

## Procedimentos para flash do DIR-300 com DD-WRT e OpenWRT :-)

por: Rodrigo Troian e Vinicius John - versão 2

Estudando e testando redes Mesh nos deparamos com a necessidade de trocar o firmware original dos roteadores para firmwares livres. Neste pequeno artigo demonstraremos a troca do firmware para o roteador D-Link modelo DIR-300, um equipamento barato e de fácil acesso no mercado brasileiro, que após a troca do sistema operacional deve agregar alguns reais a mais, pois o mesmo passa a permitir seu uso com WDS, múltiplas interfaces wifi virtuais, estatísticas, acesso ssh, QOs, rodar protocolos de rede mesh (olsr, b.a.t.m.a.n., etc.) enfim, coisas que só o GNU/Linux faz para você!

Tenha clareza de que você estará modificando o software que controla todo o hardware do seu equipamento, o sistema operacional. Qualquer problema que ocorra principalmente durante os processos críticos de gravação poderá danificar o hardware permanentemente, faça por sua conta e risco! Mas os resultados compensam, e como! :-)

A parte inicial da troca do firmware deste router é igual tanto para DD-WRT como OpenWRT. Este script foi escrito e testado usando sistema operacional GNU/Linux e OS X como estação.

**Dicas:** Tenha paciência e vá com calma. Verificando duas vezes antes de dar os comandos. No caso de uploads de arquivo, verifique se os arquivos existem e o servidor tftp está escutando. Aguarde os comandos retornarem ao prompt, muitas vezes demoram um pouco na execução. Se houver possibilidade, conecte seu roteador e micro a nobreak.

**Material necessário:** microcomputador com placa de rede disponível, servidor tftp (no GNU/Linux usamos atftpd), putty no GNU/Linux e telnet no OS X, cabo de rede, roteador D-Link Dir-300 e o seguinte script para abrir o gerenciador de boot do roteador, conforme segue:

```
gnu:/# cat scriptdd.bash
#!/bin/bash
echo ""
echo "Usando IP padrao gerenciador boot original do DIR-300: 192.168.20.81"
host="192.168.20.81"

#Para acessar o gerenciador de boot do dd-wrt comente as duas linhas acima e
#descomente as duas abaixo
#echo "Usando IP padrao do dd-wrt: 192.168.0.1"
#host="192.168.1.1"

while true
do
if eval "ping -c 1 -s 1 $host" > /dev/null; then
echo "Acordou o roteador. Tentando conectar via telnet"

#no machintosh, somente o comando telnet funcionou
#telnet $host 9000

#para micros com sistema gnu/linux, precisamos instalar o putty e criar um arquivo
#ctrlc.txt com o conteudo "^C" (sem aspas) e colocar no mesmo diretorio deste script
putty telnet://$host:9000 -m ctrlc.txt
break

else
echo "Esperando pelo gerenciador de boot. Pressione CTRL + C para sair."
sleep 1
fi
done
```

### Começando então:

Para saber os detalhes técnicos do seu roteador, (coisa que estranhamente os fabricantes não fazem), um caminho é acessar o site do DD-WRT e encontrar o seu roteador (marcar/modelo), lá estão quantidade de memória ram, memória flash e processador do equipamento.

#### 0. Baixar os arquivos necessários:

No site do DD-WRT baixe os arquivos ap61.ram e ap61.rom.

No site do OpenWRT.org, encontre a versão que você quer (kamikaze/backfire - última), acesse a pasta correspondente ao sistema, no caso do DIR-300 atheros, e baixe openwrt-atheros-vmlinux.lzma e openwrt-atheros-root.squashfs. Cuidado com estes arquivos, é importante um correto download. Já copie os arquivos na raiz do servidor tftp. Nós utilizamos o OpenWRT na versão kamikaze 8.09.2.

1. Configurar IP local manualmente para 192.168.20.80/24 (255.255.255.0). Se estiver usando o network-manager no GNU/Linux faça através dele ou desative-o e faça manualmente.
2. Conectar o cabo de rede no micro e na porta WAN do roteador.
3. Desligar a energia elétrica do router e religue, pressionando já o botão reset localizado na parte de trás do router. Assim que ligar, inicie o script no micro (conforme comando abaixo) e mantenha o reset pressionado até a mensagem "Router Awake" aparecer já como saída do script.

```
gnu:/# bash scriptdd.bash
```

4. Aguarde o prompt do RedBoot aparecer (gerenciador de boot padrão do DIR-300)

```
Usando IP padrao DIR-300: 192.168.20.81
Waiting for Redboot to boot. Press CTRL + C to quit
Router Awake
Trying 192.168.20.81...
Connected to 192.168.20.81.
Escape character is '^]'.
RedBoot>
```

5. Levante o tftp server (no Debian/Ubuntu atftpd), com os arquivos necessários já copiados para sua raiz (ap61.ram e ap61.rom para /tftpboot). Tenha certeza que o tftp está rodando ("ps aux | grep ftp" ou "netstat -anp | grep ftp"). Caso utilize firewall, desative-o ou libere a porta padrão do tftp (69).

