sudo make me a sandwich. Решаем повседневные задачи при помощи командной строки

Ник Чадовский, 12 ноября 2015, в разделах: Веб, IT, Дизайн и интерфейсы

Содержание статьи

- Простые графики
- Баннеры с прогрессбаром
- Сложные задачи
- Поиск информации
- Поиск в интернете

Графический интерфейс, не обман конечного пользователя. Пользуясь, в последнее время, исключительно консолью, я стал замечать, что мне не хватает привычных инструментов для решения встречающихся повседневных задач. Поэтому, в поисках аналогов привычных утилит, я стал искать, что же еще можно сделать из консоли. Оказалось, что в Linux-подобных системах для работы с графикой и не только есть тоннами разнообразный набор утилит, позволяющий делать всякие вещи. Вот о таких утилитах для решения повседневных задач и пойдет речь.

ПРОСТЫЕ ГРАФИКИ

Рисование в терминале

Большинство из нас знает, что полезные графики по-прежнему, особенно в небольшие проекты, удобнее всего делать в Excel. Но, если вы, как и я, предпочитаете не выходить из терминала, то вам, скорее всего, понадобится что-нибудь вроде gnuplot. Но, если вам нужно что-то простое, то можно обойтись и без него. Для этого есть замечательная утилита — ttyplot. Для начала нужно ее установить. В Debian/Ubuntu:

```
sudo apt-get install ttyplot
```

Теперь, к примеру, можно отслеживать загрузку процессора. Сначала узнаем, какой процессор у нас. Для этого утилита cat /proc/cpuinfo выдаст нам всю информацию о процессоре:

```
cat /proc/cpuinfo
```

Теперь, зная, что у нас, к примеру, процессор на 2 ядра, запускаем ttyplot:

```
ttyplot -t "Load" -c 2 -s 2 -r 1 -u "%" -p 100 -b 0 -e 1 -l 0 -f 0 -d 0 -c 0 -a 0 -g 0 -m 0 -n 0 -o 0 -p 0 -r 0 -s 0 -t 0 -u 0 -v 0 -w 0 -x 0 -y 0 -z 0 -h 0 -V 0
```

Теперь, к примеру, можно отслеживать загрузку оперативной памяти. Для этого нужно сначала узнать, сколько у нас оперативки. Для этого есть утилита free:

```
free -m
```

Теперь запускаем ttyplot:

```
ttyplot -t "RAM" -c 1 -s 2 -r 1 -u "%" -p 100 -b 0 -e 1 -l 0 -f 0 -d 0 -c 0 -a 0 -g 0 -m 0 -n 0 -o 0 -p 0 -r 0 -s 0 -t 0 -u 0 -v 0 -w 0 -x 0 -y 0 -z 0 -h 0 -V 0
```

Аналогично, для сети — по количеству пакетов по интерфейсу. Особенно удобно смотреть, к примеру, сколько пакетов в секунду отдает ваш сервер. Для этого нужно сначала узнать, какой у вас интерфейс. Для этого есть утилита ifconfig:

```
ifconfig
```

Теперь запускаем ttyplot:

```
ttyplot -t "eth0" -c 1 -s 2 -r 1 -u "packets/sec" -p 100 -b 0 -e 1 -l 0 -f 0 -d 0 -c 0 -a 0 -g 0 -m 0 -n 0 -o 0 -p 0 -r 0 -s 0 -t 0 -u 0 -v 0 -w 0 -x 0 -y 0 -z 0 -h 0 -V 0
```

Аналогично, для сети — по количеству пакетов по интерфейсу. Особенно удобно смотреть, к примеру, сколько пакетов в секунду отдает ваш сервер. Для этого нужно сначала узнать, какой у вас интерфейс. Для этого есть утилита ifconfig:

```
ifconfig
```

