

Linux jako serwer dial-up, callback oraz dial-on-demand

Linux ze słuchawką?

Autor: [Marcin Krawiec](#)

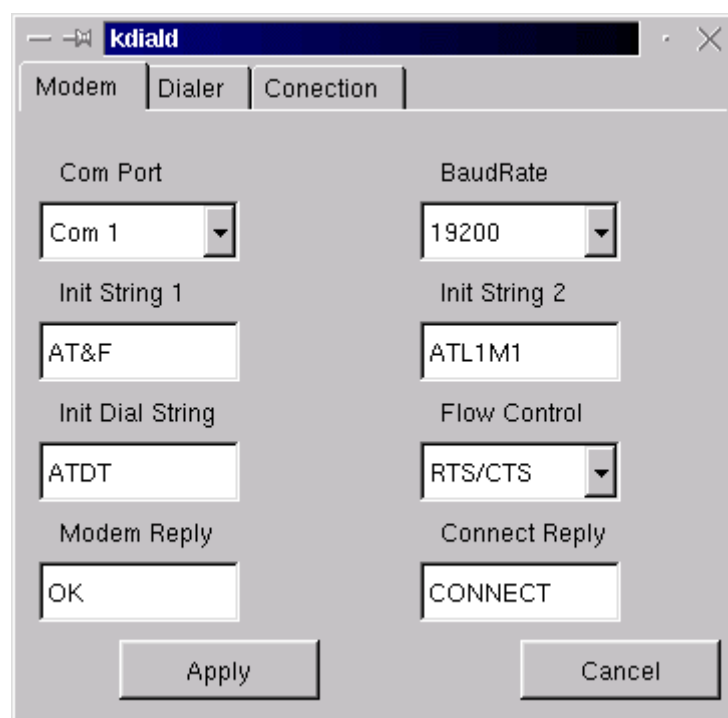
Skonfigurowanie usługi oddzwaniania (callback) lub wdzwaniania do serwera opartego na Linuksie nie jest niczym trudnym. Przedstawiamy sposób rozwiązania tego problemu oraz szczegóły konfiguracji usługi typu dial-on-demand.

Przykładowe [skrypty](#) konfigurujące PPP w Windows 9x

[Przykładowy plik dynamic.filter](#)

[Podstawowe parametry pracy diald](#)

Dostawcy usług internetowych często wykorzystują serwer z Linuksem, do którego podłączonych jest wiele modemów umożliwiającym użytkownikom dostęp do Internetu typu dial-up. Skonfigurowanie takiego serwera nie jest trudne. Podstawą systemu jest jądro z protokołem PPP wkompielowanym weń na stałe bądź w postaci modułów. W przypadku kompilacji PPP do modułów musimy umieścić następujące wpisy do pliku /etc/conf.modules:



Określanie portu modemu oraz jego parametrów

dla jądra w wersji 2.3.x i późniejszych:

```
alias ppp ppp_generic
alias tty-ldisc-3 slhc
alias char-major-108 ppp_async
```

dla jąder 2.2.x:

```
alias ppp0 ppp_deflate
alias ppp-compress-1 off
alias ppp-compress-2 off
alias ppp-compress-21 bsd_comp
alias ppp-compress-24 ppp_deflate
alias ppp-compress-26 ppp_deflate
```

Wkompilowanie PPP na stałe jest prostsze i nie wymaga wprowadzania powyższych zapisów do conf.modules, za to znacznie zwiększa rozmiar nowo powstałego jądra. Opisywane rozwiązanie będzie umożliwiać dostęp dial-up i callback wszystkim systemom zawierającym emulator terminala oraz stos TCP/IP. Dzięki temu użytkownicy będą mogli korzystać zarówno z Windows 95/98, jak i Windows 3.x (z Trumpet Winsock), a nawet z DOS-u za pośrednictwem np. Telixa.

