


# Настраиваем централизованное управление электропитанием в сети при помощи NUT

 [interface31.ru/tech\\_it/2022/08/nastraivaem-centralizovannoe-upravlenie-elektropitaniem-v-seti-pri-pomoshhi-nut.html](https://interface31.ru/tech_it/2022/08/nastraivaem-centralizovannoe-upravlenie-elektropitaniem-v-seti-pri-pomoshhi-nut.html)

## Записки IT специалиста

Технический блог специалистов ООО "Интерфейс"

- [Главная](#)
- Настраиваем централизованное управление электропитанием в сети при помощи NUT

Всем известно, что для защиты от сбоев электропитания нужно купить бесперебойник. Но сам по себе источник бесперебойного питания не решает проблему, а только отсрочивает негативные последствия. Действительно, если питание пропадет надолго, то после разрядки батарей ИБП ваши системы аварийно завершат работу. Избежать этого позволяют модели, имеющие обратную связь, но, чтобы использовать эти возможности нам потребуется система управления электропитанием и сегодня мы расскажем об открытом и бесплатном продукте - **NUT** (*Network UPS Tools*).



### Онлайн-курс по устройству компьютерных сетей

На углубленном курсе "[Архитектура современных компьютерных сетей](#)" вы с нуля научитесь работать с Wireshark и «под микроскопом» изучите работу сетевых протоколов. На протяжении курса надо будет выполнить более пятидесяти лабораторных работ в Wireshark.

Почему именно NUT? Ведь многие производители ИБП предлагают собственные утилиты, в т.ч. и под Linux, при этом они явно лучше знают свою продукцию. В теории это так, но практика вовсе не так радужна. Во-первых, качество "родного" ПО очень сильно отличается от вендора к вендору. Где-то это вполне неплохие решения, а где-то люта́я мешанина скриптов, бинарников и устаревших технологий.

Кроме того, ИБП у вас может быть не один, причем разных производителей и устраивать зоопарк проприетарного ПО - не самый лучший вариант. Что касается второго аргумента, то существует **UPS management protocol - RFC 9271**, если бесперебойник его поддерживает, то все равно какое ПО мы будем использовать для работы с ним.

NUT - это открытое программное обеспечение, поддерживающее большое количество источников бесперебойного питания и предоставляющее единый интерфейс работы с ними, кроме того оно работает в клиент-серверном режиме что позволяет организовать сетевое управление электропитанием, это удобно, когда к

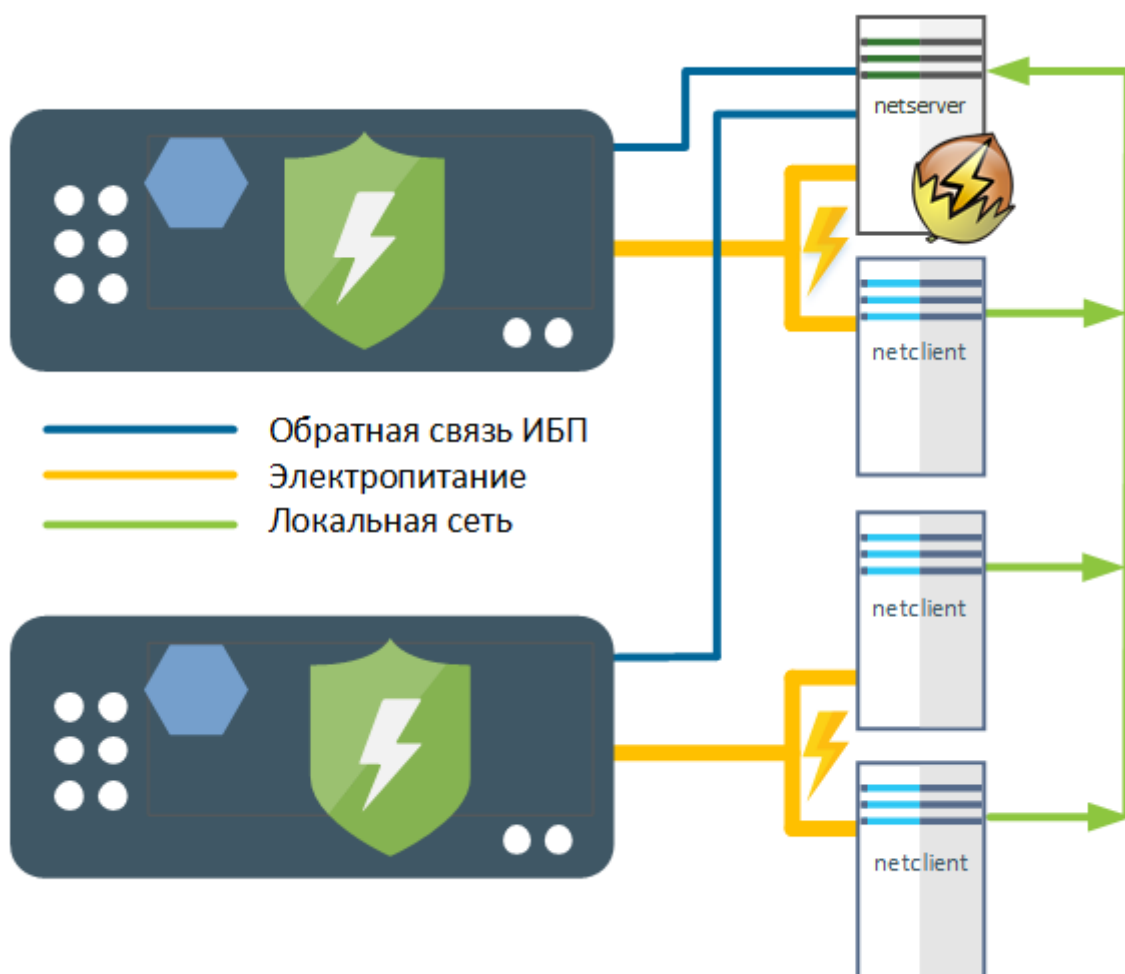
одному бесперебойнику подключено несколько компьютеров. С полным списком поддерживаемых устройств вы можете ознакомиться на официальном сайте:  
<https://networkupstools.org/stable-hcl.html>

