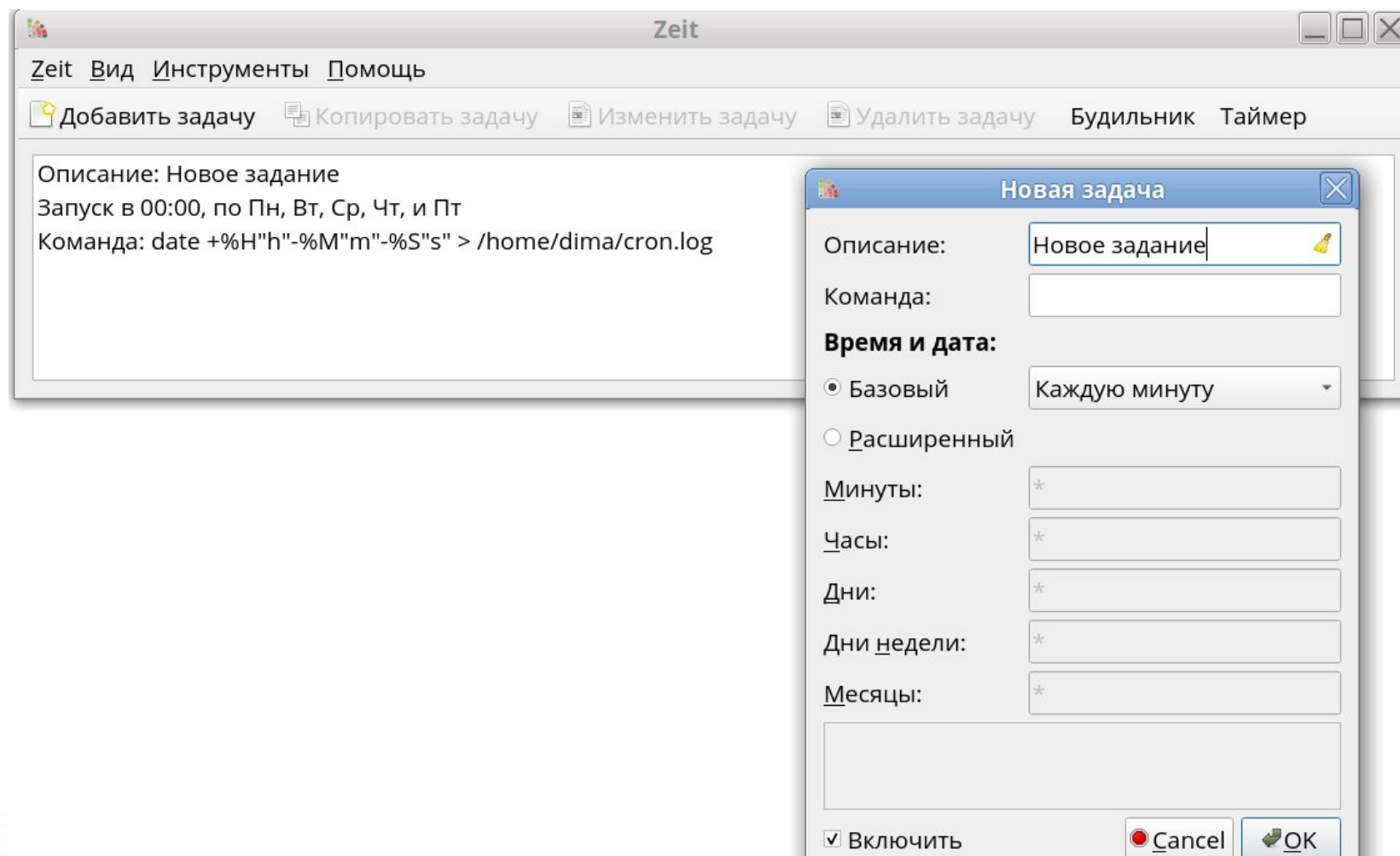


# Планирование заданий в РЕД ОС

# Графический планировщик zeit



# Запуск разовых задач at

Служба **atd** предназначена для выполнения команд в заданное время.

