку в каталог общих библиотек.

Команда `ldconfig`

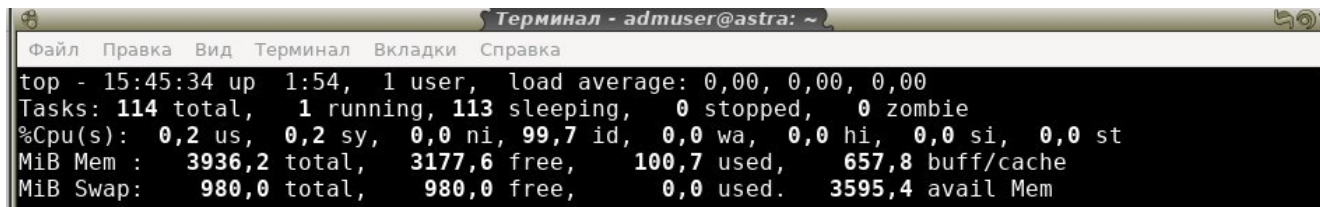
## Мониторинг процессов

Мониторинг процессов в Linux может быть осуществлён различными способами. Один из наиболее популярных— использование утилиты `top`.

Консольная утилита `top` представляет собой интерактивный мониторинг процессов в Linux.

Она отображает список процессов и информацию о потребляемых ими ресурсах: процессор, память, потоки ввода-вывода и другие.

Вывод `top` состоит из двух секций: верхняя и нижняя.



```

Терминал - admuser@astra: ~
Файл  Правка  Вид  Терминал  Вкладки  Справка
top - 15:45:34 up 1:54, 1 user, load average: 0,00, 0,00, 0,00
Tasks: 114 total, 1 running, 113 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0,2 us, 0,2 sy, 0,0 ni, 99,7 id, 0,0 wa, 0,0 hi, 0,0 si, 0,0 st
MiB Mem : 3936,2 total, 3177,6 free, 100,7 used, 657,8 buff/cache
MiB Swap: 980,0 total, 980,0 free, 0,0 used. 3595,4 avail Mem
  
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
2785	admuser	20	0	11600	4316	3592	R	0,3	0,1	0:00.26	top
1	root	20	0	103340	11700	8828	S	0,0	0,3	0:03.67	systemd
2	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kthreadd
3	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	rcu_gp
4	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	rcu_par_gp
6	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworker/0:0H-kblockd
9	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	mm_percpu_wq
10	root	20	0	0	0	0	S	0,0	0,0	0:00.04	ksoftirqd/0
11	root	20	0	0	0	0	I	0,0	0,0	0:00.87	rcu_sched
12	root	rt	0	0	0	0	S	0,0	0,0	0:00.06	migration/0
13	root	-51	0	0	0	0	S	0,0	0,0	0:00.00	idle_inject/0
14	root	20	0	0	0	0	S	0,0	0,0	0:00.00	cpuhp/0
15	root	20	0	0	0	0	S	0,0	0,0	0:00.00	cpuhp/1
16	root	-51	0	0	0	0	S	0,0	0,0	0:00.00	idle_inject/1
17	root	rt	0	0	0	0	S	0,0	0,0	0:00.12	migration/1
18	root	20	0	0	0	0	S	0,0	0,0	0:00.04	ksoftirqd/1
20	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworker/1:0H-kblockd
21	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kdevtmpfs
22	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	netns
23	root	20	0	0	0	0	S	0,0	0,0	0:00.00	rcu_tasks_kthre
24	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kauditd
25	root	20	0	0	0	0	S	0,0	0,0	0:00.00	khungtaskd
26	root	20	0	0	0	0	S	0,0	0,0	0:00.00	oom_reaper
27	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	writeback
28	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kcompactd0
29	root	25	5	0	0	0	S	0,0	0,0	0:00.00	ksmd
30	root	39	19	0	0	0	S	0,0	0,0	0:00.00	khugepaged
77	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kintegrityd

## Управление приоритетом процесса

Управление приоритетом процессов в Linux может помочь улучшить производительность системы, распределяя ресурсы процессора и памяти между процессами.

