

**Rudson Ribeiro Alves**  
**Mestre em Física (Semi-Condutores)**

Usuário Unix de 1991-1998  
Usuário Slackware deste 1994  
Professor da UVV deste 1998



**Mini-curso de GNU/Linux**  
**Aula 05**

**Noções básicas sobre GNU/Linux**  
**Plataforma Debian**

# Cronograma 1/1

- Redes
  - Configuração
  - Servidores: ssh, ftp e http
  - Configurando Clientes: NIS e NFS

# Configurando Rede

O GNU/Linux pode ter várias interfaces de rede com diferentes IPs. Alguns das interfaces suportadas são:

- ◆ **Loopback:** lo
- ◆ **Ethernet:** eth0, eth1, ...
- ◆ **Wi-Fi:** wlan0, wlan1, wifi0, ...
- ◆ **Token Ring:** tr0, tr1, ...
- ◆ **PPP:** ppp0, ppp1, ...

Tradicionalmente endereços de IP são agrupados em classes:

	IP addresses	net mask	length
Class A	1.0.0.0 - 126.255.255.255	255.0.0.0	/8
Class B	128.0.0.0 - 191.255.255.255	255.255.0.0	/16
Class C	192.0.0.0 - 223.255.255.255	255.255.255.0	/24

Endereços fora destes intervalos são usado para propósitos especiais.

# Configurando Rede

Alguns comandos/ferramentas úteis para a configuração de uma rede são:

- ♦ ***modprobe***: carregar módulo da placa de rede
- ♦ ***dmesg***: apresentar mensagens de inicialização do sistema
- ♦ ***lspci***: listar dispositivos de rede
- ♦ ***google***: pesquisa...
- ♦ ***ipconfig***: configura uma interface de rede
- ♦ ***dhcpd* ou *dhclient***: cliente dhcp
- ♦ ***route***: manipula os roteamentos da rede
- ♦ ***/etc/resolv.conf***: arquivos com os IPs dos servidores de nomes (DNS)
- ♦ ***hostname***: define o nome da máquina e rede

Geralmente a rede é configurada com êxito no momento da instalação do sistema. Mas, eventualmente, você pode necessitar fazer uma intervenção manual na configuração, como a adição de mais dispositivos de rede, novo roteamento, ...

# Configurando Rede: Geral

**1º Passo:** Identificar os dispositivos de rede. Isto pode ser feito através do comando *lspci* (graficamente *kinfocenter -> PCI*)

```
# lspci | grep -i net
03:02.0 Network controller: Broadcom Corporation BCM4306
802.11b/g Wireless LAN Controller (rev 03)
03:06.0 Ethernet controller: Realtek Semiconductor Co., Ltd.
RTL-8139/8139C/8139C+ (rev 10)
```

**2º Passo:** Verificar o carregamento dos módulos dos dispositivos. Isto pode ser feito checando as mensagens de inicialização do sistema com o comando *dmesg* (graficamente pelo *kssystemlog -> messages*)

```
# dmesg | grep -i eth
8139too Fast Ethernet driver 0.9.28
eth0: RealTek RTL8139 at 0xe0820400, 00:0f:b0:6c:bb:6e, IRQ 10
eth0: Identified 8139 chip type 'RTL-8100B/8139D'
wlan0: ethernet device 00:90:4b:ac:98:9e using NDIS driver:
bcmwl5, version: 0x3644101, NDIS version: 0x501, vendor: '',
14E4:4320:103C:12F8.5.conf
eth1: link down
ADDRCONF(NETDEV_UP): eth1: link is not ready
```

# Configurando Rede: Geral

**3º Passo:** Carregar módulos de rede se necessário. Em alguns casos os módulos da placa de rede podem não ser carregados na inicialização, embora isto não seja muito comum.