Теперь запускаем ttyplot:

```
ttyplot -t "eth0" -c 1 -s 2 -r 1 -u "packets/sec" -p 100 -b 0 -e 1 -l 0 -f 0 -d 0 -c 0 -a 0 -g 0 -m 0 -n 0 -o 0 -p 0 -r 0 -s 0 -t 0 -u 0 -v 0 -w 0 -x 0 -y 0 -z 0 -h 0 -V 0
```

БАННЕРЫ С ПРОГРЕССБАРОМ

Интересная утилита. Не только встроенная в самые дистрибутивы, но почти обязательная к использованию в любом проекте, использующем консоль. Называется pv. Для ее использования достаточно установить пакет pv и использовать ее для перенаправления потока данных. Например, при копировании файлов:

```
cp -R /var/www/* /var/www/backup
```

Теперь пишем:

```
cp -R /var/www/* | pv -s 1G -p -t -e -r -b /var/www/backup
```

Теперь, при копировании, мы будем видеть прогресс-бар, который показывает прошедшее время, скорость копирования, сколько осталось времени до конца, сколько уже скопировано и сколько еще скопировать. Для того, чтобы использовать pv для отображения прогресса при сжатии архивов, нужно сначала, к примеру, сделать tar-архив:

```
tar -cf - /var/www/* | pv -s 1G -p -t -e -r -b > /var/www.tar
```

Теперь, когда у нас есть tar-архив, можно его сжать, к примеру, gzip-ом:

```
pv -p -t -e -r -b /var/www.tar | gzip > /var/www.tar.gz
```

Аналогично, pv может использоваться для отображения прогресса при распаковке. Для этого нужно сначала узнать, сколько распакованных данных занимает ваш архив. Для этого есть, к примеру, утилита gzip:

```
gzip -l /var/www.tar.gz
```

Теперь, зная, что у нас, к примеру, распакованный архив занимает 1G, запускаем:

```
pv -s 1G -p -t -e -r -b /var/www.tar.gz | tar -xf -
```

Теперь, при распаковке, мы будем видеть прогресс-бар, который показывает прошедшее время, скорость распаковки, сколько осталось времени до конца, сколько уже распаковано и сколько еще распаковать. Для того, чтобы использовать pv для отображения прогресса при копировании файлов, нужно использовать pv вместе с rsync. Для этого нужно, к примеру, сделать:

```
rsync -avz /var/www/* /var/www/backup | pv -l -s 10000 > /dev/null
```

Здесь мы говорим rsyncу, что нам нужно 10000 строк вывода, которые и будут использоваться pv для отображения прогресса. Есть еще интересная утилита — progress. Она позволяет отслеживать прогресс выполнения уже запущенных команд. Для начала нужно установить progress. В Debian/Ubuntu:

```
sudo apt-get install progress
```

Теперь, к примеру, мы запускаем какую-нибудь долгую команду. Пусть это будет, к примеру, копирование /var/www/* в /var/www/backup. Запускаем:

```
cp -R /var/www/* /var/www/backup
```

Теперь, в другом терминале, мы пишем:

```
progress -w
```

Теперь, в первом терминале, мы видим прогресс-бар, который показывает, на чем сейчас находится наша команда. Для того, чтобы использовать progress для отображения прогресса при сжатии архивов, нужно, к примеру, запустить:

```
tar -cf - /var/www/* | pv -s 1G -p -t -e -r -b > /var/www.tar
```

Теперь, в другом терминале, мы пишем:

```
progress -w
```

Теперь, в первом терминале, мы видим прогресс-бар, который показывает, на чем сейчас находится наша команда. Для того, чтобы использовать progress для отображения прогресса при распаковке, нужно, к примеру, запустить:

```
pv -s 1G -p -t -e -r -b /var/www.tar.gz | tar -xf -
```

Теперь, в другом терминале, мы пишем:

```
progress -w
```

Теперь, в первом терминале, мы видим прогресс-бар, который показывает, на чем сейчас находится наша команда. Для того, чтобы использовать progress для отображения прогресса при копировании файлов, нужно использовать pv вместе с rsync. Для этого нужно, к примеру, запустить:

```
rsync -avz /var/www/* /var/www/backup | pv -l -s 10000 > /dev/null
```

Теперь, в другом терминале, мы пишем:

```
progress -w
```

Теперь, в первом терминале, мы видим прогресс-бар, который показывает, на чем сейчас находится наша команда.

