

# **PROJETO DE REDES**

[www.projetoderedes.com.br](http://www.projetoderedes.com.br)



**CIMCORP**

*Treinamento*

**ADMINISTRAÇÃO  
DE SISTEMAS**

*Solaris 2*

*Eduardo G. Coutinho*

# INDICE

---

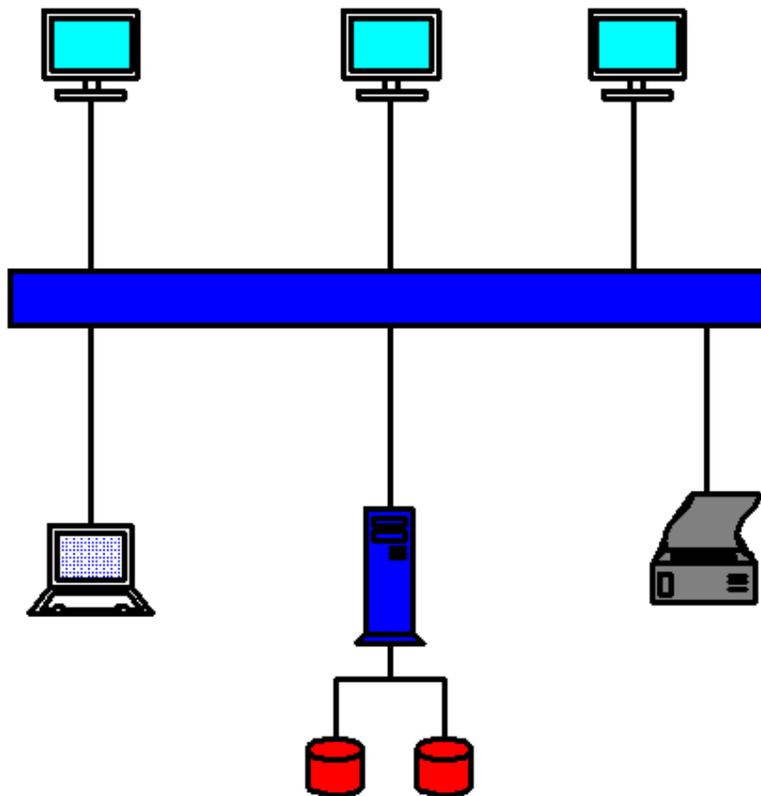
1. INTRODUÇÃO .....	4
1.1. AMBIENTE UNIX DISTRIBUÍDO .....	5
1.2. TIPOS DE EQUIPAMENTO .....	6
1.3. TAREFAS DO ADMINISTRADOR .....	7
1.4. TAREFAS DO ADMINISTRADOR (cont.) .....	8
1.5. ROOT .....	9
2. DISCOS E PARTIÇÕES .....	10
2.1. TERMINOLOGIA .....	11
2.2. ESTRUTURA DO DISCO .....	12
2.3. PRATO DO DISCO .....	13
2.4. PARTIÇÕES no Solaris .....	14
2.5. SISTEMAS DE ARQUIVOS E PARTIÇÕES .....	15
2.6. LAYOUT DO DISCO .....	16
2.7. INODES .....	17
3. Startup e Shutdown .....	18
3.1. BOOTING .....	19
3.2. BOOTING (cont.) .....	20
3.3. INTERRUPÇÕES NO SISTEMA .....	21
3.4. PARADA DO SISTEMA .....	22
3.5. PARADA DO SISTEMA (cont.) .....	23
3.6. OPÇÕES DE BOOT .....	24
4. INSTALAÇÃO DE SISTEMA OPERACIONAL .....	25
4.1. INSTALAÇÃO DE Solaris 2.x .....	26
5. CONFIGURAÇÃO DO KERNEL .....	27
5.1. PORQUE CONFIGURAR O KERNEL .....	28
5.2. Configuração do KERNEL .....	29
6. TCP/IP .....	30
6.1. Arquitetura TCP/IP .....	31
6.2. Fluxo de Mensagens .....	32
6.3. Endereços Ethernet .....	33
6.4. Endereçamento IP .....	34
6.5. Endereçamento IP (cont.) .....	35
6.6. Máscara e Subnetting .....	36
6.7. Máscara e Subnetting (Exemplo) .....	37
6.8. Protocolos da Família TCP/IP .....	38
6.9. ROTEAMENTO IP .....	39
6.10. Aplicações de Rede .....	40
6.11. TELNET .....	41
6.12. FTP - File Transfer Protocol .....	42
7. Network File System (NFS) .....	43
7.1. Network File System .....	44
7.2. Servidor NFS .....	45
7.3. Cliente NFS .....	46
7.4. Interação Cliente/Servidor .....	47
7.6. Importação de Sistema de Arquivos .....	49
7.7. Comandos Informativos .....	50
8. MAIL .....	51
8.1. Características do E-mail .....	52
8.2. Domínios .....	53
8.3. Funcionamento do Mail .....	54
8.4. SMTP - Simple Mail Transfer Protocol .....	55
8.5. Mail - Arquivos Necessários .....	56
8.6. Mail - Outros Arquivos .....	57
8.7. Instalação do sendmail .....	58