Programy mgetty i callback

Do zrealizowania naszego zadania niezbędny jest pakiet mgetty, który zazwyczaj znajduje się w każdej dystrybucji. Pozwala on wykorzystać wszelkie dostępne funkcje zawarte we współczesnych modemach, takie jak przesyłanie danych, odbiór i wysyłanie faksów czy usługi głosowe (np. automatyczna sekretarka, skrzynka głosowa). Z naszego punktu widzenia interesujące są tylko funkcje służące do przesyłania danych.

Aby system mógł porozumiewać się z naszym modemem, musimy umieścić w pliku /etc/inittab następującą linijkę:

```
sl:2345:respawn:/sbin/mgetty
-s 115200 -D ttyS3
```

gdzie:

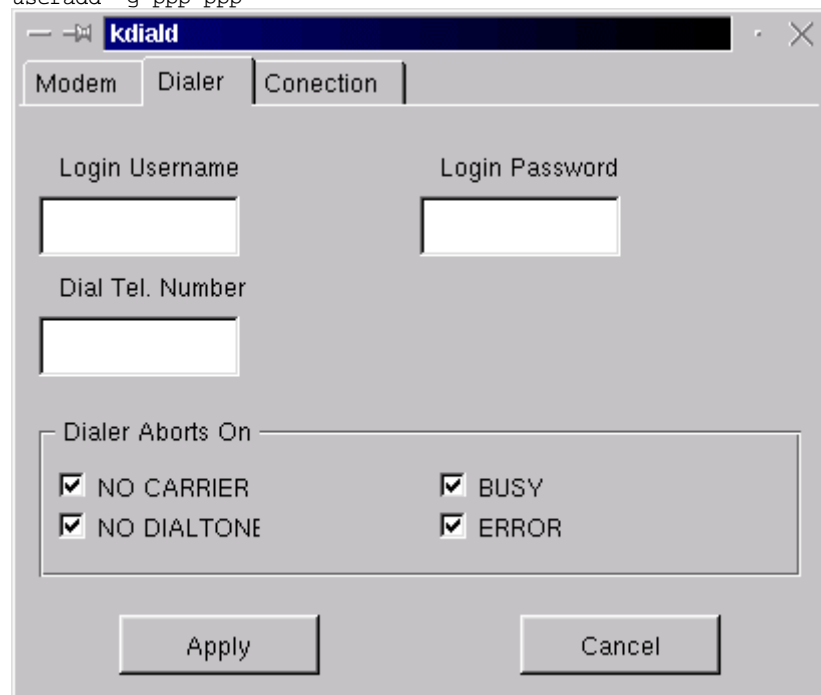
- **s** - prędkość komunikacji modem-port szeregowy,
- **D** - znak, że transmitujemy tylko dane,
- **ttys3** - port szeregowy z modemem odpowiadający portowi COM4 w systemach DOS/Windows.

Jeśli mamy więcej modemów podpiętych do serwera, wpisujemy tyle wierszy, ile jest modemów, pamiętając o zmianie w każdej linii numeru portu szeregowego.

Po zapisaniu zmian podłączamy modemy, a następnie za pomocą opcji init q zmuszamy proces init do ponownego przeczytania swoich ustawień. Po skonfigurowaniu modemu tak, by oczekiwał na połączenie na przypisanym mu porcie, należy jeszcze dodać użytkownika, najlepiej o nazwie ppp. Służą do tego polecenia:

```
groupadd ppp
```

```
useradd -g ppp ppp
```



Ustalanie opcji połączenia (użytkownik, hasło, numer telefonu)

Następnie zmieniamy hasło użytkownika ppp, musimy też zadbać o to, by nie miał on dostępu do powłoki bash i

korzystał z osobnego shella, który nazwiemy np. ppplogin. Shell ten będzie zwykłym plikiem tekstowym, który powinien zawierać instrukcję:

```
#!/bin/bash
```

```
exec /usr/sbin/pppd -detach
```