Структурно NUT состоит из трех основных частей:

- **Драйвер** - используется для связи с ИБП.
- **Сервер (upsd)** - при помощи драйвера связывается с ИБП и получает сообщения об его состоянии.
- **Служба мониторинга (upsmmon)** - контролирует сервер и выполняет действия на основе полученной от сервера информации.

**Важный момент!** Сервер NUT никак не связывается с клиентами и не передает им никаких команд, все решения принимает клиент, который при помощи службы мониторинга получает с сервера состояние выбранного ИБП и сообщения от него, на основании которых затем предпринимаются некоторые действия.

Таким образом один сервер NUT может контролировать сразу несколько физически подключенных к нему через интерфейс обратной связи ИБП и предоставлять информацию об их состоянии клиентам через локальную сеть. Это позволяет реализовывать достаточно сложные схемы управления электропитанием, наподобие приведенной ниже:



Таким образом имея единственный сервер NUT мы получаем централизованную систему управления питанием, которая обеспечит корректное выключение всех подключенных к ИБП компьютеров, а затем самого ИБП. При возобновлении электропитания сначала включатся ИБП, а затем все подключенные к ним устройства (для этого они должны иметь соответствующие настройки в BIOS).

## Установка NUT и настройка драйвера ИБП

---

Установка NUT проста, в Debian / Ubuntu существует одноименный метапакет, который устанавливает сразу и сервер, и клиент. В других системах обратитесь к документации по доступным пакетам.

```
apt install nut
```

Теперь подключим к серверу ИБП и перезагрузим систему, после чего попробуем автоматически определить его настройки при помощи служебного пакета **nut-scanner**:

```
nut-scanner
```

Утилита просканирует все доступные интерфейсы и выведет готовые параметры подключения для обнаруженных устройств, в нашем случае был найден ИБП серии **PowerCom Smart King Pro+** подключенный через USB.

```
root@ups-mon:/home/zaliv# nut-scanner
SNMP library not found. SNMP search disabled.
Neon library not found. XML search disabled.
IPMI library not found. IPMI search disabled.
Scanning USB bus.
No start IP, skipping NUT bus (old connect method)
[nutdev1]
    driver = "usbhid-ups"
    port = "auto"
    vendorid = "0D9F"
    productid = "0004"
    bus = "004"
```

Фактически это готовые опции для настройки драйвера, советуем сохранить их для использования в дальнейшем. Теперь откроем файл **/etc/nut/ups.conf** и в самый конец добавим секцию, аналогичную указанной выше. В квадратных скобках указываем желаемое имя, ниже следуют опции подключения. В минимальной конфигурации будет достаточно первых двух: **driver** и **port**. Если предполагается использовать несколько ИБП, то мы можем уточнить производителя и модель: **vendorid** и **productid**, а вот опцию **bus** - лучше не использовать, при переключении ИБП в другой разъем она изменится. В случае использования нескольких ИБП одной модели лучше использовать другую опцию - **serial** с указанием серийного номера устройства.