8.8. /etc/aliases .....	59
9. IMPRESSORAS .....	60
9.1. CONEXÕES .....	61
9.2. Comandos de Impressão .....	62
9.3. DESTRAVANDO IMPRESSORAS.....	63
10. BACKUP E RECUPERAÇÃO.....	64
10.1. BACKUPS .....	65
10.2. TAR.....	66
10.3. CPIO .....	67
10.4. DD.....	68
10.5. PREPARATIVOS PARA O BACKUP .....	69
10.6. BACKUP DE SISTEMAS DE ARQUIVOS.....	70
10.7. BACKUP DE VÁRIOS SISTEMAS DE ARQUIVOS EM UMA FITA .....	71
10.8. ESTRATÉGIAS PARA DUMP.....	72
10.9. RECUPERAÇÃO DE BACKUP.....	73
10.10. UFSRESTORE INTERATIVO.....	74
10.11. UFSRESTORE DE UM SISTEMA DE ARQUIVOS .....	75
11.12. UFSRESTORE DO /ROOT .....	76
11.13. UFSRESTORE DO /ROOT (Cont.).....	77
11. MANUTENÇÃO DE SISTEMA DE ARQUIVOS .....	78
11.1. MANUTENÇÃO DE SISTEMA DE ARQUIVOS .....	79
11.2. FSCK.....	80
11.3. FSCK (CONT).....	81
11.4. FASES DO FSCK.....	82
11.5. MONITORANDO ESPAÇO EM DISCO .....	83
11.6. /usr/etc/cron e /var/spool/cron/crontabs .....	84
11.7. /usr/bin/crontab e /var/spool/cron.....	85
11.8. AT .....	86
11.9. USO DE QUOTAS DE DISCO .....	87
12. SEGURANÇA .....	88
12.1. INTRODUÇÃO .....	89
12.2. CONTAS DOS USUÁRIOS.....	90
12.3. Criando conta de usuário.....	91
12.4. Segurança das Contas.....	92
12.5. Segurança do Sistema de Arquivos.....	93
12.6. Segurança do Sistema de Arquivos.....	94
12.7. Segurança da Rede .....	95
12.8. Segurança da Rede .....	96
12.9. Monitoração da Segurança.....	97
12.10. Monitoração da Segurança.....	98
12.11. Comandos que podem auxiliar na segurança .....	99
12.12. Ferramentas Úteis .....	100
12.13. Leitura Recomendada.....	101

# 1. INTRODUÇÃO

# 1.1. AMBIENTE UNIX DISTRIBUÍDO

- Cada estação de trabalho possui sua própria CPU, memória e sistema de I/O
- Procedimentos de backup envolvem várias máquinas e vários discos
- Cada estação de trabalho possui o seu próprio superusuário
- Utilitários tais como NFS e NIS tornam a administração diferente e mais fácil.
- sistema pode suportar múltiplas arquiteturas
- Vários níveis do sistema operacional podem ser suportados simultaneamente na rede



## 1.2. TIPOS DE EQUIPAMENTO

- **Standalone**
- **Servidor**
- **Diskless**
- **Dataless**
- **X-terminal**

### NOTAS

Uma estação de trabalho standalone possui uma área de root completa mais o diretório /usr em seu disco. Não necessita de serviços da rede para funcionar e faz o boot a partir de seu próprio disco. Pode ou não possuir uma unidade de fita acoplada.

Um servidor se caracteriza pela quantidade substancial de espaço em disco para uso de outras máquinas da rede, chamadas clientes. Normalmente possui uma unidade de fita.

Uma máquina diskless é aquela que necessita dos serviços da rede para acessar o disco de um servidor. Ela recebe o código para boot através da rede. Normalmente as áreas root e swap de uma máquina diskless residem em um mesmo servidor, mas o código executável e os arquivos dos usuários podem residir em várias máquinas da rede.

