

# Conteúdos

## 1. Serviços de rede (12 horas)

Nesta secção, vamos explorar como os serviços de rede são geridos no Linux, desde o seu início até ao encerramento, e os principais ficheiros e diretórios envolvidos neste processo.

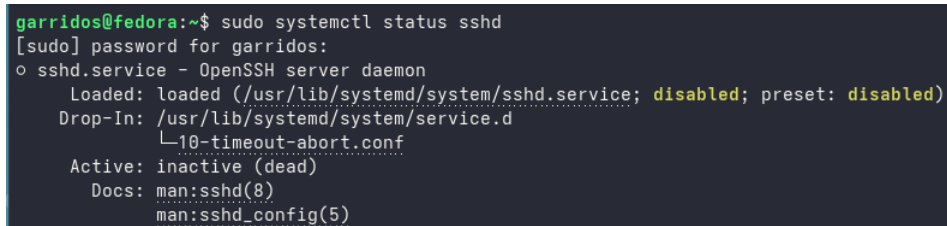
- **Conceito Chave: O Papel do `systemd`** (3 horas)

O `systemd` é o gestor de sistema e de serviços padrão na maioria das distribuições Linux modernas, substituindo sistemas mais antigos como o SysVinit. Ele usa "unidades" para gerir processos, o que lhe dá um controlo mais granular, robusto e eficiente sobre os serviços. Em vez de scripts de inicialização simples, as unidades `systemd` podem definir dependências, o que garante que os serviços iniciam na ordem correta.

- **Gestão de Serviços com `systemctl`** (5 horas)

O comando `systemctl` é a ferramenta central para interagir com o `systemd`. Vamos cobrir os subcomandos essenciais:

- `systemctl start [serviço]`: Inicia um serviço.
- `systemctl stop [serviço]`: Para um serviço em execução.
- `systemctl restart [serviço]`: Reinicia um serviço.
- `systemctl status [serviço]`: Mostra o estado detalhado do serviço.
- `systemctl enable [serviço]`: Habilita o serviço para iniciar no arranque do sistema.
- `systemctl disable [serviço]`: Impede que o serviço inicie no arranque.



```
garridos@fedora:~$ sudo systemctl status sshd
[sudo] password for garridos:
○ sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/ssh.service; disabled; preset: disabled)
   Drop-In: /usr/lib/systemd/system/service.d
            └─10-timeout-abort.conf
   Active: inactive (dead)
     Docs: man:sshd(8)
           man:sshd_config(5)
```

Figura 1: Exemplo da saída do comando `systemctl status sshd`. Fonte: Imagem encontrada via Google Images.

A resposta mostra o estado atual do serviço (ativo/inativo), o PID (Process ID), há quanto tempo está a correr e as últimas linhas dos seus logs. Isto é crucial para diagnosticar rapidamente se um serviço está a funcionar.

- **Dicas de Resolução de Problemas** (2 horas)

- **Verificação de Logs**: Se um serviço não inicia, o primeiro passo é verificar os seus logs. O comando `journalctl -u [serviço]` mostra todo o histórico de logs do serviço, o que pode revelar a causa do erro.
- **Estado de Ativação**: Use `systemctl is-enabled [serviço]` para verificar se um serviço está configurado para iniciar automaticamente no arranque do sistema.

- **Exercício de Consolidação** (2 horas) 1. Inicie e verifique o estado do serviço web Apache (`httpd`). 2. Habilite o Apache para iniciar automaticamente no próximo arranque do sistema. 3. Crie um ficheiro de unidade `service` básico para uma aplicação simples, como um script de "Olá, mundo!" e ative-o.

- **Para Aprofundar** Explore a diferença entre os comandos `systemctl start`, `restart` e `reload`. Investigar a estrutura de um ficheiro de unidade `service` em `/etc/systemd/system/` também é um excelente próximo passo.

## 2. XINET.d (4 horas)

O `xinetd` (e o seu antecessor, o `inetd`) funciona como um "super-servidor" que gere a inicialização de serviços de rede que não precisam de estar ativos a todo o momento, como o `telnet` ou `ftp`. Ele espera por pedidos de conexão numa porta específica e, quando um pedido chega, inicia o serviço correspondente.

