

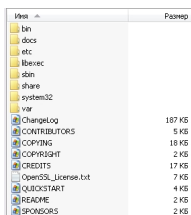
На сегодняшний день последней стабильной версией проxy-сервера **Squid под Windows** является 2.7.STABLE8. Скачать уту и другие версии Squid под OS Windows можно по здесь:

<http://squid.acmeconsulting.it/>

. Там же можно найти и версию Squid 3.0., тоже стабильную, но носящую статус «Экспериментальной» и еще не поддерживаемой разработчиком. Поэтому остановимся на версии 2.7.STABLE8 и будем рассматривать в статье именно ее.

Скачиваем дистрибутив и распаковываем его в папку c:squid

После распаковки мы должны иметь в папке c:squid следующую структуру файлов:



Имя	Размер
bin	
docs	
etc	
libexec	
sbin	
share	
system32	
var	
ChangeLog	187 КБ
CONTRIBUTORS	5 КБ
COPYING	18 КБ
COPYRIGHT	2 КБ
CREDITS	17 КБ
OpenSSL_license.txt	7 КБ
README	4 КБ
SPONSORS	2 КБ

Теперь установим squid как системную службу Windows. Для этого запускаем командную строку и набираем в ней команду:

```
c:squidbinsquid -i -f C:squidetcsquid.conf -n Squid
```

После выполнения этой команды мы должны увидеть следующее сообщение:

```
Registry stored HKLM\SOFTWARE\GNU\Squid\2.6\Squid\ConfigFile value C:\squid\etc\squid.conf
```

```
Squid Cache version 2.7.STABLE7 for i686-pc-winnt
```

```
installed successfully as Squid Windows System Service.
```

To run, start it from the Services Applet of Control Panel.

Don't forget to edit squid.conf before starting it.

Т.е. система сообщает, что в соответствующий параметр реестра записано расположение конфигурационного файла squid (C:squidetc\squid.conf), Squid успешно установлен как Сервис под Windows, а также нам напоминает, чтобы мы не забыли отредактировать конфигурационный файл squid.conf под свою систему. Собственно этим и займемся.

Переходим в директорию C:squidetc и переименовываем все находящиеся в ней файлы:

cachemgr.conf.default => cachemgr.conf

mime.conf.default => mime.conf

squid.conf.default => squid.conf

squid\_radius\_auth.conf.default - можно не переименовывать, если вы не предполагаете использовать аутентификацию по протоколу radius для пользователей прокси-сервера squid.

Далее открываем в любом текстовом редакторе файл squid.conf. Настроим минимально необходимую рабочую конфигурацию (в дальнейшем вы сможете донстроить и оптимизировать конфигурацию под свои нужды). Итак, в файле squid.conf

раскомментируем (убираем символ #) или дописываем следующие строчки:

```
acl all src all
```

```
acl manager proto cache_object
```

```
acl localhost src 127.0.0.1/32
```

```
acl to_localhost dst 127.0.0.0/8 0.0.0.0/32
```

```
# Оставляем ниже одну из записей для своей локальной сети, если прокси-сервер используется
```

```
# только на одной машине, на которой он и установлен, то эти записи нужно закомментировать
```

```
acl localnet src 10.0.0.0/8 # RFC1918 possible internal network
```

```
acl localnet src 172.16.0.0/12 # RFC1918 possible internal network
```

```
acl localnet src 192.168.0.0/16 # RFC1918 possible internal network
```

```
#
```

```
acl SSL_ports port 443
```

```
acl Safe_ports port 80          # http
```

```
acl Safe_ports port 21         # ftp
```

```
acl Safe_ports port 443       # https
```

```
acl Safe_ports port 70        # gopher
```

```
acl Safe_ports port 210       # wais
```

```
acl Safe_ports port 1025-65535 # unregistered ports
```

```
acl Safe_ports port 280       # http-mgmt
```

```
acl Safe_ports port 488       # gss-http
```

```
acl Safe_ports port 591       # filemaker
```

```
acl Safe_ports port 777       # multiling http
```

```
acl CONNECT method CONNECT
```

```
# Only allow cachemgr access from localhost
```

```
http_access allow manager localhost
```

```
http_access deny manager
```

```
# Deny requests to unknown ports
```

```
http_access deny !Safe_ports
```

```
# Deny CONNECT to other than SSL ports
```

```
http_access deny CONNECT !SSL_ports
```

```
http_access allow localnet
```

```
# And finally deny all other access to this proxy
```

```
http_access deny all
```

```
#Allow ICP queries from local networks only
```

```
icp_access allow localnet
```

```
icp_access deny all
```

# Прописываем порт, на который сквид будет ожидать соединения от клиентов.

**http\_port 3128**

#We recommend you to use at least the following line.

hierarchy\_stoplist cgi-bin ?