Quando isto ocorre, o *dmesg* não apresentará nenhuma informação sobre a placa. A carga de um módulo de rede é feita pelo comando *modprobe*.

```
# modprobe 8139too  
carrega o módulo da RealTek RTL8139
```

Se for necessário remover algum módulo utilize o comando *rmmmod* (*remove module*)

```
# rmmmod 8139too  
remove o módulo da RealTek RTL8139 da memória.
```

**4º Passo:** Definir o nome da máquina e da rede. Este processo é feito pelo comando *hostname*. O Nome da rede é automaticamente escrita no arquivo */etc/HOSTNAME* ou */etc/hostname*, dependendo da distribuição.

```
# hostname arabel.forgotten  
nomeia a máquina como "arabel" e a rede como "forgotten"
```

# Configurando Rede: IP Fixo

**5º Passo:** Configurar um endereço de rede com o *ifconfig*. Configurar uma rede com o *ifconfig* é bem simples:

```
# ifconfig [disp]
```

apresenta a configuração de rede para o dispositivo [disp]. Se [disp] for omitido, apresenta as configurações de todos os dispositivos carregados.

```
# ifconfig [disp] down
```

abaixa (down) a rede do dispositivo [disp] (eth0, eth1, lo, ...)

```
# ifconfig [disp] [addr] netmask [mask] broadcast [addr] up
```

levanta a rede no dispositivo [disp] com o endereço [addr], máscara de rede [mask], broadcast [baddr].

```
# ifconfig wlan0
```

```
wlan0      Link encap:Ethernet  HWaddr 00:90:4B:AC:98:9E
```

```
           inet addr:192.168.1.105  Bcast:192.168.1.255  Mask:255.255.255.0
```

```
           inet6 addr: fe80::290:4bff:feac:989e/64  Scope:Link ...
```

```
# ifconfig eth0 192.168.1.10
```

# Configurando Rede: IP Fixo

**6º Passo:** Configurar o roteamento da rede. O roteamento é configurado pelo comando *route*:

```
# route [-n]
```

apresenta a tabela de roteamento da rede.

```
# route add -net 192.168.1.0 netmask 255.255.255.0 dev wlan0
```

adiciona o roteamento para a rede 192.168.1.0 através do dispositivo wlan0

```
# route del -net 192.168.1.0 netmask 255.255.255.0 dev wlan0
```

remove o roteamento para a rede 192.168.1.0 através do dispositivo wlan0

```
# route add default gw 192.168.1.1
```

define o gateway padrão como sendo 192.168.1.1

```
# route add -net 127.0.0.0 netmask 255.0.0.0 lo
```

adiciona o roteamento para o dispositivo loopback

```
# route -n
```

Kernel IP routing table

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
192.168.1.0	0.0.0.0	255.255.255.0	U	0	0	0	wlan0
127.0.0.0	0.0.0.0	255.0.0.0	U	0	0	0	lo
0.0.0.0	192.168.1.1	0.0.0.0	UG	0	0	0	wlan0

# Configurando Rede: IP Fixo

**7º Passo:** Configurar o DNS. Os IPs dos servidores de nomes ficam armazenados no arquivo `/etc/resolv.conf`. Os servidores de nomes devem ser adicionados em linhas iniciadas por “nameserver”, como segue abaixo:

```
# cat /etc/resolv.conf
# Generated by dhcpd for interface wlan0
search forgotten
nameserver 201.30.255.6
nameserver 200.255.125.211
nameserver 200.19.74.21
```

# Configurando Rede: DHCP

**5º Passo:** Configurar uma rede via dhcp é bem mais simples. Os passos de 1 a 4 são os mesmo anteriores. A configuração é feita chamando o comando *dhclient* como segue

```
# dhclient eth0
```

Configurar a placa eth0 via dhcp. O *dhclient* configura o roteamento e DNS da rede.