Plikowi ppplogin koniecznie trzeba nadać atrybut umożliwiający jego wykonywanie (chmod 755 ppplogin) i zmienić jego właściciela na root i grupę root (chown root. root ppplogin). Należy też pamiętać o ustawieniu atrybutu SUID dla demona pppd (chmod 4755 /usr/sbin/pppd). Aby domyślnym shellem użytkownika ppp był ppplogin, wydajemy jeszcze polecenie chsh ppp. Modyfikujemy też parametry demona pppd w pliku /etc/ppp/options, tak by zawierał wpisy:

proxyarp

crtststs

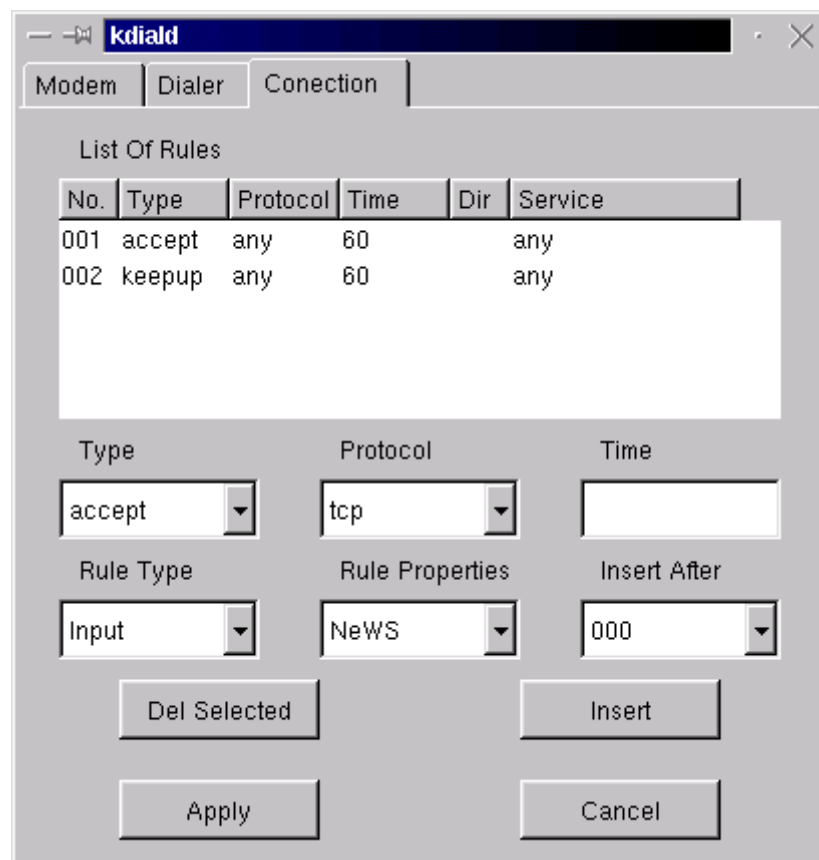
modem

lock

debug

gdzie:

- **proxyarp** - umożliwia wyjście poza sieć lokalną,
- **crtststs** - uaktywnia sprzętową kontrolę przepływu danych,
- **modem** - pozwala na używanie modemowych sygnałów sterujących,
- **lock** - tworzy specjalny plik blokujący, który uniemożliwia innym programom dostęp do modemu,
- **kdebug 3** - tworzy szczegółowe komunikaty w logach systemowych.



Wybór pakietów i protokołu zestawiającego łącze

Teraz dla każdego portu szeregowego, do którego podpięty mamy modem, należy stworzyć plik

/etc/ppp/options.ttySx (gdzie x odpowiada numerowi danego portu) o zawartości:

```
serwerIP:modemIP
```

SerwerIP jest tu adresem IP serwera, a modemIP - adresem IP, który będzie przyznany użytkownikowi wdzwanającemu się przez modem. Adresy te mogą należeć do puli prywatnych lub publicznych adresów IP.