#Default:

cache\_replacement\_policy lru

# Прописываем расположение директории, в которой будет храниться наш кэш:

**cache\_dir ufs c:/squid/var/cache 100 16 256**

store\_dir\_select\_algorithm least-load

max\_open\_disk\_fds 0

minimum\_object\_size 0 KB

maximum\_object\_size 4096 KB

cache\_swap\_low 90

cache\_swap\_high 95

update\_headers on

access\_log c: /

logfile\_daemon c: /squid/libexec/logfile-daemon.exe

cache\_log c: /squid/var/logs/cache.log

cache\_store\_log c: /squid/var/logs/store.log

logfile\_rotate 10

emulate\_httpd\_log off

log\_ip\_on\_direct on

mime\_table c:/squid/etc/mime.conf

log\_mime\_hdrs off



```
pid_filename c:/squid/var/logs/squid.pid
```

```
log_fqdn off
```

```
strip_query_terms on
```

```
buffered_logs off
```

Таким образом, мы определили в конфигурационном файле три базовых блока - управление доступом (подсети, порты, асl-листы), настройка КЭШа, настройка логов, а также некоторые системные настройки (кстати, не все обязательные). Также обратите внимание на то, что знак в именах директорий нужно писать именно на юниксовый манер «/», а не Windows - «\».

Теперь сформируем структуры КЭШа прокси-сервера squid, выполнив команду:

```
c:squidbinsquid -z -f C:squidetcsquid.conf
```

После выполнения этой команды мы должны увидеть следующее сообщение:

```
Creating Swap Directories
```

Т.е., все успешно кэш-директория создана. Также обратите внимание на то, что при выполнении этой команды сквид проверяет конфигурационный файл `squid.conf` на корректность, и если в нем что-то написано не так, то выдалось бы соответствующее сообщение. Для создания кэш-файла мы использовали тип файловой системы `ufs` (опция в конф. Файле - `cache_dir ufs c:/squid/var/cache 100 16 256`). Также можно использовать тип `awin32`, посмотрите документацию и выберете наиболее подходящий для Вас. Также полезно заглянуть в лог-файл `C:squidvarlogscache.log` и посмотреть нет ли там каких-либо ошибок.

Теперь делаем первый пробный старт прокси-сервера Сквид. Запустим его для начала в виде консольного приложения - в случае каких-либо ошибок они будут выведены на экран и нужно будет исправить соответствующие опции в конфигурационном файле.

```
start c:squidbinsquid -f C:squidetcsquid.conf
```