ПОИСК ИНФОРМАЦИИ

Иногда нужно быстро найти информацию в каком-нибудь справочнике. Например, существует такая утилита, как man. Но, если вы, как и я, предпочитаете не выходить из терминала, то вам, скорее всего, не хватает и справочников, открывающихся сетью в интернетах и интернетах.

Для начала нужно, к примеру, установить утилиту lynx. В Debian/Ubuntu:

```
sudo apt-get install lynx
```

Теперь, к примеру, мы хотим узнать, что такое ls. Для этого пишем:

```
lynx -dump http://man.cgi.com/cgi-bin/man/man2html?ls+1
```

Теперь, к примеру, мы хотим узнать, что такое ls. Для этого пишем:

```
lynx -dump http://man.cgi.com/cgi-bin/man/man2html?ls+1 | grep -i "[a-z]*[a-z]"
```

Теперь, к примеру, мы хотим узнать, что такое ls. Для этого пишем:

```
lynx -dump http://man.cgi.com/cgi-bin/man/man2html?ls+1 | grep -i "[a-z]*[a-z]" | wc -l
```

Теперь, к примеру, мы хотим узнать, что такое ls. Для этого пишем:

```
lynx -dump http://man.cgi.com/cgi-bin/man/man2html?ls+1 | grep -i "[a-z]*[a-z]" | wc -l | tee /tmp/ls.log
```

Теперь, к примеру, мы хотим узнать, что такое ls. Для этого пишем:

```
lynx -dump http://man.cgi.com/cgi-bin/man/man2html?ls+1 | grep -i "[a-z]*[a-z]" | wc -l | tee /tmp/ls.log | mail -s "ls" root
```

Теперь, к примеру, мы хотим узнать, что такое ls. Для этого пишем:

```
lynx -dump http://man.cgi.com/cgi-bin/man/man2html?ls+1 | grep -i "[a-z]*[a-z]" | wc -l | tee /tmp/ls.log | mail -s "ls" root | tail -f /tmp/ls.log
```

Теперь, к примеру, мы хотим узнать, что такое ls. Для этого пишем:

```
lynx -dump http://man.cgi.com/cgi-bin/man/man2html?ls+1 | grep -i "[a-z]*[a-z]" | wc -l | tee /tmp/ls.log | mail -s "ls" root | tail -f /tmp/ls.log | tee /tmp/ls.log.2
```

Теперь, к примеру, мы хотим узнать, что такое ls. Для этого пишем:

```
lynx -dump http://man.cgi.com/cgi-bin/man/man2html?ls+1 | grep -i "[a-z]*[a-z]" | wc -l | tee /tmp/ls.log | mail -s "ls" root | tail -f /tmp/ls.log | tee /tmp/ls.log.2 | wc -l
```

Теперь, к примеру, мы хотим узнать, что такое ls. Для этого пишем:

```
lynx -dump http://man.cgi.com/cgi-bin/man/man2html?ls+1 | grep -i "[a-z]*[a-z]" | wc -l | tee /tmp/ls.log | mail -s "ls" root | tail -f /tmp/ls.log | tee /tmp/ls.log.2 | wc -l | tee /tmp/ls.log.3
```

Теперь, к примеру, мы хотим узнать, что такое ls. Для этого пишем:

```
lynx -dump http://man.cgi.com/cgi-bin/man/man2html?ls+1 | grep -i "[a-z]*[a-z]" | wc -l | tee /tmp/ls.log | mail -s "ls" root | tail -f /tmp/ls.log | tee /tmp/ls.log.2 | wc -l | tee /tmp/ls.log.3 | tail -f /tmp/ls.log
```