Kolejną czynnością jest konfiguracja samego mgetty. Pliki z jego ustawieniami znajdują się w katalogu /etc/mgetty+sendfax. Plik mgetty.config należy zmodyfikować w ten sposób, by zawierał informacje dotyczące wykorzystanego portu:

```
port ttyS3
data-only y
speed 115200
```

Następnie - w pliku callback.config - muszą pojawić się wpisy:

```
dial-prefix ATDT
max-time 120
retry-time 60
```

gdzie:

- **dial-prefix ATDT** - ustala wybieranie na tonowe (domyślnie impulsowe),
- **max-time 120** - maksymalny czas, przez jaki serwer będzie próbował oddzwonić, jeśli linia będzie zajęta,
- **retry-time 60** - czas w sekundach pomiędzy kolejnymi próbami oddzwaniania.

Najciekawszym plikiem jest login.config. W nim należy określić pseudoużytkowników, tj. użytkowników definiowanych tylko na potrzeby programu mgetty i nie istniejących w systemie oraz ustalić ich powiązania z numerami telefonów i poleceniami wykonywanymi przy logowaniu. Przykładowy plik może wyglądać tak:

```
/AutoPPP/ - - /usr/sbin pppd file /etc/ppp/ppp.serv
praca - - /usr/sbin/ callback -S 3214323
teren - - /usr/sbin/ callback -S
* - - /bin/login @
```

Pierwszy wiersz automatycznie uruchamia demona pppd w przypadku, gdy ktoś zadzwoni. Drugi powoduje, że gdy gość zaloguje się do serwera przez modem jako użytkownik "praca", program callback rozłączy go i po chwili oddzwoni pod zapisany numer. Kolejny wiersz pliku oznacza, że po zalogowaniu użytkownika "teren" program callback najpierw zapyta, pod jaki numer oddzwonić, następnie rozłączy się i wybierze wprowadzony numer. Ostatni wiersz z pliku login.config umożliwia zalogowanie dowolnego użytkownika istniejącego w systemie i pracę w trybie interaktywnym (jak przez telnet) lub - po zalogowaniu się jako użytkownik ppp z hasłem ppp - na podłączenie do sieci bez oddzwaniania (jak w TP SA).

Opcja -S zmusza program callback do użycia tego portu szeregowego, z którego został wywołany. W przeciwnym razie nie będzie on wiedział, jak skomunikować się z mgetty w celu oddzwonienia i nie zadziała.

alpha.greenie.net/mgetty -> Mgetty - zamiennik systemowego getty wzbogacający jego funkcję o możliwości logowania przez modemy oraz usługę callback; rozszerzony o pakiet mgetty+sendfax pozwala na kontrolę nad faksmodemami;
diald.sourceforge.net -> Diald - inteligentne narzędzie zestawiające połączenie PPP/SLIP na żądanie;
homepages.go.com/~linux_rules -> Diald applet - prosty applet na panel w GNOME do kontrolowania procesu diald;
diald-top.sourceforge.net -> Diald-top - program w przejrzysty sposób wyświetlający listę wszystkich pakietów przechodzących przez łącze zestawione przez diald;
www.dinkytoy.nis.za -> Kdiald - nakładka na KDE ułatwiająca i przyspieszająca skonfigurowanie diald
www.flyn.org/#AEN83 -> Wife's Network Controller - applet dla GNOME kontrolujący diald
www.csclub.uwaterloo.ca/... -> wmdctrl - aplikacja pracująca pod WindowMakerem, wyświetlająca statystyki z diald i pozwalająca na sterowanie jego działaniem

Parametry dla pppd znajdują się w pliku ppp.serv:

```
auth
-chap
```

```
+pap
login
asynmap 0
dns-addr 1.1.1.1
dns-addr 1.1.1.2
```

Powyższe opcje włączają dla użytkownika dzwoniącego autoryzację PAP oraz przyznają adresy serwerów DNS. Plik z hasłami do autoryzacji poprzez PAP znajduje się w /etc/ppp/pap-secrets.

diald

Program diald umożliwia zestawienie połączenia na żądanie, tzw. dial-on-demand. Przydaje się zwłaszcza w małych sieciach, które nie mają stałego dostępu do Internetu. Program pozwala na znaczne oszczędności, udostępniając pozornie stałe podłączenie do Internetu. Zasada działania tego rozwiązania jest następująca: gdy dowolny komputer z sieci lokalnej zażąda dostępu do Internetu (np. odwoła się do jakiegoś serwera WWW), wówczas diald automatycznie nawiązuje połączenie, a po wykryciu braku aktywności na łączu przez pewien czas, zamyka połączenie.