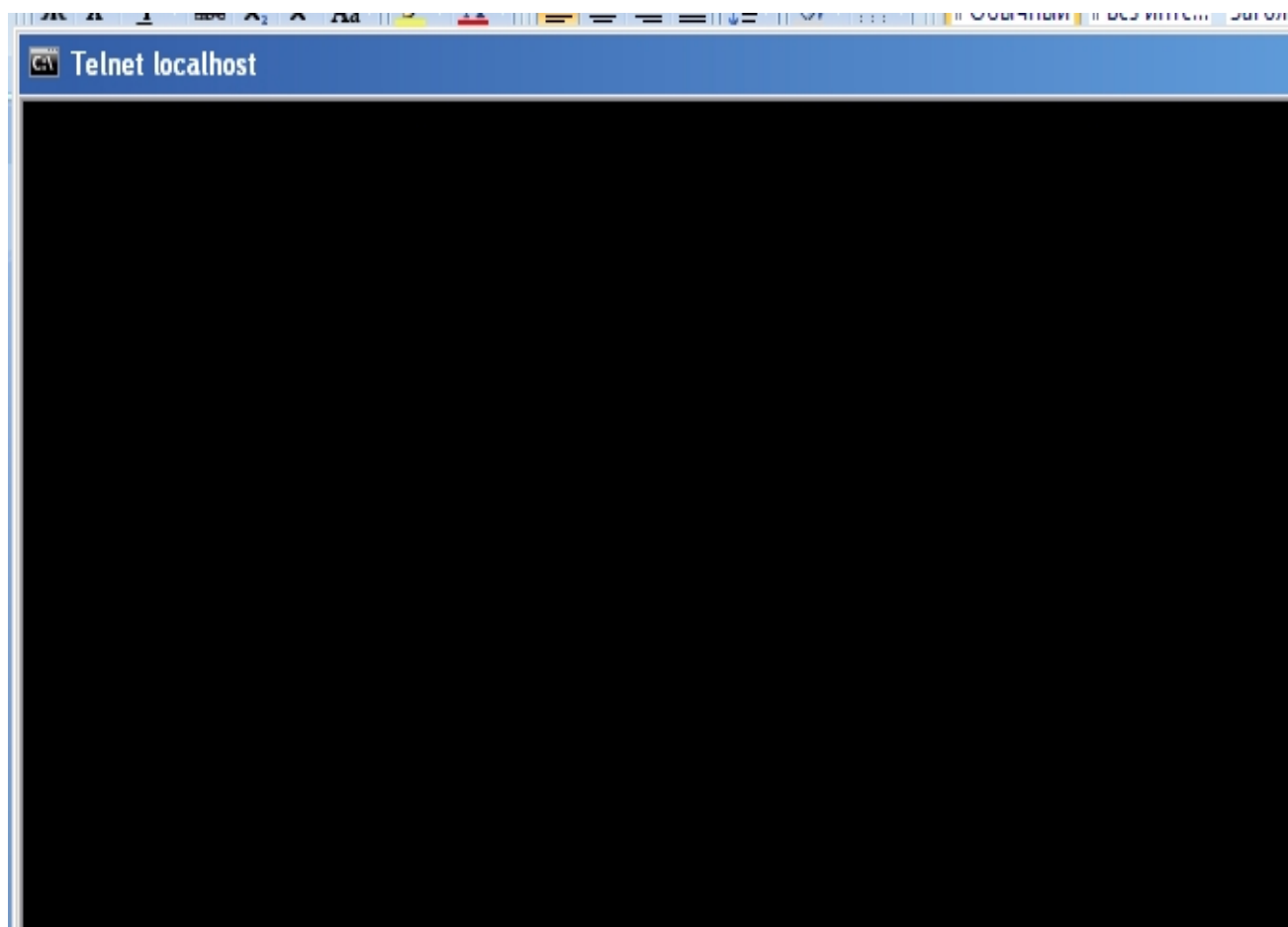
```
C:\> c:\squid\bin\squid.exe

C:\squid>start c:\squid\bin\squid -f C:\squid\etc\squid.conf
C:\squid>
```

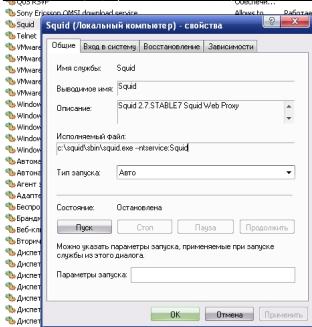
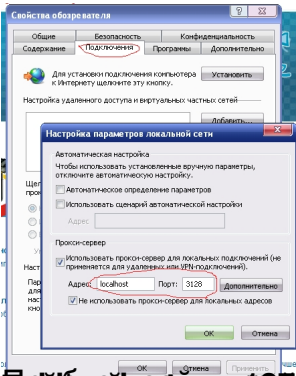
Все отлично, ошибок никаких не отобразилось на экране. По идее, теперь наш **Squid** гот

ов принять первого клиента. Попробуем к нему обратиться, выполнив команду:

**telnet localhost 3128**



Все отлично, мы как говорится «провалились», т.е. вошли на порт 3128, который мы сконфигурировали ранее для приема клиентских запросов по протоколу http, никаких ошибок подключения выдано на экран не было. Теперь можно попробовать настроить браузер.



...завульгарить прокси-время от времени - то можно создать пару ...