**at <время> <дата>**

**at 12:15 PM today**

**at 12:15 tomorrow**

**at now +1 minute**

**at 11:10**

**at> echo "Test"**

**at> <EOT>       # Нажатие Ctrl+D**

**echo "echo Test" | at 12:15 tomorrow**

**echo "cp test /tmp" | at 12:15**

# Запуск разовых задач at

Список задач текущего пользователя — **atq**

6 Tue Nov 10 12:15:00 2020 a root

4 Tue Nov 10 12:15:00 2020 a root

Просмотр установленной задачи — **at -c**

Запуск скрипта — **at -f <filename> -m <time>**. Скрипт может находиться в любой папке, но лучше в папке указанной в переменной **\$PATH**

# Запуск разовых задач at

Удаление задачи из очереди — **atrm номер**

Запуск задачи во время простоя компьютера - **batch**. Задача выполнится когда средняя загрузка будет меньше **0.8**.

Файлы **/etc/at.allow** и **/etc/at.deny** позволяют ограничить доступ к командам **at** и **batch**.

Если файл **at.allow** существует, использовать **at** и **batch** разрешено только перечисленным в нём пользователям, а файл **at.deny** игнорируется.

Если файл **at.allow** не существует, пользователям, перечисленным в **at.deny**, использовать **at** и **batch** не разрешается

# Периодические задачи crontab

Запуск задач по расписанию командой **cron**

Служба **crond** предназначена для периодического выполнения задач

Конфигурационный файл - **/etc/crontab** (не рекомендуется изменять)

Конфигурационные каталоги:

**/etc/cron.d**

**/etc/cron.hourly** - каждый час;

**/etc/cron.daily** - каждый день;

**/etc/cron.weekly** - каждую неделю;

**/etc/cron.monthly** - каждый месяц.



Задачи можно добавлять также с помощью **crontab -e**

Откроется редактор по умолчанию, обычно это **vi**. Редактор **vi** можно временно сменить командой **export EDITOR=nano**

При закрытии редактора конфигурация будет установлена автоматически.

Просмотр задач — **crontab -l**.

# Периодические задачи crontab

*	*	*	*	*
Минута: 0-59	Час: 0-23	День месяца: 1-31	Месяц: 1-12 <b>JAN-DEC</b>	День недели: 7,0-6 <b>SUN-SAT</b>

\* любое значение  
, перечисление  
- диапазон  
/ шаг применения

Вместо первых 5 полей можно использовать специальные псевдонимы:

**@reboot** — при загрузке, только один раз;

**@yearly, @annually** — раз год;

**@monthly** — раз в месяц;

**@weekly** — раз в неделю;

**@daily, @midnight** — каждый день;

**@hourly** — каждый час.

# Периодические задачи crontab

Пример задач в crontab

# выполнять каждый день в 0 часов 5 минут, результат складывать в log/daily

**5 0 \* \* \* echo "Выполняется каждый день в 0:5"**

**15 14 1 \* \* echo "Выполняется в 1 числа каждого числа в 14:15"**

**0 22 \* \* 1-5 echo "Рабочий день закончен" | mail -s "Уже 22:00" john**

**23 \*/2 \* \* \* echo "Выполняется в 0:23, 2:23, 4:23 и т. д."**

**5 4 \* \* sun echo "Выполняется в 4:05 в воскресенье"**

**0 0 1 1 \* echo "С новым годом!"**

**15 10,13 \* \* 1,4 echo "Выполняется в понедельник и четверг в 10:15 и 13:15"**

**0-59 \* \* \* \* echo "Выполняется ежеминутно"**

**\* \* \* \* \* echo "Это тоже ежеминутно"**

**0-59/2 \* \* \* \* echo "Выполняется по чётным минутам"**

**1-59/2 \* \* \* \* echo "Выполняется по нечётным минутам"**

**\*/5 \* \* \* \* echo "Прошло пять минут"**

**@hourly echo "Ежечасно, т. е. аналог 0 \* \* \* \*"**

**@weekly echo "Еженедельно, т. е. аналог 0 0 \* \* 0"**



# Периодические задачи crontab

Расписания заданий для каждого пользователя свои. Они хранятся в **/var/spool/cron/**, но редактировать их нужно через **crontab -e**

