

# Network UPS Tools (NUT)



Network UPS Tools (NUT) - комплект программ для мониторинга ИБП.

Поддерживает мониторинг ИБП через [APCUPSd](#).

Для Linux-систем рекомендуется вместо [APCUPSd](#).

<http://www.networkupstools.org/>

 [The Network UPS Tools repository](#)

[Monitoring diagrams](#)

[GenericUPS - Driver for contact-closure UPS equipment](#)

## Настройка

Папка: `/etc/nut/`

В файле `nut.conf` задаётся режим работы:

- Сервер:

`nut.conf`

```
MODE=netserver
```

- Клиент:

`nut.conf`

```
MODE=netclient
```

Список необходимых процессов для работы:

- Сервер
  - `upsd`
  - `upsmon`
- Клиент
  - `upsmon`

Для проверки можно использовать команду:

```
ps -aux | grep ups
```

## iptables

Настройки для [IPTables](#)

```
iptables -A INPUT -p tcp -i $LAN_IF -s $LAN --sport 3493 -j ACCEPT
iptables -A OUTPUT -p tcp -o $LAN_IF -d $LAN --dport 3493 -j ACCEPT
```

```
iptables -A INPUT -p tcp -i $LAN_IF -s $LAN --dport 3493 -j ACCEPT
iptables -A OUTPUT -p tcp -o $LAN_IF -d $LAN --sport 3493 -j ACCEPT
```

## Сервер — upsd

upsd отвечает за передачу данных от драйверов к клиентским программам по сети. Его следует запускать сразу после [updrvctl](#) в сценариях запуска вашей системы.

По возможности, upsd следует поддерживать в рабочем состоянии, поскольку это единственный источник информации о состоянии для клиентов мониторинга, таких как [upsmon](#).

Установка:

```
apt-get install nut-server
```

Пакет nut-server включает в себя и [upsmon](#)

Состояние:

[nut-server\\_status.sh](#)

```
#!/bin/bash

netstat -tulpn|grep upsd
service nut-server status
```

## upsd.conf

[upsd.conf](#)

```
MAXAGE 15
MAXCONN 8
STATEPATH /var/run/nut
LISTEN 0.0.0.0 3493
#LISTEN 127.0.0.1 3493
```

Проверка:

```
netstat -an |grep 3493
```

## upsd.users

[upsd.users](#)

```
[admin]
password = admin
actions = SET
instcmds = ALL

[upsmaster]
password = password
upsmon master

[upsslave]
```

```
password = password
upsmon slave
```

## ups.conf

### ups.conf

```
[ИМЯ_ИБП]
driver = upscore2
port = /dev/ttyS0
desc = "Описание ИБП"
```

⚠️ Нужно обязательно дать права на файл устройства (порта), иначе будет ошибка:

```
upsd: Can't connect to UPS [ИМЯ_ИБП] (upscode2-ИМЯ_ИБП): No such file or directory
```

Пример:

```
chmod 0660 /dev/ttyS0
chown root:nut /dev/ttyS0
```

Для того, чтобы после перезагрузки были правильные права на файл, нужно создать файл `99-nut.rules`

Примеры:

- [/etc/udev/rules.d/99-nut.rules](#)

```
KERNEL=="ttyS0", GROUP="nut"
```

- пример для [Moxa C104H](#)

## upsdrvctl

UPS driver controller

Тест выключения

```
upsdrvctl -t shutdown
```

<http://networkupstools.org/docs/man/upsdrvctl.html>

## Клиент мониторинга — upsmon

Установка:

```
apt-get install nut-client
```

upsmon предоставляет важную функцию, которую вы ожидаете найти в ПО для мониторинга ИБП - это безопасное отключение при сбое питания.

В иерархии ПО NUT это клиент. Он отдельно описан в документации, так как он очень важен. Вы настраиваете его, указывая все ИБП, информацию о которых необходимо получать, это делается в файле в [upsmon.conf](#).

Каждый ИБП можно определить как один из двух возможных типов:

- ведущий (master)

Этот ИБП подаёт питание на работающую систему, и эта система также отвечает за её отключение, когда батарея разряжена. Это происходит после безопасного отключения всех ведомых (slave) систем.

Если ваш ИБП подключён непосредственно к последовательному порту системы, процесс [upsmon](#) в этой системе должен определить этот ИБП как ведущий (master).