- **Conceito Chave: Servidor "On-Demand"** (1 hora)

O `xinetd` economiza recursos do sistema, pois os serviços geridos por ele só correm quando necessário, em vez de estarem sempre ativos.

- **Configuração e Gestão** (2 horas)

As configurações do `xinetd` estão nos ficheiros do diretório `/etc/xinet.d/`. Cada serviço tem o seu próprio ficheiro de configuração. Vamos analisar o ficheiro do serviço `telnet`:

```
service telnet
{
disable          = yes
id               = telnet-ipv4
type            = UNLISTED
...
}
```

A linha `disable = yes` é a chave: para ativar o serviço, deve ser alterada para `no`.

- **Exercício de Consolidação** (1 hora)

1. Encontre o ficheiro de configuração do serviço `ftp` no diretório `/etc/xinet.d/`. 2. Altere o valor da opção `disable` para habilitá-lo. 3. Reinicie o serviço `xinetd` para que as alterações entrem em vigor: `systemctl restart xinetd`. 4. Crie um novo ficheiro de configuração `/etc/xinet.d/my-service` para um serviço simples (por exemplo, um servidor de eco) e verifique se ele é iniciado corretamente.

---

## 3. TCPWrappers (4 horas)

O `TCPWrappers` é uma ferramenta de segurança simples mas eficaz para controlar o acesso a serviços de rede, atuando como um "firewall de nível de serviço". Ele permite a criação de regras de acesso (permitir/negar) baseadas em endereços IP ou nomes de host.

- **Conceito Chave: O Ciclo `hosts.allow` -> `hosts.deny`** (2 horas)

As regras são processadas numa ordem específica: o sistema verifica primeiro o ficheiro `/etc/hosts.allow`. Se uma regra corresponder, o acesso é concedido e o `hosts.deny` é ignorado. Se não houver correspondência, o sistema verifica o `hosts.deny`. Se uma regra corresponder, o acesso é negado.

- **Sintaxe e Exemplos** (1 hora)

A sintaxe geral é '[lista de serviços]: [lista de clientes]'. A lista de clientes pode ser um IP, um nome de host ou um wildcard.

Em `/etc/hosts.allow`:

```
# Permite acesso SSH a um IP específico e a todos os IPs da sub-rede 192.168.1.0/24
sshd: 192.168.1.100, 192.168.1.
```

Em `/etc/hosts.deny`:

```
# Nega acesso SSH a todos os outros IPs, exceto os da regra acima
sshd: ALL
```

- **Exercício de Consolidação** (1 hora)

1. Adicione uma regra para permitir o acesso ao serviço `ftp` a partir de um IP específico e a partir de um nome de domínio ('`meu.host.local`'). 2. Adicione uma segunda regra para negar o acesso a `ftp` a todos os outros IPs. 3. Use o comando `tcpdmatch` para testar se as suas regras estão a funcionar como esperado.

#### 4. NIS (6 horas)

O NIS (Network Information Service) é um sistema de diretório centralizado que permite que informações de contas de utilizadores, grupos e hosts sejam distribuídas por uma rede. É útil para ambientes de rede pequenos e uniformes.

- **Arquitetura NIS (Servidor Mestre e Escravo)** (3 horas)

O NIS é composto por um servidor Mestre, que mantém as informações de mapas (utilizadores, grupos, etc.), e servidores Escravos, que replicam os dados do Mestre. Os clientes contactam os servidores para obter as informações de autenticação. Os principais daemons são 'ypserv' (servidor NIS) e 'ypbind' (cliente NIS).

- **Configuração do Cliente** (2 horas)

Para configurar um cliente, é necessário definir o domínio NIS e indicar o servidor NIS. Isso é tipicamente feito nos ficheiros de configuração do cliente ('/etc/nsswitch.conf') e no comando 'ypbind'.

```
# Em /etc/nsswitch.conf, adicione "nis"
passwd:    files nis
group:     files nis
```

Após a configuração do cliente, podemos listar os utilizadores do servidor NIS com o comando ypcat.

```
$ ypcat passwd
```

- **Exercício de Consolidação** (1 hora)

1. Use o comando `ypwhich` para verificar a que servidor NIS a sua máquina cliente está conectada.
2. Use o comando `ypcat` para listar as contas de grupo disponíveis.
3. Crie um novo utilizador no servidor NIS e verifique se ele aparece na lista de utilizadores em uma máquina cliente.