[Pingwin na skrzyżowaniu](#) (routing), 22/2000, str. 74, ID=709
[Elastyczne jądro Pingwina](#) (Linux 2.4), 24/2000, str. 75, ID=838
[Konfiguracja SDI w systemach Windows, Linux i NetWare](#), 13/2000, str. 96, ID=469
[PPP na Linuksie](#), 23/1999, str. 90, ID=965
[Dynamiczny czy statyczny](#) (protokół DHCP, konfiguracja serwera pod Linuksem), 22/2000, str. 80, ID=707
[Zdobycie shell](#) (bezpieczeństwo), 22/1999, str. 79, ID=933
[Nie tylko zdrowy rozsadek](#) (bezpieczeństwo), 4/2000, str. 88, ID=186
[X Window - wygoda ponad bezpieczeństwo](#), 24/1999, str. 101, ID=992

Efektywne skonfigurowanie dialda wymaga najpierw określenia reguł jego działania, np. aby nie nawiązywał połączenia po wykryciu sygnału sprawdzającego aktywność komputera o danym adresie (ping). W tym celu trzeba ustalić, jakie pakiety mogą powodować zestawienie połączenia PPP (np. WWW, FTP, IRC). Wszelkie parametry zawarte są w pliku dynamic.filter, na podstawie którego diald decyduje, jaką akcję podjąć. Dobrą pomocą jest tutaj komenda man diald-examples. Przykładowy plik dynamic.filter znajduje się na stronie 82. Program ten jest wstępnie skonfigurowany, np. nie powoduje zestawiania łącza na żądanie serwera DNS, ściągającego co pół godziny aktualizację stref z domenami.

Diald może być także bardzo użytecznym narzędziem w przypadku sieci podłączonych do Internetu przez SDI. Praktyka pokazuje, że usługa ta czasami potrafi z niewiadomych przyczyn zrywać połączenie z serwerem dostępowym i konieczne jest ponowne zestawienie połączenia. W takim przypadku diald zadba o to, aby przy braku łączności wznowić ją przy pierwszej próbie przejścia zasobów z Internetu. Diald udostępnia też funkcję kontroli godzin, w których łączenie się z Siecią jest dozwolone. Podstawowe parametry pracy diald definiujemy w pliku diald.conf przedstawionym powyżej.

Istnieje wiele przyjaznych programów, które mogą ułatwić konfigurację i administrację narzędziem diald. W przeważającej większości są to graficzne nakładki, które potrafią także monitorować używane łącze i pokazywać obszerne statystyki odnośnie jego używania.

Skrypty dla Windows

Konfiguracja klienta dial-up pod Windows 9x:

1. tworzymy nowe połączenie, używając Dial-up Networking;
2. w parametrach połączenia (Nowe połączenie -> Właściwości -> Konfiguruj -> Połączenie -> Zaawansowane -> Dodatkowe ustawienia) wpisujemy linię AT&C0S0=1, co spowoduje właściwe działanie callbacku oraz zmusi modem do automatycznego odebrania nadchodzącego połączenia po pierwszym dzwonku;
3. przy korzystaniu z Dial-up Networking z Windows 9x, w celu ręcznego wpisania numeru telefonu należy uaktywnić opcję "Wywołaj okno terminala po wywołaniu numeru" albo użyć skryptu automatyzującego operację:

```
proc main
waitfor "ogin:"
transmit "praca^M"
```

```
waitfor "ogin:"
transmit "ppp^M"
waitfor "ssword:"
transmit "ppp^M"
delay 1
endproc
```

Skrypt ten nazywamy np. callback.scp i umieszczamy w katalogu \Program Files\Accessories.