**OBS:** Em algumas distribuições, o comando utilizado é *dhcpcd*.

# Configurando Rede: via script

Com exceção da alteração feita pelo comando *hostname*, as demais configurações serão perdidas na primeira reinicialização do sistema. Para fixar estas configurações, elas devem ser declaradas no arquivo `/etc/network/interfaces`, como segue:

## IP fixo

```
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static
address 192.168.1.90
gateway 192.168.1.1
netmask 255.255.255.0
network 192.168.1.0
broadcast 192.168.1.255
```

## dhcp

```
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface -
# use DHCP to find our address
auto eth0
iface eth0 inet dhcp
```

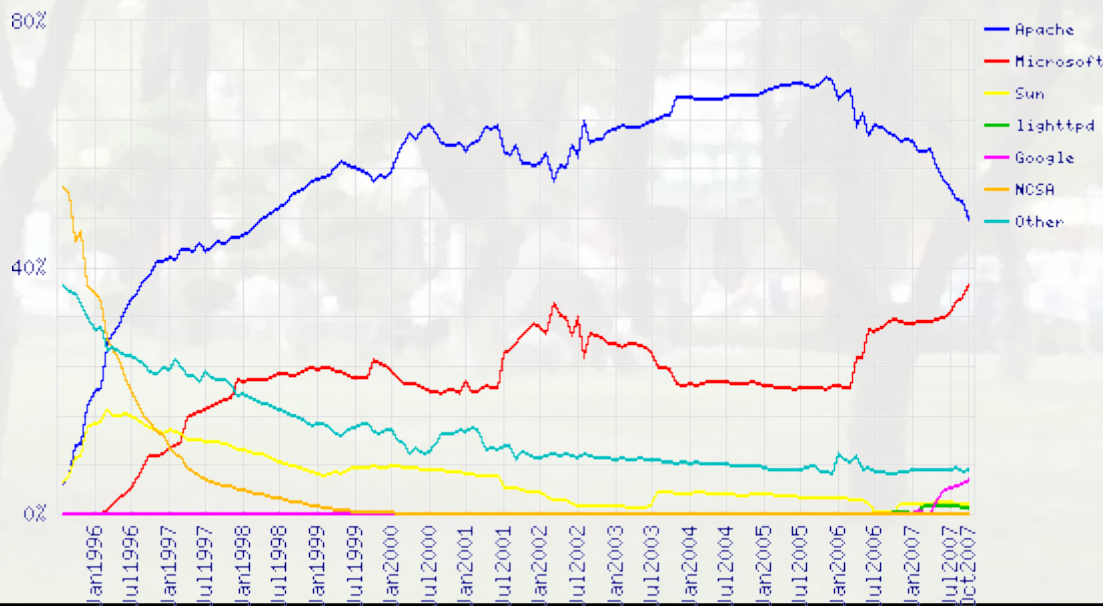
Após qualquer alteração a rede pode ser reiniciada com o script `/etc/init.d/networking`

```
# /etc/init.d/networking restart
```

# Serviço: http com o apache

O servidor Apache (ou Servidor HTTP Apache, em inglês: *Apache HTTP Server*, ou simples: Apache) é o mais bem sucedido servidor web livre. Foi criado em 1995 por Rob McCool, então funcionário do NCSA (*National Center for Supercomputing Applications*), Universidade Independente [1].

Numa pesquisa realizada em outubro de 2007, [Netcraft](#), foi constatado que a utilização do Apache supera 50% nos servidores ativos no mundo [2].