#### 5. DHCP (10 horas)

O DHCP (Dynamic Host Configuration Protocol) é o protocolo padrão para atribuir configurações de rede (como endereços IP) a dispositivos de forma automática.

- **Conceito Chave: O Processo DORA** (4 horas)

O DHCP funciona através de um processo de quatro etapas: **D**iscover (descoberta), **O**ffer (oferta), **R**equest (pedido) e **A**cknowledge (confirmação). Uma imagem pode ilustrar bem este fluxo.

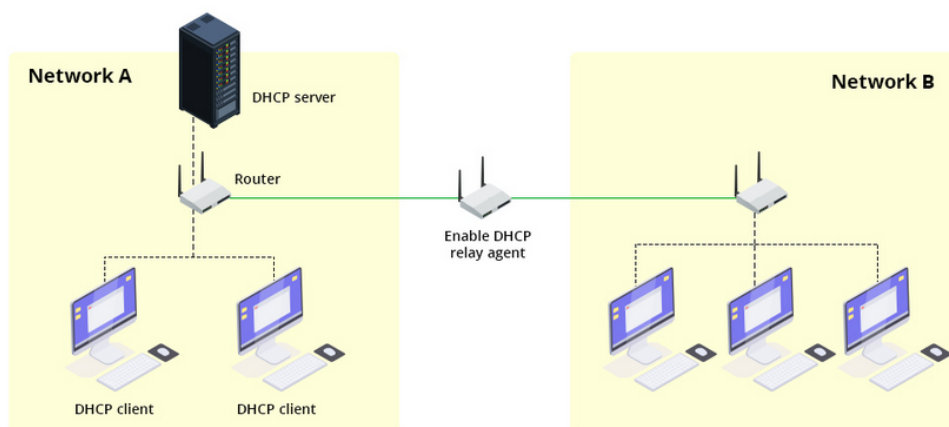


Figura 2: Diagrama do processo DORA (Discover, Offer, Request, Acknowledge). Fonte: Imagem encontrada via Google Images.

- **Exemplo Detalhado: Configuração do Servidor DHCP** (3 horas)

O ficheiro de configuração principal é o `/etc/dhcp/dhcpd.conf`. Vamos analisar uma configuração típica:

```
# Configuração para uma sub-rede
subnet 192.168.1.0 netmask 255.255.255.0 {
# Range de IPs dinâmicos
range 192.168.1.100 192.168.1.200;
# Gateway padrão (router)
option routers 192.168.1.1;
# Servidores DNS
option domain-name-servers 8.8.8.8, 8.8.4.4;
# Tempo de concessão
default-lease-time 600;
max-lease-time 7200;
}
```

- **Alocações Estáticas e Dinâmicas** (1 hora)

Para garantir que uma máquina específica receba sempre o mesmo IP (alocação estática), use a diretiva 'host' com o endereço MAC.

```
host meu-pc {
hardware ethernet 00:1A:2B:3C:4D:5E;
fixed-address 192.168.1.50;
}
```

- **Exercício de Consolidação** (2 horas)

1. Configure um servidor DHCP para uma sub-rede 10.0.0.0/24 com um 'range' de IPs dinâmicos.
2. Adicione uma entrada estática no ficheiro 'dhcpd.conf' para atribuir o IP '10.0.0.10' a uma máquina com um endereço MAC específico.
3. Reinicie o serviço e verifique nos logs se as concessões de IP estão a ser feitas corretamente.

## 6. DNS (12 horas)

O DNS (Domain Name System) é a base da internet, atuando como um "livro de endereços" que traduz nomes de domínio em endereços IP.

- **Conceito Chave: Resolução de Nomes** (6 horas)

Quando um utilizador digita um nome de domínio, o cliente DNS consulta o servidor DNS para obter o endereço IP correspondente, permitindo que a conexão seja estabelecida. Este processo é hierárquico, começando pelos servidores de raiz e descendo até ao servidor autoritativo para o domínio em questão.