6. Digite no prompt do RedBoot. Estaremos carregando o sistema de boot do dd-wrt para a memória ram. Se houver interrupção de energia, será necessário gravar novamente a ram.

```
RedBoot> load ap61.ram
Using default protocol (TFTP)
Entry point: 0x800410bc, address range: 0x80041000-0x800680d8
```

Caso der timeout na saída do comando load, o router não está conseguindo conectar ao servidor tftp, reveja suas configurações. Também é útil utilizar um sniffer de pacotes para ver o que está acontecendo.

Reiniciando o router  
RedBoot> go

7. Troque o IP da placa de rede do micro para 192.168.1.2/24. O equipamento agora já irá carregar o gerenciador de boot do dd-wrt gravado na ram.

```
gnu:/usr/tftp# telnet 192.168.1.1 9000
Trying 192.168.1.1...
Connected to 192.168.1.1.
Escape character is '^]'.
DD-WRT>
```

Observe que o prompt já é diferente. Lembre-se de reiniciar o servidor tftp depois da mudança de ip da placa de rede.

### Continuando então.

Apagando a flash de inicialização, incluindo a rom de inicialização original, portanto a partir de agora, não pode haver interrupção de energia.

```
DD-WRT> fis init
About to initialize [format] FLASH image system - continue (y/n)? y
*** Initialize FLASH Image System
... Erase from 0xbffe0000-0xbfff0000: .
... Program from 0x80ff0000-0x81000000 at 0xbffe0000: .
DD-WRT>
```

Confirmando endereço ip do micro.

```
DD-WRT> ip_address -h 192.168.1.2
IP: 192.168.1.1/255.255.255.0, Gateway: 0.0.0.0
Default server: 192.168.1.2
```

Carregando a rom de inicialização.

```
DD-WRT> load -r -b %{FREEMEMLO} ap61.rom
Using default protocol (TFTP)
Raw file loaded 0x80080000-0x800a8717, assumed entry at 0x80080000
```

Montando a imagem da rom.

```
DD-WRT> fis create -l 0x30000 -e 0xbfc00000 RedBoot
An image named 'RedBoot' exists - continue (y/n)? y
... Erase from 0xbfc00000-0xbfc30000: ...
... Program from 0x80080000-0x800a8718 at 0xbfc00000: ...
... Erase from 0xbffe0000-0xbfff0000: .
... Program from 0x80ff0000-0x81000000 at 0xbffe0000: .
```

Reiniciando o router.

```
DD-WRT> reset
```

Conectando novamente

```
bash-3.2# telnet 192.168.1.1 9000
Trying 192.168.1.1...
Connected to 192.168.1.1.
Escape character is '^]'.
DD-WRT>
```

Confirmando mais uma vez o ip do micro

```
DD-WRT> ip_address -h 192.168.1.2
IP: 192.168.1.1/255.255.255.0, Gateway: 0.0.0.0
Default server: 192.168.1.2
```

Gravando a imagem da rom de boot do dd-wrt.

```
DD-WRT> fis init
About to initialize [format] FLASH image system - continue (y/n)? y
*** Initialize FLASH Image System
... Erase from 0xbffe0000-0xbfff0000: .
... Program from 0x80ff0000-0x81000000 at 0xbffe0000: .
```

Trocado o boot loader do roteador, não há problemas com interrupção de energia. Começamos os procedimentos para instalar o firmware OpenWRT do DD-WRT são diferentes e seguem separados.

```
gnu$ telnet 192.168.1.1 9000
Trying 192.168.1.1...
Connected to 192.168.1.1.
Escape character is '^]'.

```

## Para o OpenWRT

Carregue no servidor tftp os seguintes arquivos (baixado já do site oficial do openwrt, se houver possibilidade, verifique o MD5):

```
openwrt-atheros-vmlinux.lzma
openwrt-atheros-root.squashfs
```

Carregando o lzma (kernel básico) para a ram. A partir de agora se faltar energia este processo deve ser reiniciado.

```
DD-WRT> load -r -b %{FREEMEMLO} openwrt-atheros-
vmlinux.lzma
Using default protocol (TFTP)
Raw file loaded 0x80040800-0x801007ff, assumed entry
at 0x80040800
```

Inicializando a imagem.

```
DD-WRT> fis create -e 0x80041000 -r 0x80041000
vmlinux.bin.l7
... Erase from 0xbfc30000-0xbfcf0000: .....
... Program from 0x80040800-0x80100800 at 0xbfc30000:
.....
... Erase from 0xbffe0000-0xbfff0000: .
... Program from 0x80ff0000-0x81000000 at 0xbffe0000: .
DD-WRT>
```

Carregando o sistema de arquivos para a ram.

```
DD-WRT> load -r -b %{FREEMEMLO} openwrt-atheros-
root.squashfs
Using default protocol (TFTP)
Raw file loaded 0x80040800-0x801e07ff, assumed entry
at 0x80040800
```