# Serviço: http com o apache

## Características:

- ◆ Possui suporte a scripts cgi usando linguagens como Perl, PHP, Shell Script, ASP, etc.
- ◆ Suporte a autorização de acesso podendo ser especificadas restrições de acesso separadamente para cada endereço/arquivo/diretório acessado no servidor.
- ◆ Autenticação requerendo um nome de usuário e senha válidos para acesso a alguma página/sub-diretório/arquivo (suportando criptografia via Crypto e MD5).
- ◆ Negociação de conteúdo, permitindo a exibição da página Web no idioma requisitado pelo Cliente Navegador.
- ◆ Suporte a tipos mime.
- ◆ Personalização de logs.
- ◆ Mensagens de erro.
- ◆ Suporte a virtual hosting (é possível servir 2 ou mais páginas com endereços/ portas diferentes através do mesmo processo ou usar mais de um processo para controlar mais de um endereço).
- ◆ Suporte a IP virtual hosting.
- ◆ Suporte a name virtual hosting.
- ◆ Suporte a servidor Proxy ftp e http, com limite de acesso, caching (todas flexivelmente configuráveis).
- ◆ Suporte a proxy e redirecionamentos baseados em URLs para endereços Internos.
- ◆ Suporte a criptografia via SSL, Certificados digitais
- ◆ Módulos DSO (Dynamic Shared Objects) permitem adicionar/remover funcionalidades e recursos sem necessidade de recompilação do programa.
- ◆ ...

# Serviço: http com o apache

Utilitários:

- ▶ **apache** - Servidor Web Principal
- ▶ **apachectl** - Shell script que faz interface com o apache de forma mais amigável
- ▶ **apacheconfig** - Script em Perl para configuração interativa básica do Apache
- ▶ **htpasswd** - Cria/Gerencia senhas criptografadas Crypto/MD5
- ▶ **htdigest** - Cria/Gerencia senhas criptografadas Crypto/MD5
- ▶ **dbmmanage** - Cria/Gerencia senhas em formato DBM (Perl)
- ▶ **logresolve** - Faz um DNS reverso dos arquivos de log do Apache para obter o endereço de hosts com base nos endereços IP's.
- ▶ **ab** - Apache Benchmarking - Ferramenta de medida de desempenho do servidor Web Apache.

# Serviço: http com o apache

Arquivos de configuração do Apache:

**/etc/apache/apache2.conf**

**/etc/apache/httpd.conf** - arquivo de configuração principal do Apache, possui diretivas que controlam a operação do daemon servidor. Um arquivo de configuração alternativo pode ser especificado através da opção "-f" da linha de comando.

**/etc/apache/access.conf** - Contém diretivas que controlam o acesso aos documentos. O nome desse arquivo pode ser substituído através da diretiva *AccessConfig* no arquivo principal de configuração.

...

# Serviço: http com o apache

## Arquivos de log:

- **access.log** - Registra detalhes sobre o acesso as páginas do servidor httpd.
- **error.log** - Registra detalhes saber erros de acesso as páginas ou erros internos do servidor.
- **agent.log** - Registra o nome do navegador do cliente (campo *UserAgent* do cabeçalho http).

## Instalação:

```
# aptitude install apache2
```

...

## Configuração:

Editar o arquivo `/etc/apache2/apache2.conf`

## Serviço: ssh com o openssh

O serviço de ssh permite fazer o acesso remoto ao console de sua máquina, em outras palavras, você poderá acessar sua máquina como se estivesse conectado localmente ao seu console (substituindo o rlogin e rsh).

A principal diferença com relação ao serviço telnet padrão, rlogin e rsh, é que toda a comunicação entre cliente/servidor é feita de forma encriptada usando chaves públicas/privadas RSA para criptografia garantindo uma transferência segura de dados[2].

# Serviço: ssh com o openssh

## Aplicativos:

- ▶ **ssh** - Cliente ssh (console remoto).
- ▶ **slogin** - Link simbólico para o programa ssh.
- ▶ **sshd** - Servidor de shell seguro ssh.
- ▶ **scp** - Programa para transferência de arquivos entre cliente/servidor
- ▶ **ssh-keygen** - Gera chaves de autenticação para o ssh
- ▶ **sftp** - Cliente ftp com suporte a comunicação segura.
- ▶ **sftp-server** - Servidor ftp com suporte a comunicação segura.
- ▶ **ssh-add** - Adiciona chaves de autenticação DSA ou RSA ao programa de autenticação.
- ▶ **ssh-agent** - Agente de autenticação, sua função é armazenar a chave privada para autenticação via chave pública (DSA ou RSA).
- ▶ **ssh-keyscan** - Scaneia por chaves públicas de autenticação de hosts especificados. O principal objetivo é ajudar na construção do arquivo local `known_hosts`.
- ▶ **ssh-copy-id** - Usado para instalação do arquivo `identity.pub` em uma máquina remota.

