FinMiner by FINOM
version: 2.1

FinMiner это программный продукт, разработанным компанией FINOM для создания структурных криптовалютных единиц на основе алгоритмов Ethash и CryptoNight. Настоящая версия **FinMiner** создана для работы со всеми криптовалютами, основанными на указанных алгоритмах, таких, как Ethereum, Ethereum Classic, Monero, Electroneum и пр. Данная версия **FinMiner** разработана для работы в ОС Windows и Linux с видеокартами Nvidia.

Испытания **FinMiner** показали высокую эффективность при работе с Ethereum, Ethereum Classic, Electroneum и другими криптовалютами. По результатам проведенных исследований, производительность **FinMiner** в целом не уступает, а часто даже превосходит производительность конкурентных программных продуктов. При этом **FinMiner** отличается высокой стабильностью и простотой настройки.

Оплата

Оплата пользования **FinMiner** осуществляется посредством комиссионного майнинга. Комиссия составляет 1% от общего времени майнинга: каждый час **FinMiner** в случайное время на 36 секунд переключает майнинг на кошельки, принадлежащие компании Finom.

Настройка

При запуске **FinMiner** читает файл настроек *config.ini* из текущего каталога программы. Чтобы задать произвольное имя конфигурационного файла, его следует написать первым аргументом в командной строке, например:

finminer.exe config_etc.ini

При использовании **FinMiner** указывать пулы в конфигурационном файле не обязательно. Если пул (список пулов) не указан, **FinMiner** в автоматическом режиме будет использовать пулы nanopool.org, соответствующие выбранной криптовалюте.

При старте **FinMiner** выводит в консольный лог основную рабочую информацию, включающую текущую версию программы, имя рига, количество и тип установленных видеокарт, а также сведения о текущих настройках программы.

Log-файлы

Во время работы **FinMiner** создает log-файлы и складывает их в папку logs текущего каталога программы.

Параметры

Настройки **FinMiner** находятся в конфигурационном файле с расширением *.ini (по умолчанию *config.ini*) в виде логических пар «параметр=значение». В данном файле допускаются пустые строки и комментарии. Строки с комментариями должны начинаться со знака **;** (точка с запятой). Параметры и значения не учитывают регистр ввода, то есть для программы нет разницы между ETH, eth, Eth.

Ниже приведен список параметров, которые можно задать в **FinMiner**:

wallet

Обязательный параметр — кошелек пользователя, на который будут переводиться средства.

paymentId

Необязательный параметр, может быть задан для кошельков, созданных на бирже, где, кроме самого кошелька, у пользователя есть еще личный платежный номер.

algorithm

Необязательный параметр, может принимать значения «Ethash» или «Cryptonight». Если этот параметр не указан, **FinMiner** определит алгоритм на основании используемой криптовалюты или формата указанного кошелька.

coin

Необязательный параметр, может принимать значения ETH, ETC, XMR, ETN для Ethereum, Ethereum Classic, Monero и Electroneum соответственно. При указании монеты **FinMiner** автоматически поймет алгоритм и сам определит необходимые для работы пулы (nanopool.org), если они не заданы отдельным параметром.

rigname

Необязательный параметр, имя рига (компьютера/рабочей машины). Отображается в статистике пула. Если этот параметр не задан, для данного рига программа сгенерирует уникальное имя и передаст на пул.

email

Необязательный параметр, электронная почта пользователя. Передается на пул, в котором будет работать данный риг. Пул может использовать ее для рассылки сервисных сообщений.

pool1, pool2, …

Необязательные параметры, пулы для майнинга. Значения в виде url:port (например: «pool1=eth-eu1.nanopool.org:9999»). Параметры должны быть указаны в порядке возрастания, с последовательной нумерацией от pool1 до poolN (например: pool1, pool2, pool3).

Конфигурационный файл

Чтобы начать работу с **FinMiner**, в конфигурационном файле достаточно указать только ваш кошелек.

Например, для заработка Ethereum конфигурационный файл будет выглядеть так:

```
wallet=<кошелёк>
```

FinMiner автоматически подставит пулы для Ethereum.