Теперь, к примеру, мы хотим узнать, что такое ls. Для этого пишем:

```
lynx -dump http://man.cgi.com/cgi-bin/man/man2html?ls+1 | grep -i "[a-z]*[a-z]" | wc -l | tee /tmp/ls.log | mail -s "ls" root | tail -f /tmp/ls.log | tee /tmp/ls.log.2 | wc -l | tee /tmp/ls.log.3 | tail -f /tmp/ls.log | tee /tmp/ls.log.4
```

Теперь, к примеру, мы хотим узнать, что такое ls. Для этого пишем:

```
lynx -dump http://man.cgi.com/cgi-bin/man/man2html?ls+1 | grep -i "[a-z]*[a-z]" | wc -l | tee /tmp/ls.log | mail -s "ls" root | tail -f /tmp/ls.log | tee /tmp/ls.log.2 | wc -l | tee /tmp/ls.log.3 | tail -f /tmp/ls.log | tee /tmp/ls.log.4 | wc -l
```

Теперь, к примеру, мы хотим узнать, что такое ls. Для этого пишем:

```
lynx -dump http://man.cgi.com/cgi-bin/man/man2html?ls+1 | grep -i "[a-z]*[a-z]" | wc -l | tee /tmp/ls.log | mail -s "ls" root | tail -f /tmp/ls.log | tee /tmp/ls.log.2 | wc -l | tee /tmp/ls.log.3 | tail -f /tmp/ls.log | tee /tmp/ls.log.4 | wc -l | tee /tmp/ls.log.5
```

Теперь, к примеру, мы хотим узнать, что такое ls. Для этого пишем:

```
lynx -dump http://man.cgi.com/cgi-bin/man/man2html?ls+1 | grep -i "[a-z]*[a-z]" | wc -l | tee /tmp/ls.log | mail -s "ls" root | tail -f /tmp/ls.log | tee /tmp/ls.log.2 | wc -l | tee /tmp/ls.log.3 | tail -f /tmp/ls.log | tee /tmp/ls.log.4 | wc -l | tee /tmp/ls.log.5 | tail -f /tmp/ls.log
```

Теперь, к примеру, мы хотим узнать, что такое ls. Для этого пишем:

```
lynx -dump http://man.cgi.com/cgi-bin/man/man2html?ls+1 | grep -i "[a-z]*[a-z]" | wc -l | tee /tmp/ls.log | mail -s "ls" root | tail -f /tmp/ls.log | tee /tmp/ls.log.2 | wc -l | tee /tmp/ls.log.3 | tail -f /tmp/ls.log | tee /tmp/ls.log.4 | wc -l | tee /tmp/ls.log.5 | tail -f /tmp/ls.log | tee /tmp/ls.log.6
```

Теперь, к примеру, мы хотим узнать, что такое ls. Для этого пишем:

```
lynx -dump http://man.cgi.com/cgi-bin/man/man2html?ls+1 | grep -i "[a-z]*[a-z]" | wc -l | tee /tmp/ls.log | mail -s "ls" root | tail -f /tmp/ls.log | tee /tmp/ls.log.2 | wc -l | tee /tmp/ls.log.3 | tail -f /tmp/ls.log | tee /tmp/ls.log.4 | wc -l | tee /tmp/ls.log.5 | tail -f /tmp/ls.log | tee /tmp/ls.log.6 | wc -l
```

Теперь, к примеру, мы хотим узнать, что такое ls. Для этого пишем:

```
lynx -dump http://man.cgi.com/cgi-bin/man/man2html?ls+1 | grep -i "[a-z]*[a-z]" | wc -l | tee /tmp/ls.log | mail -s "ls" root | tail -f /tmp/ls.log | tee /tmp/ls.log.2 | wc -l | tee /tmp/ls.log.3 | tail -f /tmp/ls.log | tee /tmp/ls.log.4 | wc -l | tee /tmp/ls.log.5 | tail -f /tmp/ls.log | tee /tmp/ls.log.6 | wc -l | tee /tmp/ls.log.7
```