В Linux существует несколько способов изменения приоритета процессов:

1. Утилиты `nice`.
2. Команда `renice`.
3. Утилита `top`.
4. Файл `/etc/security/limits.conf`.



Сигналы в Linux — это механизм для уведомления процессов о событиях, таких как завершение работы другого процесса или получение сигнала от другого процесса. Каждый сигнал имеет свой уникальный номер и может быть отправлен в процесс с помощью функции `kill()`.

Сигналы в Linux — это основа для многих механизмов IPC.

## Управление заданиями

Управление заданиями в Linux является важным аспектом работы в операционной системе и включает планирование заданий на выполнение, запуск заданий, управление запущенными заданиями и многое другое.

Наиболее популярный инструмент планирования задач в Linux — это **cron**.

Cron позволяет планировать выполнение задач в указанные моменты времени или по расписанию. Для этого необходимо:

1. Создать файл `crontab`, который содержит список задач и расписание их выполнения. Создание файла `crontab` можно осуществить с помощью команды `crontab -e`
2. В файле `crontab` каждая строка соответствует одной задаче и имеет формат  
`* * * * * command`

В файле `/etc/cron.allow` можно указать список пользователей, у которых будет доступ к `cron`, или в файле `/etc/cron.deny` указать пользователей, которым запретим доступ.

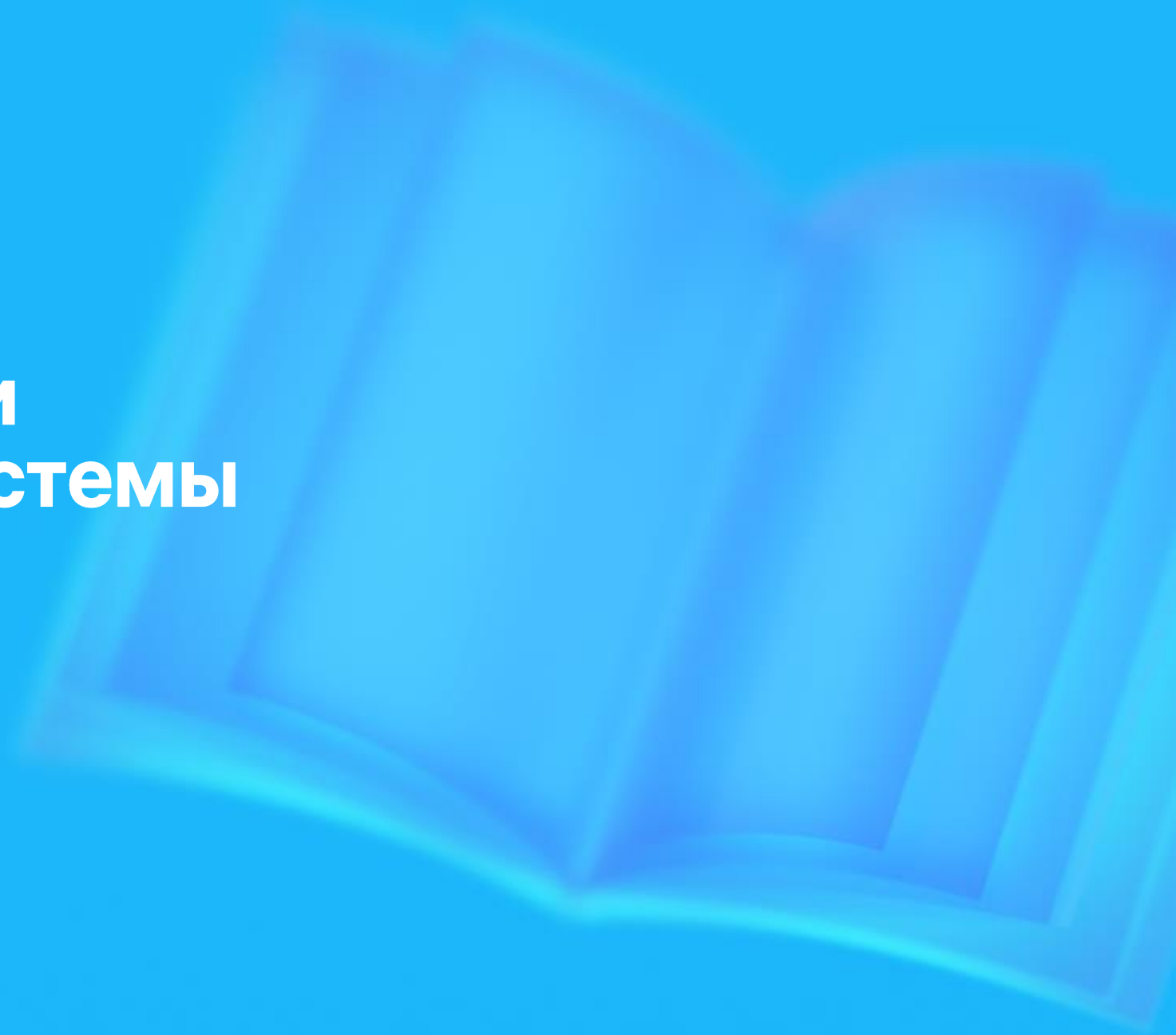
## Управление заданиями

Полезно иногда посмотреть в файл `/etc/crontab`, туда тоже могут записываться расписания.

В Linux существует множество других программ для управления заданиями, включая аналоги `crontab`, такие как `anacron` и `fcron`. Есть также интерактивные планировщики задач, например, GNOME Schedule и KAlarm, которые предоставляют более удобный интерфейс для планирования задач.

- Команда `kill`.
- Команда `bg`.
- Команда `fg` (foreground).