Чтобы работать с Ethereum Classic, следует добавить указание монеты:

```
wallet=<кошелёк>
coin=ETC
```

В этом случае **FinMiner** будет использовать пулы, соответствующие Ethereum Classic.

Аналогично, чтобы майнить Monero или Electroneum, следует указать кошелек и платежный номер (при наличии). Если длина кошелька 95 символов или больше, то **FinMiner** автоматически поймет, что это кошелек для алгоритма CryptoNight. Если кошелек Electroneum начинается с «etn», **FinMiner** автоматически определит алгоритм и пулы без указания монеты.

ВАЖНО!

Для монет на алгоритмах Ethash и CryptoNight, не поддерживаемых nanopool.org, **в обязательном** порядке следует указать кошелек (**wallet**), платежный номер (при наличии) (**paymentId**), алгоритм (**algorithm**) и пулы (**pool1…**). В этом случае **FinMiner** будет функционировать по принципу работы Ethereum или Monero, но результаты операций будут определяться заданиями, приходящими с соответствующих пулов, которые указаны в файле конфигурации.

Примеры конфигурационных файлов

Пример полного файла конфигурации для Ethereum:

```
wallet = 0xffffffffffffffffffffffffffffffffffffffff
algorithm = Ethash
rigName = rig1
email = someemail@org
pool1 = eth-eu1.nanopool.org:9999
pool2 = eth-eu2.nanopool.org:9999
pool3 = eth-us-east1.nanopool.org:9999
pool4 = eth-us-west1.nanopool.org:9999
pool5 = eth-asia1.nanopool.org:9999
```

Пример эквивалентного файла для Ethereum:

```
wallet = 0xffffffffffffffffffffffffffffffffffffffff
rigName = rig1
email = someemail@org
```

Пример минимального файла для Ethereum:

```
wallet=0xffffffffffffffffffffffffffffffffffffffff
```

Пример полного файла для Ethereum Classic:

```
wallet = 0xffffffffffffffffffffffffffffffffffffffff
algorithm = Ethash
coin=Etc
rigName = rig1
email = someemail@org
pool1 = etc-eu1.nanopool.org:19999
pool2 = etc-eu2.nanopool.org:19999
pool3 = etc-us-east1.nanopool.org:19999
pool4 = etc-us-west1.nanopool.org:19999
pool5 = etc-asia1.nanopool.org:19999
```

Пример эквивалентного файла для Ethereum Classic:

```
wallet = 0xffffffffffffffffffffffffffffffffffffffff
coin=Etc
rigName = rig1
email = someemail@org
```

Пример минимального файла для Ethereum Classic:

```
wallet=0xffffffffffffffffffffffffffffffffffffffff
coin=Etc
```

Пример полного файла для Monero:

```
wallet = XXXXXXXXXX
paymentId=YYYYYYYYYY
algorithm = Cryptonight
rigName = rig1
email = someemail@org
pool1 = xmr-eu1.nanopool.org:14433
pool2 = xmr-eu2.nanopool.org:14433
pool3 = xmr-us-east1.nanopool.org:14433
pool4 = xmr-us-west1.nanopool.org:14433
pool5 = xmr-asia1.nanopool.org:14433
```

Пример эквивалентного файла для Monero:

```
wallet = XXXXXXXXXX
paymentId=YYYYYYYYYY
rigName = rig1
email = someemail@org
```

Пример минимального файла для Monero:

```
wallet = XXXXXXXXXX
```

Пример полного файла для Electroneum:

```
wallet = etnXXXXXXXXXX
paymentId = YYYYYYYYYY
algorithm = Cryptonight
coin = ETN
rigName = rig1
email = someemail@org
pool1 = etn-eu1.nanopool.org:13433
pool2 = etn-eu2.nanopool.org:13433
pool3 = etn-us-east1.nanopool.org:13433
pool4 = etn-us-west1.nanopool.org:13433
pool5 = etn-asia1.nanopool.org:13433
```

Пример эквивалентного файла для Electroneum:

```
wallet = etnXXXXXXXXXX
paymentId = YYYYYYYYYY
coin = ETN
```

```
rigName = rig1
email = someemail@org
```

Пример минимального файла для Electroneum:

```
wallet = etnXXXXXXXXXX
```