Теперь, к примеру, мы хотим узнать, что такое ls. Для этого пишем:

```
lynx -dump http://man.cgi.com/cgi-bin/man/man2html?ls+1 | grep -i "[a-z]*[a-z]" | wc -l | tee /tmp/ls.log | mail -s "ls" root | tail -f /tmp/ls.log | tee /tmp/ls.log.2 | wc -l | tee /tmp/ls.log.3 | tail -f /tmp/ls.log | tee /tmp/ls.log.4 | wc -l | tee /tmp/ls.log.5 | tail -f /tmp/ls.log | tee /tmp/ls.log.6 | wc -l | tee /tmp/ls.log.7 | tail -f /tmp/ls.log
```

Теперь, к примеру, мы хотим узнать, что такое ls. Для этого пишем:

```
lynx -dump http://man.cgi.com/cgi-bin/man/man2html?ls+1 | grep -i "[a-z]*[a-z]" | wc -l | tee /tmp/ls.log | mail -s "ls" root | tail -f /tmp/ls.log | tee /tmp/ls.log.2 | wc -l | tee /tmp/ls.log.3 | tail -f /tmp/ls.log | tee /tmp/ls.log.4 | wc -l | tee /tmp/ls.log.5 | tail -f /tmp/ls.log | tee /tmp/ls.log.6 | wc -l | tee /tmp/ls.log.7 | tail -f /tmp/ls.log | tee /tmp/ls.log.8
```

Теперь, к примеру, мы хотим узнать, что такое ls. Для этого пишем:

```
lynx -dump http://man.cgi.com/cgi-bin/man/man2html?ls+1 | grep -i "[a-z]*[a-z]" | wc -l | tee /tmp/ls.log | mail -s "ls" root | tail -f /tmp/ls.log | tee /tmp/ls.log.2 | wc -l | tee /tmp/ls.log.3 | tail -f /tmp/ls.log | tee /tmp/ls.log.4 | wc -l | tee /tmp/ls.log.5 | tail -f /tmp/ls.log | tee /tmp/ls.log.6 | wc -l | tee /tmp/ls.log.7 | tail -f /tmp/ls.log | tee /tmp/ls.log.8 | wc -l
```

Теперь, к примеру, мы хотим узнать, что такое ls. Для этого пишем:

```
lynx -dump http://man.cgi.com/cgi-bin/man/man2html?ls+1 | grep -i "[a-z]*[a-z]" | wc -l | tee /tmp/ls.log | mail -s "ls" root | tail -f /tmp/ls.log | tee /tmp/ls.log.2 | wc -l | tee /tmp/ls.log.3 | tail -f /tmp/ls.log | tee /tmp/ls.log.4 | wc -l | tee /tmp/ls.log.5 | tail -f /tmp/ls.log | tee /tmp/ls.log.6 | wc -l | tee /tmp/ls.log.7 | tail -f /tmp/ls.log | tee /tmp/ls.log.8 | wc -l | tee /tmp/ls.log.9
```

Теперь, к примеру, мы хотим узнать, что такое ls. Для этого пишем:

```
lynx -dump http://man.cgi.com/cgi-bin/man/man2html?ls+1 | grep -i "[a-z]*[a-z]" | wc -l | tee /tmp/ls.log | mail -s "ls" root | tail -f /tmp/ls.log | tee /tmp/ls.log.2 | wc -l | tee /tmp/ls.log.3 | tail -f /tmp/ls.log | tee /tmp/ls.log.4 | wc -l | tee /tmp/ls.log.5 | tail -f /tmp/ls.log | tee /tmp/ls.log.6 | wc -l | tee /tmp/ls.log.7 | tail -f /tmp/ls.log | tee /tmp/ls.log.8 | wc -l | tee /tmp/ls.log.9 | tail -f /tmp/ls.log
```