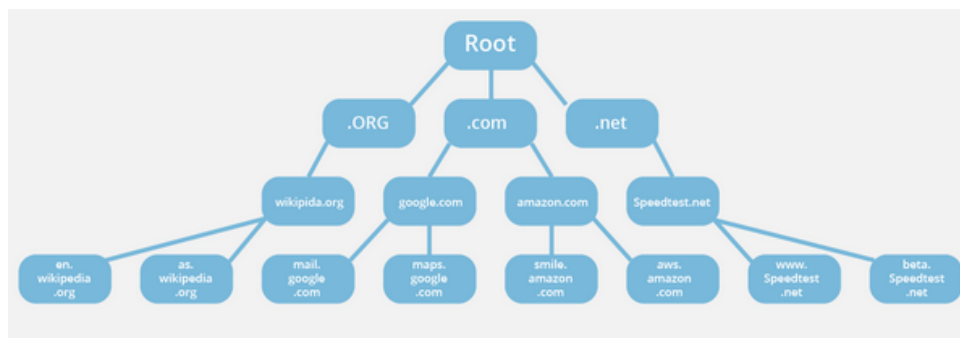


Figura 3: Fluxo de uma consulta DNS típica, mostrando a hierarquia de servidores. *Fonte: Imagem encontrada via Google Images.*

- **Tipos de Registos DNS (A, AAAA, CNAME, MX, PTR)** (4 horas)

Vamos explorar os registos mais comuns usados em ficheiros de zona:

- A: Mapeia um nome de domínio para um endereço IPv4.
- AAAA: Mapeia um nome de domínio para um endereço IPv6.
- CNAME: Cria um alias para outro nome de domínio.
- MX: Especifica o servidor de correio para um domínio.
- PTR: Usado para a pesquisa inversa, mapeando um IP para um nome de domínio.

- **Exemplo Detalhado: Ficheiro de Zona** (1 hora)

Um ficheiro de zona simples para `exemplo.com`:

```
$TTL 86400
@ IN SOA ns1.exemplo.com. admin.exemplo.com. (
2023010101 ; Serial
3600      ; Refresh
1800      ; Retry
604800    ; Expire
86400     ; Minimum TTL
)

@ IN NS ns1.exemplo.com.
@ IN A 192.168.1.10

www IN A 192.168.1.11
mail IN A 192.168.1.12
```

- **Exercício de Consolidação** (1 hora)

1. Crie um ficheiro de zona completo para um domínio fictício, incluindo um registo ‘A’ para a máquina principal, um ‘CNAME’ para o servidor web (‘www’), e um ‘MX’ para o servidor de correio. 2. Recarregue o serviço DNS para aplicar as alterações e teste a resolução de nomes usando os comandos ‘dig’ e ‘nslookup’.

—

## 7. LOGS (2 horas)

Os logs são ficheiros de registo que fornecem informações sobre o que está a acontecer no sistema e nas aplicações. São cruciais para a monitorização e a resolução de problemas.

- **Onde Encontrar Informação** (1 hora)

O diretório `/var/log` é o local central para a maioria dos logs do sistema e de aplicações. Cada ficheiro tem uma função específica (ex: `messages` para logs gerais, `auth.log` para autenticação).

```
/var/log/syslog: Mensagens gerais do sistema (Debian/Ubuntu)
/var/log/messages: Mensagens gerais do sistema (CentOS/Fedora)
/var/log/auth.log: Autenticação de utilizadores
/var/log/dmesg: Mensagens de arranque do kernel
```

- **Exemplo Detalhado: Analisar os Logs** (30 minutos)

Use `tail` para ver as últimas entradas de um ficheiro de log ou `grep` para procurar por mensagens específicas.

```
$ tail -f /var/log/messages
$ grep "sshd" /var/log/auth.log
```

Aprofunde a utilização do ‘journalctl’ para filtrar logs por serviço, tempo ou prioridade.

- **Exercício de Consolidação** (30 minutos)

1. Force um erro (por exemplo, ao tentar iniciar um serviço com a sintaxe incorreta). 2. Use o comando ‘journalctl -u [serviço]’ para encontrar a mensagem de erro e identifique o motivo do erro. 3. Use o ‘grep’ para filtrar as tentativas de login falhadas no ficheiro de log de autenticação (‘/var/log/auth.log’).