# Процесс загрузки и выключения системы



## Этапы процесса загрузки

1. **BIOS/UEFI**: первичная инициализация аппаратных компонентов, например, проверка и установка системных часов, определение жёстких дисков, проверка наличия оперативной памяти и других аппаратных средств. Если найдены ошибки, компьютер может выдавать соответствующие сообщения или сообщения об ошибках;
2. **Bootloader**: компьютер производит передачу управления загрузчику GRUB, который инициализирует загрузку системы. Из меню управления загрузчика можно выбрать операционную систему, соответствующую определённому дистрибутиву Astra Linux;
3. **Kernel**: загружается ядро операционной системы Linux, которое определяет аппаратные конфигурации, файловые системы, маршрутизацию и управление устройствами ввода/вывода и другие настройки. После загрузки ядра происходит инициализация основных компонентов и драйверов системы.

## Этапы процесса загрузки

**4. Init:** запуск сервисов, указанных в файле `/etc/inittab` или `/etc/init.d`, и скриптов, которые автоматически запускаются при загрузке. Эти сервисы могут включать сетевые службы, демоны, процессы и другие приложения;

**5. Runlevel:** Astra Linux переходит в режим работы, определённый текущим уровнем запуска (runlevel). По умолчанию используется пять уровней запуска:

- N — уровень запуска по умолчанию, который определяет работу в многофункциональном режиме, сочетающем консольный и графический доступ.
- 0 — выключение операционной системы.
- 1 — однопользовательский режим, предоставляющий консольный доступ без сетевых сервисов.
- 2 — многофункциональный режим, но без графического доступа.
- 6 — перезагрузка операционной системы.

На каждом уровне запуска определённый набор сервисов запускается автоматически, что позволяет оптимизировать работу системы в зависимости от её конфигурации и потребностей пользователя;

**6. Services:** система активирует и продолжает работу запущенных сервисов. Сервисы продолжают работать в фоновом режиме, выполняя задачи, которые были назначены им на стадии инициализации и запуска. Они могут выполнять различные задачи: от отслеживания сетевых подключений до выполнения процессов бэкапа.