Теперь, к примеру, мы хотим узнать, что такое ls. Для этого пишем:

```
lynx -dump http://man.cgi.com/cgi-bin/man/man2html?ls+1 | grep -i "[a-z]*[a-z]" | wc -l | tee /tmp/ls.log | mail -s "ls" root | tail -f /tmp/ls.log | tee /tmp/ls.log.2 | wc -l | tee /tmp/ls.log.3 | tail -f /tmp/ls.log | tee /tmp/ls.log.4 | wc -l | tee /tmp/ls.log.5 | tail -f /tmp/ls.log | tee /tmp/ls.log.6 | wc -l | tee /tmp/ls.log.7 | tail -f /tmp/ls.log | tee /tmp/ls.log.8 | wc -l | tee /tmp/ls.log.9 | tail -f /tmp/ls.log | tee /tmp/ls.log.10
```

Теперь, к примеру, мы хотим узнать, что такое ls. Для этого пишем:

```
lynx -dump http://man.cgi.com/cgi-bin/man/man2html?ls+1 | grep -i "[a-z]*[a-z]" | wc -l | tee /tmp/ls.log | mail -s "ls" root | tail -f /tmp/ls.log | tee /tmp/ls.log.2 | wc -l | tee /tmp/ls.log.3 | tail -f /tmp/ls.log | tee /tmp/ls.log.4 | wc -l | tee /tmp/ls.log.5 | tail -f /tmp/ls.log | tee /tmp/ls.log.6 | wc -l | tee /tmp/ls.log.7 | tail -f /tmp/ls.log | tee /tmp/ls.log.8 | wc -l | tee /tmp/ls.log.9 | tail -f /tmp/ls.log | tee /tmp/ls.log.10 | wc -l
```

Теперь, к примеру, мы хотим узнать, что такое ls. Для этого пишем:

```
lynx -dump http://man.cgi.com/cgi-bin/man/man2html?ls+1 | grep -i "[a-z]*[a-z]" | wc -l | tee /tmp/ls.log | mail -s "ls" root | tail -f /tmp/ls.log | tee /tmp/ls.log.2 | wc -l | tee /tmp/ls.log.3 | tail -f /tmp/ls.log | tee /tmp/ls.log.4 | wc -l | tee /tmp/ls.log.5 | tail -f /tmp/ls.log | tee /tmp/ls.log.6 | wc -l | tee /tmp/ls.log.7 | tail -f /tmp/ls.log | tee /tmp/ls.log.8 | wc -l | tee /tmp/ls.log.9 | tail -f /tmp/ls.log | tee /tmp/ls.log.10 | wc -l | tee /tmp/ls.log.11
```

Теперь, к примеру, мы хотим узнать, что такое ls. Для этого пишем:

```
lynx -dump http://man.cgi.com/cgi-bin/man/man2html?ls+1 | grep -i "[a-z]*[a-z]" | wc -l | tee /tmp/ls.log | mail -s "ls" root | tail -f /tmp/ls.log | tee /tmp/ls.log.2 | wc -l | tee /tmp/ls.log.3 | tail -f /tmp/ls.log | tee /tmp/ls.log.4 | wc -l | tee /tmp/ls.log.5 | tail -f /tmp/ls.log | tee /tmp/ls.log.6 | wc -l | tee /tmp/ls.log.7 | tail -f /tmp/ls.log | tee /tmp/ls.log.8 | wc -l | tee /tmp/ls.log.9 | tail -f /tmp/ls.log | tee /tmp/ls.log.10 | wc -l | tee /tmp/ls.log.11 | tail -f /tmp/ls.log
```