У простом случае один компьютер подключён к одному ИБП. Это означает, что вы запускаете драйвер, [upsd](#) и [upsmon](#) в режиме ведущий (master).

- ведомый (slave)

Этот ИБП может подавать питание на работающую систему, но эта система не может выключить её напрямую. Используйте этот режим, когда вы используете несколько компьютеров на одном ИБП. Очевидно, только один может быть подключён к последовательному порту на ИБП, и эта система является ведущей (master), а все остальные ведомые (slave).

## upsmon.conf

```
MONITOR ИМЯ_ИБП@СЕРВЕР 1 upsslave password slave
MINSUPPLIES 1
SHUTDOWNCMD "/sbin/shutdown -h +0"
POLLFREQ 5
POLLFREQALERT 5
HOSTSYNC 15
DEADTIME 15
RBWARNTIME 43200
NOCOMMWARNTIME 300
FINALDELAY 5

NOTIFYMSG ONLINE "UPS %s on line power"
NOTIFYMSG ONBATT "UPS %s on battery"
NOTIFYMSG LOWBATT "UPS %s battery is low"
NOTIFYMSG FSD "UPS %s: forced shutdown in progress"
NOTIFYMSG COMMOK "Communications with UPS %s established"
NOTIFYMSG COMMBAD "Communications with UPS %s lost"
NOTIFYMSG SHUTDOWN "Auto logout and shutdown proceeding"
NOTIFYMSG REPLBATT "UPS %s battery needs to be replaced"
NOTIFYMSG NOCOMM "UPS %s is unavailable"
NOTIFYMSG NOPARENT "upsmon parent process died - shutdown impossible"
```

MINSUPPLIES — Число ИБП, которые поддерживают работоспособность системы.

Строка описания одного ИБП:

MONITOR СИСТЕМА PV ПОЛЬЗОВАТЕЛЬ ПАРОЛЬ ТИП	
СИСТЕМА	ИМЯ_ИБП@СЕРВЕР
PV	Число блоков питания, которые ИБП питает эту систему. Большинство обычных компьютеров имеют один блок питания, и ИБП питает его, поэтому это значение будет 1. <b>Если нужно только контролировать ИБП, который на самом деле не подаёт питание на эту систему нужно использовать значение 0.</b> Это полезно, когда вы хотите, чтобы <a href="#">upsmon</a> только опраивлял уведомления об изменениях статуса ИБП
ПОЛЬЗОВАТЕЛЬ	Имя пользователя
ПАРОЛЬ	Пароль пользователя

ТИП	Тип: master или slave
-----	-----------------------

Состояние:

[nut-client\\_status.sh](#)

```
#!/bin/bash
service nut-client status
```

Пример:

[upsmon.conf](#)

```
MONITOR ИМЯ_ИБП@localhost 1 upsmanager password master
MINSUPPLIES 1
SHUTDOWNCMD "/sbin/shutdown -h +0"
POLLFREQ 5
POLLFREQUALERT 5
HOSTSYNC 15
DEADTIME 15
POWERDOWNFLAG /etc/killpower
RBWARNTIME 43200
NOCOMMWARNTIME 300
FINALDELAY 5
```

## Другие клиенты

Клиенты общаются с [upsd](#) по сети и работают с данными из драйверов. Существуют инструменты для доступа к командной строке и несколько специальных клиентов, которые можно запускать через ваш веб-сервер как программы [CGI](#).

### upsc

Это простой клиент, который отображает значения переменных, известных upsd и вашим драйверам ИБП. По умолчанию он перечислит каждую переменную или только одну, если вы укажете дополнительный аргумент. Это может быть полезно в сценариях оболочки для мониторинга чего-либо без написания собственного сетевого кода.

upsc — это быстрый способ узнать, правильно ли работают ваши драйверы и [upsd](#).

upsc -L	список всех настроенных ИБП
upsc ИМЯ_ИБП	список всех доступных переменных с их текущими значениями
upsc ИМЯ_ИБП ИМЯ_ПЕРЕМЕННОЙ	текущее значение переменной

 Для подавления сообщения:

```
Init SSL without certificate database
```

Нужно использовать следующую конструкцию:

```
upsc ИМЯ_ИБП ИМЯ_ПЕРЕМЕННОЙ 2>&1 | grep -v SSL
```

## upslog

upslog будет записывать информацию о состоянии из [upsd](#) в файл через заданные промежутки времени. Вы можете использовать это для создания графиков или отчетов с помощью других программ.