Также мы советуем изучить документацию производителя, многие из них указывают рекомендуемые конфигурации драйвера для работы с NUT, например, для PowerCom у нас вышло следующее:

```
[powercom1]
driver = usbhid-ups
port = auto
vendorid = 0d9f
productid = 0004
serial = AAA-BBBB-CCC
pollonly
```

Последняя опция как раз взята из рекомендаций производителя, без нее ИБП работал с NUT нестабильно, периодически отваливаясь, что требовало либо его переподключения, либо перезагрузки компьютера.

Теперь проверим работу драйвера:

```
upsdrvctl start
```

Если все сделано правильно, то вы увидите что-то подобное:

```
root@ups-mon:/lib/nut# upsdrvctl start
Network UPS Tools - UPS driver controller 2.7.4
Network UPS Tools - Generic HID driver 0.41 (2.7.4)
USB communication driver 0.33
Using subdriver: PowerCOM HID 0.5
root@ups-mon:/lib/nut# |
```

Как видим, драйвер запустился нормально и увидел наш ИБП, теперь остановим его, чтобы избежать конфликтов при дальнейшей настройке.

```
upsdrvctl stop
```

Если же драйвер запускается с ошибками попробуйте переключить ИБП в другой разъем или перезагрузить компьютер.

## Настройка сервера NUT

---

Прежде всего откроем **/etc/nut/nut.conf** и установим режим работы:

```
MODE=netserver
```

Для сервера доступны режимы:

- **standalone** - сервер запущен локально и принимает соединения только на localhost, используется для управления локально подключенным ИБП, который не питает других ПК,
- **netserver** - сетевой режим, сервер принимает соединения по локальной сети, подходит для построения сетевой системы управления питанием.

Мы сразу будем настраивать сетевой режим, большинство настроек при этом не будут отличаться от локальных, поэтому вы также можете настроить и локальный режим по этой инструкции.

Теперь перейдем в **/etc/nut/upsd.conf**, в нем нам потребуется указать сетевые опции, а именно адрес и порт, на котором будет принимать подключения сервер:

```
LISTEN 127.0.0.1 3493
LISTEN 192.168.99.120 3493
```

Обратите внимание на регистр написания, название опции должно быть обязательно в верхнем регистре, для локального режима достаточно будет только указания первой строки.

Следующим шагом заведем пользователей в файле **/etc/nut/upsd.users**. Мы рекомендуем, как минимум, завести административного пользователя, который будет иметь права менять настройки UPS и пользователей для чтения данных состояния бесперебойника. Вы можете завести разные учетные данные для разных ПК или одну на всех - выбор остается за вами. В нашем случае содержимое файла выглядит так:

```
[upsadmin]
password = Pa$$word_1
    actions = SET
    instcmds = ALL
```

```
[upsmon-m]
password = 123456789
    upsmon master
```

```
[upsmon-s]
    password = 987654321
    upsmon slave
```

В нашем случае мы завели трех пользователей: административного, пользователя локального клиента (master) и удаленных клиентов (slave).

Сохраним все изменения и попробуем запустить службу сервера и сразу проконтролируем ее статус:

```
systemctl start nut-server
systemctl status nut-server
```

```

root@ups-mon:/etc/nut# systemctl status nut-server
● nut-server.service - Network UPS Tools - power devices information server
   Loaded: loaded (/lib/systemd/system/nut-server.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2022-08-18 19:39:32 MSK; 19s ago
     Main PID: 644 (upsd)
        Tasks: 1 (limit: 2340)
       Memory: 676.0K
          CPU: 3ms
      CGroup: /system.slice/nut-server.service
              └─644 /lib/nut/upsd

abr 18 19:39:32 ups-mon upsd[643]: fopen /run/nut/upsd.pid: No such file or directory
abr 18 19:39:32 ups-mon upsd[643]: listening on 192.168.99.120 port 3493
abr 18 19:39:32 ups-mon upsd[643]: listening on 192.168.99.120 port 3493
abr 18 19:39:32 ups-mon upsd[643]: listening on 127.0.0.1 port 3493
abr 18 19:39:32 ups-mon upsd[643]: listening on 127.0.0.1 port 3493
abr 18 19:39:32 ups-mon upsd[643]: Connected to UPS [powercom1]: usbhid-ups-powercom1
abr 18 19:39:32 ups-mon upsd[643]: Connected to UPS [powercom1]: usbhid-ups-powercom1
abr 18 19:39:32 ups-mon upsd[644]: Startup successful
abr 18 19:39:32 ups-mon systemd[1]: Started Network UPS Tools - power devices information server.
abr 18 19:39:40 ups-mon upsd[644]: User upsmon-m@127.0.0.1 logged into UPS [powercom1]