Для настройки GRUB2 пользователям нужно отредактировать конфигурационный файл `/etc/default/grub` и сохранить изменения. Некоторые параметры, которые можно менять, включают таймер отсчёта, действия, выполняемые перед загрузкой операционной системы по умолчанию, режимы консоли вывода и другие.

После сохранения изменений в конфигурационном файле `/etc/default/grub` пользователь должен запустить команду `update-grub`. Это обновляет файл `grub.cfg` и применяет настройки, внесённые в файл `/etc/default/grub`.

В файле `/etc/default/grub` можно настроить параметры загрузки ядра Astra Linux. Внесите изменения в этот файл и выполните команду `update-grub` для обновления файла конфигурации `grub.cfg`. Пример файла `/etc/default/grub` на Astra Linux.

Модули возможно загружать или выгружать вручную командой `modprobe` или `rmmod`.

Можно изменить список автоматически загружаемых модулей, добавив их в список в файле `/etc/modules`. Здесь вы можете перечислить имена модулей, которые операционная система должна загружать при старте.

Модули ядра можно также настраивать через конфигурационный файл `/etc/modprobe.d/`, например, чтобы загрузить модуль с определёнными параметрами.

Кроме того, вы можете использовать команду `lsmod` для просмотра списка текущих загруженных модулей. Команда `modinfo` позволяет получить информацию о конкретном модуле ядра.

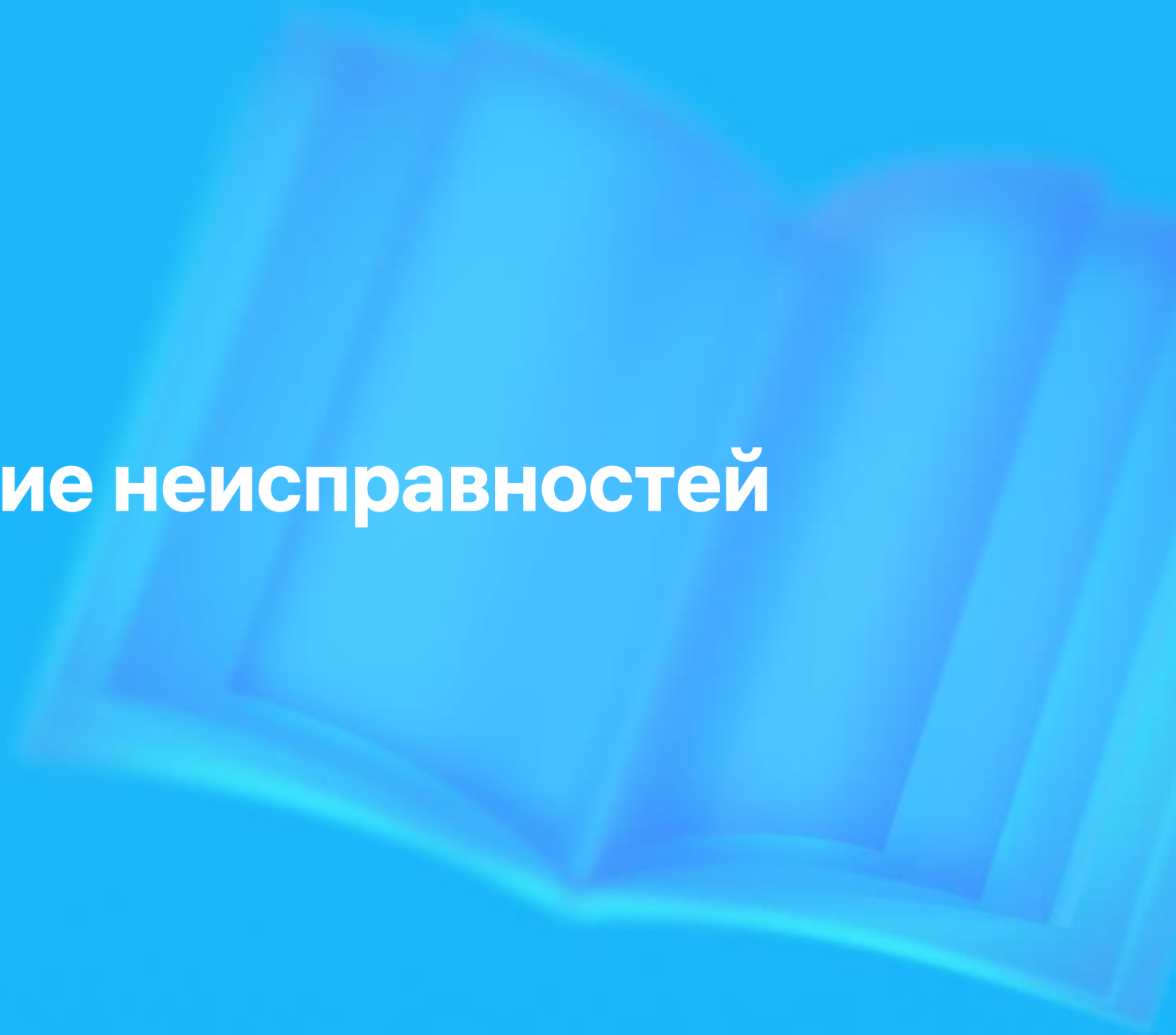


## Управление процессами

Управление процессами systemd происходит с помощью systemctl:

- `systemctl list-units --type service --all` — просмотр всех юнитов в системе;
- `systemctl start name` — запустить сервис;
- `systemctl stop name` — остановить сервис;
- `systemctl restart name` — перезапустить сервис;
- `systemctl status name` — посмотреть статус сервиса;
- `systemctl reload name` — перечитать конфигурацию;