Пример вывода протокола в файл с помощью upslog

[/etc/init.d/zzz\\_upslog](#)

```
#!/bin/bash

upslog -s ИМЯ_ИБП@ИМЯ_СЕРВЕРА -l/var/log/ups.log -i 30 -f \
"%TIME @Y@m@d @H@M@S% \
%VAR battery.charge% \
%VAR input.voltage% \
%VAR ups.load% \
[%VAR ups.status%] \
%VAR ups.temperature% \
%VAR input.frequency% \
%VAR battery.voltage% \
%VAR battery.voltage.high% \
%VAR battery.voltage.low% \
%VAR battery.voltage.nominal% \
%VAR input.current.nominal% \
%VAR input.voltage% \
%VAR input.voltage.fault% \
%VAR input.voltage.nominal% \
%VAR output.voltage% \
%VAR ups.delay.shutdown% \
%VAR ups.delay.start%"
```

## upsrw

upsrw позволяет отображать и изменять переменные чтения / записи в оборудовании вашего ИБП. Не все устройства или драйверы реализуют это, поэтому это может не повлиять на вашу систему. Драйвер, поддерживающий переменные чтения / записи, даст следующие результаты:

Для изменения значения переменных требуются прав администратора. Права пользователей задаются в [upsd.users](#)

Примеры

- Получение текущего значения переменной `battery.date`:

```
upsc ups_01 battery.date
```

- Изменение значения переменной `battery.date` в ИБП [APC](#) (формат даты: DD/MM/YY):

```
upsrw -s battery.date=30/07/21 -u admin -p password ups_01@localhost
```

## upscmd

Некоторое оборудование и драйверы ИБП поддерживают понятие мгновенной команды - такой функции, как запуск теста батареи или выключение нагрузки. Вы можете использовать `upscmd` для вывода списка или вызова мгновенных команд, если ваше оборудование/драйверы их поддерживают.

`upscmd` требует прав администратора для запуска мгновенных команд. Чтобы определить пользователей и пароли в [upsd](#), см. [upsd.users](#)

Список команд, которые поддерживаются данным ИБП:

```
upscmd -u admin -p password -l UPS_01@localhost
```

## Калибровка батарей



Процесс калибровки нужно начинать когда батареи ИБП заряжены полностью!

ИБП должен быть нагружен, глупо калибровать его на «холостом ходу».

Если есть сомнения в качестве батарей, необходимо производить калибровку не на «боевой» нагрузке, а на эквиваленте нагрузки.

В ходе этого процесса ИБП работает от батареи и разряжает их до 25%, а затем возвращается на работу от сети. ИБП использует данные нагрузки и время для вычисления новой оценки времени работы.

Запуск калибровки:

```
upscmd -u admin -p password ups_01@localhost calibrate.start
```

Для наблюдения за состоянием ИБП удобно использовать следующие команды:

- `watch upsc UPS_01 battery.charge`
- `watch upsc UPS_01 battery.runtime`

---

[Jason Healy, Director of Networks and Systems — Network UPS Tools \(Last Updated Mar 18, 2008\)](#)

## upssched



Timer helper for scheduling events from [upsmo](#)n

<https://networkupstools.org/docs/man/upssched.html>

## Веб мониторинг

```
apt-get install nut-cgi
```

## Zabbix

Мониторинг через [Zabbix](#), работает через [Zabbix агент](#).

Шаблон и код на основе этого проекта <https://github.com/delin/Zabbix-NUT-Template>

Параметр	Имя переменной	Примечания
Имя драйвера NUT	<code>driver.name</code>	
Модель ИБП	<code>ups.model</code>	