```

Теперь можем получить данные о состоянии ИБП командой:

```
upsc powercom1@localhost
```

Где мы указываем имя нашего драйвера и имя или адрес узла, на котором запущена серверная часть NUT.

В выводе мы можем увидеть частоту и напряжение на входе и на выходе устройства, текущую нагрузку в процентах (ups.load) и статус бесперебойника, который может иметь следующие значения:

- **OL** - On Line - работа от сети
- **OB** - On Battery - работа от батареи
- **LB** - Low Battery - низкий заряд батареи

Как мы уже говорили, сам сервер не предпринимает никаких действий, а только получает данные от ИБП и предоставляет их клиентам, но именно состояние LB служит триггером для службы мониторинга инициирующим выключение устройств.

## Настройка службы мониторинга в Linux

Начнем с локального сервера, клиентская часть здесь уже установлена, поэтому просто откроем файл **/etc/nut/upsmon.conf** и добавим в него строку:

```
MONITOR powercom1@localhost 1 upsmon-m 123456789 master
```

Разберем указанные опции: **powercom1@localhost** - указывают имя драйвера и сервер NUT, следующая за ними единица говорит о том, что питание устройства следует отключить, для отладки можно временно указать там ноль, в этом случае будут генерироваться все события и записи в логах, но отключение питания производиться не будет. Затем указываем имя пользователя и пароль, а также тип устройства - **master**, это говорит о том, что перед нами головное устройство и оно будет выключено только после того, как выключатся все подчиненные компьютеры.

Запустим службу мониторинга:

```
systemctl start nut-monitor
```

Остальные параметры можно оставить по умолчанию. Но следует обратить внимание на опции **NOTIFYFLAG**, которые описывают реакцию службы на те или иные события. По умолчанию, когда опция закомментирована, применяются действия **SYSLOG** и **WALL**, т.е. пишется событие в системный лог и отправляется сообщение всем залогиненным пользователям.

Дополнительно можно включить выполнение скрипта действием **EXEC**, сам скрипт общий для всех событий и указывается в опции **NOTIFYCMD**, само событие и реакцию на него следует определять в самом скрипте, но это уже выходит за рамки данной статьи, поэтому рекомендуем обратиться к документации.

На удаленных устройствах нам нет необходимости устанавливать все пакеты, достаточно только клиентского:

```
apt install nut-client
```

После чего откроем **/etc/nut/nut.conf** и укажем режим работы:

```
MODE=netclient
```

Затем в **/etc/nut/upsmon.conf** внесем строку:

```
MONITOR powercom1@192.168.99.120 1 upsmon-s 987654321 slave
```

Мы не будем подробно разбирать синтаксис, так как это сделали выше, отметим изменения. При указании драйвера и узла мы использовали IP-адрес сервера NUT (можно также его FQDN), а в конце строки указали **slave**, что обозначает подчиненный мониторинг, кроме состояния ИБП он отслеживает также команды от головного мониторинга (master). Понятно, что в качестве драйвера следует указывать именно тот ИБП, к которому физически подключен данный узел.

После чего точно также запускаем службу:

```
systemctl start nut-monitor
```

Для проверки получим сведения об ИБП:

```
upsc powercom1@192.168.99.120
```

Для просмотра подключенных клиентов выполните команду:

```
upsc -c powercom1@192.168.99.120
```

```
root@pve-egais2:~# upsc -c powercom1@192.168.99.120
Init SSL without certificate database
192.168.99.3
192.168.99.5
192.168.99.8
192.168.99.202
192.168.99.9
192.168.99.22
192.168.99.4
127.0.0.1
```



А теперь как это все работает, мы не будем углубляться в подробности, а представим максимально упрощенную схему, достаточную для понимания происходящих событий.