Jeśli nie chcemy, żeby serwer do nas oddzwaniał, logujemy się jako użytkownik ppp, wykorzystując następujący skrypt:

```
proc main
waitfor "ogin:"
transmit "ppp^M"
waitfor "ssword:"
transmit "ppp^M"
delay 1
endproc
```

Spowoduje to automatyczne zestawienie połączenia PPP między naszym komputerem a serwerem dial-up bez oddzwaniań.

Skrypt pobierający nazwę użytkownika oraz jego hasło z okienka Dial-up w Windows 9x może wyglądać tak:

```
proc main
waitfor "ogin:"
transmit $USERID
transmit "^M"
waitfor "ssword:"
transmit $PASSWORD
transmit "^M"
delay 1
endproc
```

Do określenia numeru, pod który serwer ma oddzwonić wykorzystujemy skrypt:

```
proc main
waitfor "ogin:"
transmit "teren^M"
waitfor "number for
callback:"
transmit "123456^M"
waitfor RING
transmit "ATA^M"
waitfor "ogin:"
transmit "ppp^M"
waitfor "ssword:"
transmit "ppp^M"
delay 1
endproc
```

```
#-----
# Regu•ki dla pakietów TCP
#-----

# Po nawi•zaniu po••czenia, je•li druga strona nie odpowie
# przez 15 sekund, ••cze jest automatycznie zamykane
keepup tcp 15 tcp.syn

# Nie podno• ••cza przy transferach domen mi•dzy serwerami DNS
ignore tcp tcp.dest=tcp.domain
ignore tcp tcp.source=tcp.domain

# Ruch pakietów Netbios (Sie• Windows widoczna w Otoczeniu
# Sieciowym) tak•e nie zestawia po••czenia PPP
ignore tcp tcp.source=tcp.netbios-ns,tcp.dest=tcp.netbios-ns

# Tylko pakiety TCP z flagami SYN powoduj• nawi•zanie po••czenia PPP
ignore tcp ip.tot_len=40,tcp.live

# Ruch pakietów dotycz•cych us•ug WWW utrzymuj• ••cze przez
# 2 minuty, pó•niej nast•puje roz••czenie
keepup tcp 120 tcp.dest=tcp.www
keepup tcp 120 tcp.source=tcp.www

# Po••czenia szyfrowane przez SSL (Secure Socket Layer) tak•e
# utrzymuj• ••cze przez 2 minuty
keepup tcp 120 tcp.dest=tcp.ssl
keepup tcp 120 tcp.source=tcp.ssl

# Gdy nie ma •adnych aktywnych po••cze• TCP, ••cze jest zamykane
```

```

# natychmiastowo
keepup tcp 5 !tcp.live
ignore tcp !tcp.live

# Ruch FTP utrzymuje ..cze przez 2 minuty
keepup tcp 120 tcp.dest=tcp.ftp
keepup tcp 120 tcp.source=tcp.ftp
keepup tcp 120 tcp.dest=tcp.ftp-data
keepup tcp 120 tcp.source=tcp.ftp-data

# Je.li powy.sze regu.y nie s. spe.nione, ..cze istnieje jeszcze
# przez 10 minut
keepup tcp 600 any

#-----
# Regu.ki dla pakietów UDP
#-----

# Ignoruj pakiety us.ugi 'who' (us.uga uniksowa)
ignore udp udp.dest=udp.who
ignore udp udp.source=udp.who

# Nie podno. ..cza, dla pakietów demona dynamicznego
# routowania (routed)
ignore udp udp.dest=udp.route
ignore udp udp.source=udp.route

# Nie podno. ..cza dla pakietów us.ugi synchronizacji czasu
# poprzez sie.
ignore udp udp.dest=udp.ntp
ignore udp udp.source=udp.ntp
ignore udp udp.dest=udp.timed
ignore udp udp.source=udp.timed

# Nie podno. ..cza dla zapyta. DNS mi.dzy 2 serwerami DNS
ignore udp udp.dest=udp.domain,udp.source=udp.domain

# Zestaw po..czenie PPP dla zapytania DNS od komputera
# z naszej sieci innego ni. serwer domen
accept udp 30 udp.dest=udp.domain
accept udp 30 udp.source=udp.domain

# Zignoruj ruch Netbios
ignore udp udp.source=udp.netbios-ns,udp.dest=udp.netbios-ns

# Dynamiczne protoko.y routowania nie podnosz. ..cza
ignore udp tcp.dest=udp.route
ignore udp tcp.source=udp.route

# Inne pakiety UDP utrzymuj. ..cze przez 2 minuty
accept udp 120 any

# Pakiety ró.ne od wymienionych powy.ej utrzymuj. nieu.ywane
# po..czenie PPP przez 30 sekund
accept any 30 any