Теперь, к примеру, мы хотим узнать, что такое ls. Для этого пишем:

```
lynx -dump http://man.cgi.com/cgi-bin/man/man2html?ls+1 | grep -i "[a-z]*[a-z]" | wc -l | tee /tmp/ls.log | mail -s "ls" root | tail -f /tmp/ls.log | tee /tmp/ls.log.2 | wc -l | tee /tmp/ls.log.3 | tail -f /tmp/ls.log | tee /tmp/ls.log.4 | wc -l | tee /tmp/ls.log.5 | tail -f /tmp/ls.log | tee /tmp/ls.log.6 | wc -l | tee /tmp/ls.log.7 | tail -f /tmp/ls.log | tee /tmp/ls.log.8 | wc -l | tee /tmp/ls.log.9 | tail -f /tmp/ls.log | tee /tmp/ls.log.10 | wc -l | tee /tmp/ls.log.11 | tail -f /tmp/ls.log | tee /tmp/ls.log.12
```

Теперь, к примеру, мы хотим узнать, что такое ls. Для этого пишем:

```
lynx -dump http://man.cgi.com/cgi-bin/man/man2html?ls+1 | grep -i "[a-z]*[a-z]" | wc -l | tee /tmp/ls.log | mail -s "ls" root | tail -f /tmp/ls.log | tee /tmp/ls.log.2 | wc -l | tee /tmp/ls.log.3 | tail -f /tmp/ls.log | tee /tmp/ls.log.4 | wc -l | tee /tmp/ls.log.5 | tail -f /tmp/ls.log | tee /tmp/ls.log.6 | wc -l | tee /tmp/ls.log.7 | tail -f /tmp/ls.log | tee /tmp/ls.log.8 | wc -l | tee /tmp/ls.log.9 | tail -f /tmp/ls.log | tee /tmp/ls.log.10 | wc -l | tee /tmp/ls.log.11 | tail -f /tmp/ls.log | tee /tmp/ls.log.12 | wc -l
```

Теперь, к примеру, мы хотим узнать, что такое ls. Для этого пишем:

```
lynx -dump http://man.cgi.com/cgi-bin/man/man2html?ls+1 | grep -i "[a-z]*[a-z]" | wc -l | tee /tmp/ls.log | mail -s "ls" root | tail -f /tmp/ls.log | tee /tmp/ls.log.2 | wc -l | tee /tmp/ls.log.3 | tail -f /tmp/ls.log | tee /tmp/ls.log.4 | wc -l | tee /tmp/ls.log.5 | tail -f /tmp/ls.log | tee /tmp/ls.log.6 | wc -l | tee /tmp/ls.log.7 | tail -f /tmp/ls.log | tee /tmp/ls.log.8 | wc -l | tee /tmp/ls.log.9 | tail -f /tmp/ls.log | tee /tmp/ls.log.10 | wc -l | tee /tmp/ls.log.11 | tail -f /tmp/ls.log | tee /tmp/ls.log.12 | wc -l | tee /tmp/ls.log.13
```

Теперь, к примеру, мы хотим узнать, что такое ls. Для этого пишем:

```
lynx -dump http://man.cgi.com/cgi-bin/man/man2html?ls+1 | grep -i "[a-z]*[a-z]" | wc -l | tee /tmp/ls.log | mail -s "ls" root | tail -f /tmp/ls.log | tee /tmp/ls.log.2 | wc -l | tee /tmp/ls.log.3 | tail -f /tmp/ls.log | tee /tmp/ls.log.4 | wc -l | tee /tmp/ls.log.5 | tail -f /tmp/ls.log | tee /tmp/ls.log.6 | wc -l | tee /tmp/ls.log.7 | tail -f /tmp/ls.log | tee /tmp/ls.log.8 | wc -l | tee /tmp/ls.log.9 | tail -f /tmp/ls.log | tee /tmp/ls.log.10 | wc -l | tee /tmp/ls.log.11 | tail -f /tmp/ls.log | tee /tmp/ls.log.12 | wc -l | tee /tmp/ls.log.13 | tail -f /tmp/ls.log
```