

# ...а служба идет

Жизнь часто преподносит нам сюрпризы, и по большей части неприятные. Один из наиболее ярких примеров — потеря информации, пусть даже не рабочих проектов, а ваших личных данных. Тех, с которыми связаны различные, порой самые теплые воспоминания.

Предотвратить катастрофу достаточно легко — нужно просто периодически создавать копии нужных данных и переносить их на какой-либо внешний накопитель. Единственная проблема здесь заключается в том, что использовать специализированные программы для бэкапа при столь скромных задачах не совсем уместно. В основном из-за того, что бесплатные средства, как правило, не обладают достаточной функциональностью, а платные, если и обладают, то стоят весьма ощутимых денег. Но мы нашли решение, позволяющее достаточно легко и быстро выполнять поставленные задачи, оставаясь при этом бесплатным. Приступим.

## | Реализация |

Переходим к написанию нужного нам скрипта. Его задача — создать в определенной нами директории архив данных (с названием, содержащим сегодняшнюю дату), который будет состоять из заранее заданных директорий. Если требуется исключить какой-либо подкаталог из указанного каталога, просто пропишите полный путь до него в файле с говорящим названием `exclude.conf`. Сам скрипт можно напечатать в любом текстовом редакторе. Содержание должно быть примерно следующим, но может варьироваться в зависимости от ваших потребностей:

```
#!/bin/sh
cd /home/user/backup && rm -f ./*.tar.gz
tar -vcf up'date +%F'.tar /home/user/work --exclude-
from=./exclude.conf
gzip -9 ./up'date +%F'.tar
```

Файл исключений `exclude.conf` может выглядеть так:

```
/home/user/work/no
```

В первой строке указывается командный интерпретатор, который будет обрабатывать все наши дальнейшие коман-

ды. Вторая указывает скрипту перейти в ту директорию, которая содержит все создаваемые архивы, и если таковые в ней уже имеются — удалить их. Если угодно, можно изменить структуру скрипта таким образом, чтобы предыдущий архив удалялся только после успешного завершения процедуры создания нового. Третья строка данного скрипта указывает программе `tar` создать архив с именем, содержащим текущую дату. Однако имя архива можно менять с помощью различных ключей команды `date`, полный список которых откроется в консоли после ввода команды `date --help`. Данный архив включает все содержимое директории `/home/user/work`, за исключением тех поддиректорий, которые описаны в файле `exclude.conf`. Последняя строка скрипта вызывает консольную утилиту `gzip` и с ее помощью упаковывает архив, который был получен на предыдущем этапе.

## | Автоматизация |

В завершение несколько автоматизируем запуск нашего скрипта для создания резервной копии. Например, мы хотим запускать его ежедневно, в пять часов утра. Для этого удобнее будет воспользоваться планировщиком `cron` и от имени пользователя, работающего с системой, дать следующую команду:

```
crontab -e
```

В появившемся окне впишем следующую строку:

```
02 5 * * * /home/al/backup/backup.sh
```

Для сохранения файла введем последовательность:

```
:qw
```

Теперь вы сможете хранить все документы в домашнем каталоге, где они будут регулярно сохраняться в надежном месте, занимая при этом минимум ресурсов. |