```

```

#####
# /etc/diald/diald.options

# Port, do którego mamy pod..czony modem
device /dev/ttyS0

# Plik logowania
accounting-log /var/log/diald.log

# Kolejka monitoruj.ca ..cze
#fifo /var/run/diald/diald.fifo

# Aktywacja debuggowania (zmniejsza wydajno..!)
#debug 31

# U.ywamy po..cze. PPP
mode ppp

# Lokalne IP (kiedy po..czenie jest zestawione, adres
# zamieniany jest na ten przyznany przez naszego providera)
local 127.0.0.1

# Zdalne IP (kiedy po..czenie jest zestawione, poni.szy
# adres zostaje nadpisany przez adres zdalnego serwera)
remote 127.0.0.4

```

```

# Maska podsieci naszego ••cza WAN (równie dobrze mo•e by•
# 255.255.255.252, je•li ma to by• ••cze point-to-point)
netmask 255.255.255.0

# Dynamiczne IP uzyskiwane od naszego providera
dynamic

# Je•li po••czenie zostanie zamkni•te przez drug• stron• ••cza PPP,
# zestaw je ponownie, gdy w kolejce czekaj• niewys•ane pakiety
two-way

# ••cze PPP b•dzie tras• domy•ln•
defaultroute
# •cie•ki do skryptów z regu•kami odno•nie routowania
#addroute "/etc/diald/addroute"
#delroute "/etc/diald/delroute"

# Skrypt zawieraj•cy polecenia, które maj• by• wykonane,
# gdy jeste•my pod••czeni do Internetu (np. opró•nij kolej• wiadomo•ci
# e-mail czekaj•cych na wys•anie)
ip-up /etc/diald/ip-up
ip-down /etc/diald/ip-down

# Skrypt u•ywany do zestawiania i zamykania po••czenia PPP
# (zawiera zazwyczaj sekwencje polece• dla modemu oraz
# numer telefonu)
connect "/etc/diald/diald.connect"
#disconnect "/etc/diald/diald.disconnect"

# Powoduje utworzenie specjalnego pliku blokuj•cego
# dost•p do modemu dla innych programów
#lock

# Opcje dla pppd
modem
crtscts
speed 115200

# Opcje pozwalaj•ce na ustawienie czasu braku aktywno•ci
# na ••czu, po up•ywie którego ••cze jest zamykane, oraz
# odst•pów czasu mi•dzy kolejnymi próbami podejmowanymi przez
# diald w celu zestawiania po••czenia w przypadku zaj•tej linii
connect-timeout 120
redial-timeout 60
start-pppd-timeout 120
died-retry-count 0
redial-backoff-start 4
redial-backoff-limit 300
dial-fail-limit 10

# Blokada dost•pu do Sieci w podanych godzinach
# Przyk•ad: dost•p o ka•dej porze w soboty i niedziele,
# a od poniedzia•ku do pi•tku tylko w godzinach 8-18
restrict 18:00:00 8:00:00 1-5 * *
down
restrict * * * * *

# Filtry definiuj•ce pakiety, które mog• zestawia• po••czenie,
# i pakiety, które s• ignorowane przez diald
#include /etc/diald/standard.filter
include /etc/diald/dynamic.filter

#####

```