- `systemctl daemon-reload` — перечитать конфигурацию для всех;
- `systemctl try-restart name` — перезапустить, если запущен;
- `systemctl enable name` — включить автозапуск сервиса;
- `systemctl disable name` — отключить автозапуск сервиса;
- `systemctl list-unit-files --type service` — список установленных юнит-файлов сервисов;
- `systemctl list-unit-files --type=target` — показывает список всех целевых состояний и их статус;
- `systemctl isolate [имя_целевого_состояния]` — переключает систему в указанное целевое состояние;
- `systemctl mask [имя_целевого_состояния]` — запрещает запуск указанного целевого состояния;
- `systemctl unmask [имя_целевого_состояния]` — разрешает запуск указанного целевого состояния, если он был запрещён.

# Поиск и устранение неисправностей



## Методология поиска и устранения неисправностей

1. Проверка сетевой связи:  
**ping -c3 8.8.8.8**
2. Просмотр сетевых соединений:  
**netstat -anv | grep tcp**
3. Проверка свободного места на диске:  
**df -h /**
4. Проверка журналов:  
**journalctl -n50**
5. Проверка процессов:  
**ps aux | grep username**
6. Проверка файловой системы:  
**sudo fsck /dev/sdb1**
7. Мониторинг системных ресурсов:  
**top/htop**

## Устранение неисправностей

### 1. Проблемы с загрузчиком GRUB.

Если система не загружается из-за проблем с загрузчиком GRUB, можно создать загрузочный диск или USB-накопитель и восстановить загрузчик.

### 2. Неполадки с жёстким диском.

### 3. Нехватка оперативной памяти.

### 4. Проблемы с драйверами устройства.

Поможет утилита **modprobe**, которая позволяет загружать или выгружать драйверы.

### 5. Неполадки с таблицей разделов.

Проверить таблицу разделов можно при помощи команды **fdisk -l**.

Во всех случаях можно попробовать загрузить систему в безопасном режиме (recovery mode) или однопользовательском режиме (single mode).

**Спасибо за внимание!**