Gravando o sistema de arquivos com o kernel. Este

## Para o DD-WRT

Carregue no servidor tftp os seguintes arquivos (baixado já do site oficial do dd-wrt, se houver possibilidade, verifique o MD5):

```
linux.bin
```

Para o DD-WRT como comentamos antes há somente um arquivo para ser carregado — kernel e sistema de arquivos juntos.

```
DD-WRT> load -r -b 0x80041000 linux.bin
Using default protocol (TFTP)
Raw file loaded 0x80041000-0x803ddfff, assumed entry at
0x80041000
```

Gravando o sistema de arquivos com o kernel. Este processo pode demorar.

```
DD-WRT> fis create linux
Connection closed by foreign host.
```

```
GNU$ telnet 192.168.1.1 9000
Trying 192.168.1.1...
Connected to 192.168.1.1.
Escape character is '^]'.

```

Duas configurações básicas para o script de boot.

```
DD-WRT> fconfig boot_script true
boot_script: Setting to true
Update RedBoot non-volatile configuration - continue
(y/n)? y
... Erase from 0xbffe0000-0xbfff0000: .
```

processo pode demorar.

```
DD-WRT> fis create rootfs
```

```
... Erase from 0xbfcf0000-0xbfe90000: .....  
... Program from 0x80040800-0x801e0800 at 0xbfcf0000:  
.....  
... Erase from 0xbffe0000-0xbfff0000: .  
... Program from 0x80ff0000-0x81000000 at 0xbffe0000: .  
DD-WRT>
```

Configurações do script de boot do sistema, inclusive indicando o kernel que o mesmo irá carregar.

```
DD-WRT> fconfig -d
```

```
Run script at boot: false ? true
```

```
Boot script:
```

```
Enter script, terminate with empty line
```

```
>> fis load -l vmlinux.bin.l7
```

```
>> exec
```

```
>>
```

```
Boot script timeout (1000ms resolution): 0 ? 5
```

```
Use BOOTP for network configuration: true ? false
```

```
Gateway IP address: ? 192.168.1.2
```

```
Local IP address: ? 192.168.1.1
```

```
Local IP address mask: ? 255.255.255.0
```

```
Default server IP address: ? 192.168.1.2
```

```
Console baud rate: 9600 ? 9600
```

```
GDB connection port: 9000 ? 9000
```

```
Force console for special debug messages: false ? false
```

```
Network debug at boot time: false ? false
```

```
Update RedBoot non-volatile configuration - continue
```

```
(y/n)? y
```

```
... Erase from 0xbffe0000-0xbfff0000: .
```

```
... Program from 0x80ff0000-0x81000000 at 0xbffe0000: .
```

Resetando o router.

```
DD-WRT> reset
```

Connection closed by foreign host.

Pronto! Se tudo deu certo você já poderá acessar seu router pela nova administração gráfica do OpenWRT chamada Luci, toda feita em Lua!

### Agora é só alegria e estudo do seu novo e poderoso roteador!

Para acessar a interface gráfica, abra seu navegador e digite <http://192.168.1.1> no endereço. Se tudo estiver correto abrirá uma tela de login. Estaremos fazendo outro artigo sobre como configurar o OpenWRT com a interface Luci, bem como sobre a configuração para redes mesh. Mas por enquanto, atenha-se que a interface wifi do roteador recém reinstalado vem por padrão desligada e para acessá-lo via ssh [root@192.168.1.1](ssh://root@192.168.1.1) é necessário antes definir uma senha para root, que por padrão vem em branco também! Boa sorte!

```
... Program from 0x80ff0000-0x81000000 at 0xbffe0000: .
```

```
DD-WRT> fconfig boot_script_timeout 4
```

```
boot_script_timeout: Setting to 4
```

```
Update RedBoot non-volatile configuration - continue  
(y/n)? y
```

```
... Erase from 0xbffe0000-0xbfff0000: .
```

```
... Program from 0x80ff0000-0x81000000 at 0xbffe0000: .
```

Configurações do script de boot do sistema, inclusive indicando o kernel que o mesmo irá carregar.

```
DD-WRT> fconfig
```

```
Run script at boot: true
```

```
Boot script:
```

```
Enter script, terminate with empty line
```

```
>> fis load -l linux
```

```
>> exec
```

```
>>
```

```
Boot script timeout (1000ms resolution): 4
```

```
Use BOOTP for network configuration: true
```

```
Default server IP address:
```

```
Console baud rate: 9600
```

```
GDB connection port: 9000
```

```
Force console for special debug messages: false
```

```
Network debug at boot time: false
```

```
Update RedBoot non-volatile configuration - continue
```

```
(y/n)? y
```

```
... Erase from 0xbffe0000-0xbfff0000: .
```

```
... Program from 0x80ff0000-0x81000000 at 0xbffe0000: .
```

Resetando o router.

```
DD-WRT> reset
```

Connection closed by foreign host.

E se tudo deu certo, já temos o dir-300 com o DD-WRT rodando.