Параметр	Имя переменной	Примечания
Нагрузка (%)	ups.load	
Напряжение батарей (В)	battery.voltage	
Напряжение на входе (В)	input.voltage	
Напряжение на выходе (В)	output.voltage	
Номинальная нагрузка (ВА)	ups.power.nominal	
Производитель ИБП	ups.mfr	
Расчётное время работы от батарей (Мин.)	battery.runtime	
Состояние	ups.status	Эта переменная может содержать несколько значений, вот пример: ups.status: ALARM_0L ups.alarm: FAN_FAIL для этого введены переменные ups.status-2...ups.status-5, если эти переменные не содержат значений то возвращается 0
Состояние, значение №2	ups.status-2	
Состояние, значение №3	ups.status-3	
Состояние, значение №4	ups.status-4	
Состояние, значение №5	ups.status-5	
Тревога	ups.alarm	Эта переменная получает значение только при возникновении тревоги (alarm) поэтому для при ошибке: Error: Variable not supported by UPS возвращается 0, также она может содержать несколько значений, для этого введены переменные ups.alarm-2...ups.alarm-10
Тревога, значение №2	ups.alarm-2	
Тревога, значение №3	ups.alarm-3	
Тревога, значение №4	ups.alarm-4	
Тревога, значение №5	ups.alarm-5	
Тревога, значение №6	ups.alarm-6	
Тревога, значение №7	ups.alarm-7	
Тревога, значение №8	ups.alarm-8	
Тревога, значение №9	ups.alarm-9	
Тревога, значение №10	ups.alarm-10	
Температура (°C)	ups.temperature	
Уровень заряда батарей (В)	battery.charge	
Частота на входе (Гц)	input.frequency	

⚠ Если переменная не поддерживается, возникает ошибка:

Error: Variable not supported by UPS

и возвращается значение

NA

[Template NUT](#)

[Ссылка на Share.Zabbix.com](#)

[etc/zabbix/scripts/nut\\_ups\\_status.sh](#)

```
#!/bin/bash
ups=$1
if [ $ups = ups.discovery ]; then
    echo -e "\n\t\data\":[
    first=1
    /bin/upsc -l 2>&1 | grep -v SSL | while read discovered ; do
```

```

        if [ $first -eq 0 ]; then
            echo -e ", "
        fi
        echo -en "\t\t{ \"#{UPSNAME}\" : \t\"${discovered}\" }"
        first=0
    done
    echo -e "\n\t]\n}"

else
key=$2

if [ $key = ups.status ]; then
    state=`/bin/upsc $ups $key 2>&1 | grep -v SSL`
    case $state in
        OL)          echo 1 ;; #'On line (mains is present)' ;;
        OB)          echo 2 ;; #'On battery (mains is not present)' ;;
        LB)          echo 3 ;; #'Low battery' ;;
        RB)          echo 4 ;; #'The battery needs to be replaced' ;;
        CHR)         echo 5 ;; #'The battery is charging' ;;
        DISCHRG)    echo 6 ;; #'The battery is discharging (inverter is providing load
power)' ;;
        BYPASS)     echo 7 ;; #'UPS bypass circuit is active echo no battery
protection is available' ;;
        CAL)        echo 8 ;; #'UPS is currently performing runtime calibration (on
battery)' ;;
        OFF)        echo 9 ;; #'UPS is offline and is not supplying power to the load' ;;
        OVER)       echo 10 ;; #'UPS is overloaded' ;;
        TRIM)       echo 11 ;; #'UPS is trimming incoming voltage (called "buck" in some
hardware)' ;;
        BOOST)      echo 12 ;; #'UPS is boosting incoming voltage' ;;
        * )         echo 0 ;; #'unknown state' ;;
    esac
else
    /bin/upsc $ups $key 2>&1 | grep -v SSL
fi
fi

```

[/etc/zabbix/zabbix\\_agentd.conf.d/userparameter\\_nut.conf](#)

```
UserParameter=upsmon[*],/etc/zabbix/scripts/nut_ups_status.sh $1 $2
```

## MS Windows

 ntUPSd — Network UPS Tools Services for Windows (2016-04-28)

Клиенты:

- WinNUT — The Windows Network UPS Tools  
WinNUT is currently being replaced by a full port to Windows, part of the official NUT distribution. [MSI packages](#) are available too.  
<http://csociety.ecn.purdue.edu/~delpha/winnut/>
- Windows NUT client — графический интерфейс к WinNUT, написан на [Autolt](#)  
<http://sourceforge.net/projects/winnutclient>

## Ссылки

[Логотип](#)

[Hardware compatibility list](#)

[SDB:Перевод конфигурационных файлов NUT](#)

[Блог ИТ-отдела компании Smart Solutions: Настройка Network UPS Tools](#)

[NUT и APC Smart-UPS 1500VA](#)

[Добавление в ZABBIX UPS управляемой NUT'ом](#)

[OpenNET: статья - Подключение источника бесперебойного питания к Linux с помощью NUT-2.2.1 \(ups backup nut linux\)](#)

<http://sysadminmosaic.ru/nut/nut>

2024-02-07 18:17