- Батареи ИБП разряжаются, и он переходит в состояние LB
- Мониторинг master-узла получает этот статус и выставляет флаг **FSD** (forced shutdown)
- Мониторинг slave-узлов видит этот флаг и инициирует выключение
- Master-узел ждет отключения подчиненных узлов и выставляет флаг **POWERDOWNFLAG**
- При необходимости master завершает свою работу и выключает ИБП

Последний момент очень важен, потому что если после того, как slave-узлы будут выключены, подача питания возобновится, то ИБП не выключится и повторного запуска подключенных устройств не произойдет. Поэтому master узел в обязательном порядке выключает ИБП, чтобы при его последующем включении произошел запуск всех подключенных к нему устройств.

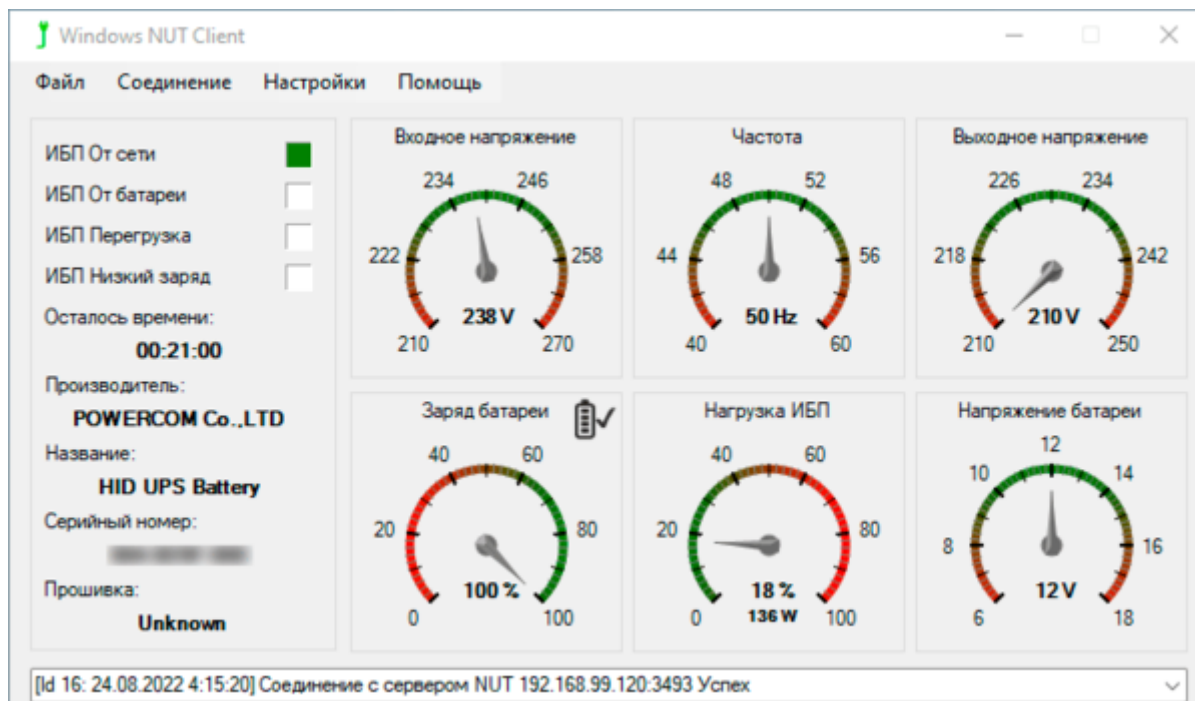
## Настройка службы мониторинга в Windows

Сервер NUT может управлять и Windows-клиентами, для этого нужно установить один из клиентов NUT, мы используем WinNUT-Client, он поставляется в виде MSI-пакета и проблем с его установкой возникнуть не должно. Настройки его также просты: указываем адрес и порт сервера, имя драйвера, логин и пароль.



Также советуем пробежаться по прочим вкладкам и настроить автозапуск приложения и выполнить, при желании, более тонкую подстройку. Кроме выполнения своей основной функции приложение также показывает основные параметры ИБП в виде графических индикаторов.





Как видим, настроить NUT просто, но данная статья и близко не затронула всех возможностей этой системы. Пока что мы организовали базовую систему централизованного управления электропитанием, но при желании ее можно бесконечно улучшать и дополнять, благо проект прекрасно документирован.