# Serviço: ssh com o openssh

## Arquivos de configuração:

- ▶ `/etc/ssh/sshd_config` - Arquivo de configuração do servidor ssh.
- ▶ `/etc/ssh/ssh_config` - Arquivo de configuração do cliente ssh.
- ▶ `~/.ssh/config` - Arquivo de configuração pessoal do cliente ssh.

## Instalação:

```
# aptitude install openssh-client openssh-server  
...
```

## Configuração:

Editar os arquivos: `/etc/ssh/sshd_config` e `/etc/ssh/ssh_config`, para servidor e cliente respectivamente.

## Iniciando e parando o servidor:

```
# /etc/init.d/ssh start  
# /etc/init.d/ssh stop
```

# Serviço: ftp com o vsftpd

*Very Secure FTP Daemon*, é um servidor FTP para sistemas UNIX-like licenciado sobre a *GNU General Public License*. Ele suporta IPv6 e SSL.

O vsftpd é o servidor padrão para muitas distribuições GNU/Linux, substituindo o proftpd.

## **Instalação:**

```
# aptitude install vsftpd
```

## **Iniciando/parando o serviço:**

```
# /etc/init.d/vsftpd start/stop
```

## **Arquivo de configuração:**

```
# /etc/vsftpd.conf
```

# Serviço: NFS – servidor

*Network file System*, é um servidor de arquivos que permite montar pastas de um servidor localmente.

## **Instalação:**

Para a instalação se um servidor nfs é necessário a instalação dos pacotes: `nfs-kernel-server nfs-common portmap`

```
# aptitude install nfs-kernel-server nfs-common portmap
```

Durante a instalação o daemon do portmap deve ser iniciado, juntamente com o nfs. Teus scripts de inicialização são, respectivamente:

```
/etc/init.d/portmat  
/etc/init.d/nfs-kernel-server
```

# Serviço: NFS – servidor – portmap

Portmap é um servidor que converte *RPC program numbers* em *DARPA protocol port numbers*. Ele deve estar rodando antes de se fazer uma chamada *RPC*.

Quando um servidor *RPC* é iniciado, ele diz ao *portmap* qual o número da porta ele está ouvindo, e qual *RPC program numbers* ele está preparado para servir.

Quando um cliente deseja fazer uma chamada *RPC* para um dado *program number*, ele primeiro contacta o *portmap* no servidor, para determinar o número da porta onde o pacote *RPC* será enviado. [4]

```
$ rpcinfo -p localhost
program vers proto  port
100000    2    tcp    111  portmapper
100000    2    udp    111  portmapper
100003    2    tcp    2049 nfs
100003    2    udp    2049 nfs
100003    3    tcp    2049 nfs
100003    3    udp    2049 nfs
100003    4    tcp    2049 nfs
100003    4    udp    2049 nfs
100005    1    tcp    44245 mountd
100005    1    udp    32771 mountd
100005    2    tcp    44245 mountd
100005    2    udp    32771 mountd
100005    3    tcp    44245 mountd
100005    3    udp    32771 mountd
100021    1    tcp    49652 nlockmgr
100021    1    udp    32770 nlockmgr
100021    3    tcp    49652 nlockmgr
100021    3    udp    32770 nlockmgr
100021    4    tcp    49652 nlockmgr
100021    4    udp    32770 nlockmgr
100024    1    tcp    51417 status
100024    1    udp    32768 status
```