Журнал работы crond находится в **/var/log/cron**

Вывод расписания текущего пользователя — **crontab -l**

Удаление расписания текущего пользователя — **crontab -r**

Файлы **/etc/cron.allow** и **/etc/cron.deny** позволяют ограничить возможность использования планировщика cron:

В файле **cron.allow** — этим пользователям **cron** разрешён. **cron.deny** игнорируется

В файле **cron.deny** — этим пользователям **cron** запрещён. Если нет **cron.allow**.

Если из этих файлов не существует, использовать cron может только **root**.

# Планировщик заданий anacron

Асинхронный или анахроничный cron, позволяет запускать запланированные задания только на ежедневной основе. Доступен только **root**. Задания запускаются **интервалами**. Если задание не выполнилось, оно будет выполнено **после загрузки**.

Файл конфигурации - **/etc/anacrontab**

SHELL=/bin/sh

PATH=/sbin:/bin:/usr/sbin:/usr/bin

MAILTO=root

RANDOM\_DELAY=45 # дополнительная пауза к базовой задержке

START\_HOURS\_RANGE=3-22 # запускать только в это время

Период	задержка	ID_задачи	команда
1	5	<i>cron.daily</i>	<i>nice run-parts /etc/cron.daily</i>
7	25	<i>cron.weekly</i>	<i>nice run-parts /etc/cron.weekly</i>
<i>@monthly</i>	45	<i>cron.monthly</i>	<i>nice run-parts /etc/cron.monthly</i>

# Таймеры systemd

В systemd для запуска заданий по расписанию предназначены юниты типа таймер (**.timer**)

В таймер-юните указывается, когда будет запущена одноименная служба Таймер-юнит должен содержать секцию **[Timer]**, внутри которой размещаются параметры, устанавливающие расписание:

- OnCalendar=день\_недели YYYY-MM-DD HH:MM:SS
- AccuracySec= -а временная точность выполнения заданий (по умолчанию 1 мин)

Получить информацию о запущенных таймерах можно командой  
**systemctl list-timers**

# Ознакомиться с роликами

Планировщик задач в РЕД ОС. Команда at

<https://rutube.ru/video/c91e9e95f20bece2ce60120947cb41b1/>

Планировщик задач в РЕД ОС. Cron

<https://rutube.ru/video/f85b5dd4e1e1c0a8aa81de8f65727be8/>

Планировщик задач в РЕД ОС. Anacron

<https://rutube.ru/video/141d8a4e3b786ac1ffa16ec26e3739c3/>

Добавление команд в автозагрузку

<https://rutube.ru/video/f870706611a010176d2b4d83df179f8a/>

# Практическая работа

1. Запустить команду `touch ~/at.file` с помощью `at` через 2 минуты от текущего
2. Вывести список задач `at`, вывод перенаправить в файл `at.log` в домашнем каталоге.
3. Создать в `crontab` задачу, выполняющуюся каждую минуту. Она должна дописывать текущее время в файл `cron.log` в домашнем каталоге пользователя.
4. Создать в `crontab` задачу, выполняющуюся в полночь по рабочим дням. Она должна удалять из домашнего каталога пользователя все файлы с расширением `log`.
5. Создать в `crontab` задачу, выполняющуюся в воскресенье в час ночи. Она должна определять свободное место на дисках в мегабайтах и вывод отправлять в файл `free.log` в домашнем каталоге пользователя.
6. Установить программу `zeit`, добавить в неё задачу, которая запускает скрипт с периодом 4 минуты записывает дату в файл.



**Спасибо за внимание!**

**[www.red-soft.ru](http://www.red-soft.ru)**  
**[redos@red-soft.ru](mailto:redos@red-soft.ru)**