# Serviço: NFS – servidor – /etc/exports

Este arquivo contém linhas com informações sobre as pastas exportadas pelo servidor. Uma entrada típica deste arquivo tem a forma:

```
diretório máquina1(opção1,opção2,...)
                               máquina2(opção1,opção2,...) ...
```

**diretório** é o diretório do servidor que será exportado

**máquinaX** são as máquinas clientes que terão acesso ao diretório exportado. Ex: 192.168.1.0/255.255.255.0 o mesmo que 192.168.1.0/24; 192.245.5.24; \*.lab-06

**opçãoX** determina o tipo de acesso que a máquina cliente terá.

Algumas opções são:

**ro** montagem com a permissão de apenas leitura

**rw** montagem com permissão de leitura e escrita

**sync** sincronizar os dados do servidor com os clientes  
(padrão)

**no\_root\_squash** esta opção dá ao root de uma máquina cliente os mesmos direitos do root do servidor.

# Serviço: NFS – servidor – /etc/exports

Exemplo do /etc/exports

```
# cat /etc/exports
/home          192.168.1.0/24(rw, sync)
/var/www      *.lab-06(ro, sync)
```

A primeira linha exporta o diretório /home para todas as máquinas da rede 192.168.1.x. A segunda linha exporta o diretório /var/www para todas as máquinas do domínio .lab-06.

Após alterar o /etc/exports é necessário reiniciar o nfs

```
# /etc/init.d/nfs-kernel-server restart
```

ou

```
# exports -xa
```

**OBS:** Para conhecer todas as opções do /etc/exports, veja o manual do sistema (man exports).

# Serviço: NFS – cliente

Para configurar um cliente NFS é necessário instalar os pacotes: nfs-common e portmap

```
# aptitude install nfs-common portmap
```

Montando um diretório NFS:

```
mount servidor:diretório_exportado diretório_destino
```

Exemplo:

```
# mount linus:/home /home
```

## Montando via fstab

Para uma montagem automática da pasta acima, basta adicionar a linha abaixo ao /etc/fstab

```
# device          mountpoint      fs-type  options      dump  fsckord
linus:/home      /home          nfs      defaults     0     0
```

# Serviço: NIS – cliente

*Network Information Service* é um serviço que provem informações para todas as máquinas de uma rede. Informações distribuídas pelo NIS geralmente são [5]:

- password/senha/home/...
- grupo

## **Instalação:**

**1º passo:** Instalando o daemon e configurando o NIS domainname

```
# aptitude install nis
```

Durante a instalação é perguntado o NIS domainname. Este nome deve ser o mesmo definido no servidor. O NIS domainname será armazenado no arquivo `/etc/defaultdomain`

**2º passo:** Definir o *NIS server*

O nome (ou IP) do servidor NIS deve ser publicado no arquivo `/etc/yp.conf`.

# Serviço: NIS – cliente

**3º passo:** Adicionar informações do servidor a máquina local  
Para que a máquina local reconheça os usuários declarados no servidor é necessário adicionar um “+” ao final dos arquivos de informação que serão compartilhados pela rede:

- ♦ /etc/passwd
- ♦ /etc/group
- ♦ /etc/shadow

Isto pode ser feito com um “echo”

```
# echo “+” >> /etc/passwd  
# echo “+” >> /etc/group  
# echo “+” >> /etc/shadow
```

**4º passo:** Reiniciar o NIS

```
# /etc/init.d/nis restart
```

# Bibliografia:

- [1] Wikipédia – pt, Servidor Apache,  
[http://pt.wikipedia.org/wiki/Servidor\\_Apache](http://pt.wikipedia.org/wiki/Servidor_Apache)
- [2] Guia Foca Linux – Avançado,  
<http://focalinux.cipsga.org.br/guia/avancado/index.htm>
- [3] Debian Help – <http://www.debianhelp.co.uk/nfs.htm>
- [4] Portmap manual (man portmap)
- [5] Debian NIS HOWTO -  
<http://lyre.mit.edu/~powell/debian-howto/nis.html>