Uma máquina dataless é aquela que possui seu próprio disco para armazenar a área de swap e a área de root. Este disco pode também ser usado para armazenar dados de usuários. Arquivos e código executável necessários para sua operação (/usr) são obtidos da rede. Desta forma o código executável de diversas máquinas dataless é armazenado em um único servidor.

## 1.3. TAREFAS DO ADMINISTRADOR

### I. Instalação de Software

- A. Sistema operacional e atualizações
- B. Aplicações de correções
- C. Formatação e particionamento de discos

### II. Customização do software

- A. Kernel
- B. Terminais
- C. Modems
- D. Impressoras
- E. Rede

### III. Segurança

- A. Adição/remoção de usuários
- B. Criação de grupos de usuários
- C. Controle de acesso às máquinas
- D. Restringir acesso privilegiado às máquinas

## NOTAS

A administração de sistemas envolve a tarefa de sempre manter o sistema operacional atualizado, rodando a última versão disponível. Quando novas máquinas são adquiridas estas devem ser configuradas, o sistema operacional instalado e conectadas à rede. Esta instalação vai depender da configuração da máquina, se diskless, dataless, standalone ou servidora. É recomendável sempre formatar o disco antes de instalar o sistema operacional.

O kernel que vem com a máquina raramente atende às necessidades do usuário. Ele deve ser configurado e adaptado às condições do ambiente operacional. Hardware acrescentado à configuração básica normalmente requer a geração de um novo kernel para inclusão dos drives de dispositivo.

A segurança é um aspecto que nunca deve ser negligenciado. Os novos usuários devem ser gerenciados de forma a só poderem fazer aquilo a que forem designados. Enfatizar a importância do sigilo da senha de acesso, restringir o acesso privilegiada operações do sistema. Criar grupos de usuários de modo que , pessoas trabalhando em um mesmo projeto pode compartilhar seus dados. Ao mesmo tempo impede o acesso de usuários não pertencentes ao grupo.

# 1.4. TAREFAS DO ADMINISTRADOR (cont.)

## I. Serviços de rede

- A. NFS
- B. NIS

## II. Manutenção do Sistema

- A. Manutenção do sistema de arquivos
- B. Backups
- C. Restore

## III. Estabelecer comunicação

- A. E-mail
- B. Usenet
- C. Servidor de FTP anonymous
- D. WWW

## NOTAS

Um dos recursos mais importantes que o sistema Unix oferece é a capacidade de troca de mensagens entre usuário.

NFS (Network File System) permite que máquinas montem diretórios residentes em outras máquinas da rede. O acesso a estes diretórios é feito de maneira transparente para o usuário que tem a impressão de estar usando arquivos locais. NIS (Network Information Services) permite que informações importantes de configuração e outras sejam compartilhadas por diferentes máquinas.

A tarefa mais importante de um administrador de sistemas é fazer backups. Inevitavelmente dados serão perdidos, defeitos em disco ocorrerão. Backups devem ser realizados de forma sistemática e regular. Os comandos dump e restore existem para facilitar essa tarefa. O comando fsck é usado automaticamente para verificar a consistência do sistema de arquivos durante o boot do sistema e reduzir a possibilidade de um sistema de arquivos se corromper.

## 1.5. ROOT

- **Conta do superusuário**
- **Cuidados especiais com a password**
- **Nunca logar diretamente como root**
- **Use shell restrita para root (`/usr/lib/rsh`)**

### NOTAS

A administração do sistema é função da conta root. Root é o nome do usuário mais privilegiado do sistema, também conhecido como superusuário. Esta conta pode fazer qualquer coisa no sistema. Como de praxe em sistemas Unix, nenhum ato do superusuário é questionado. Devido a este poder, é de extrema importância que a senha da área root seja protegida com muito cuidado. Ela não deve ser facilmente deduzida (nome da máquina, nome do administrador invertido, iniciais, nome do cachorro, nome do carro, placa do carro, etc.) e deve ser mudada periodicamente. recomenda-se também que tarefas de rotina não sejam executadas sob a área root. Entrar nessa área apenas para desempenhar tarefas que requerem privilégios especiais e em seguida voltar a conta comum.

## **2. DISCOS E PARTIÇÕES**

## 2.1. TERMINOLOGIA

### **Controladoras de discos**

- dispositivo que se comunica com o disco
- uma controladora suporta vários discos
- uma estação de trabalho pode ter vários discos
- SCSI - Small Computer System Interface
- IDE - Integrated Drive Electronics

### **Driver de Dispositivo**

Software residente no kernel que faz a controladora operar

### **Nome de entrada do dispositivo**

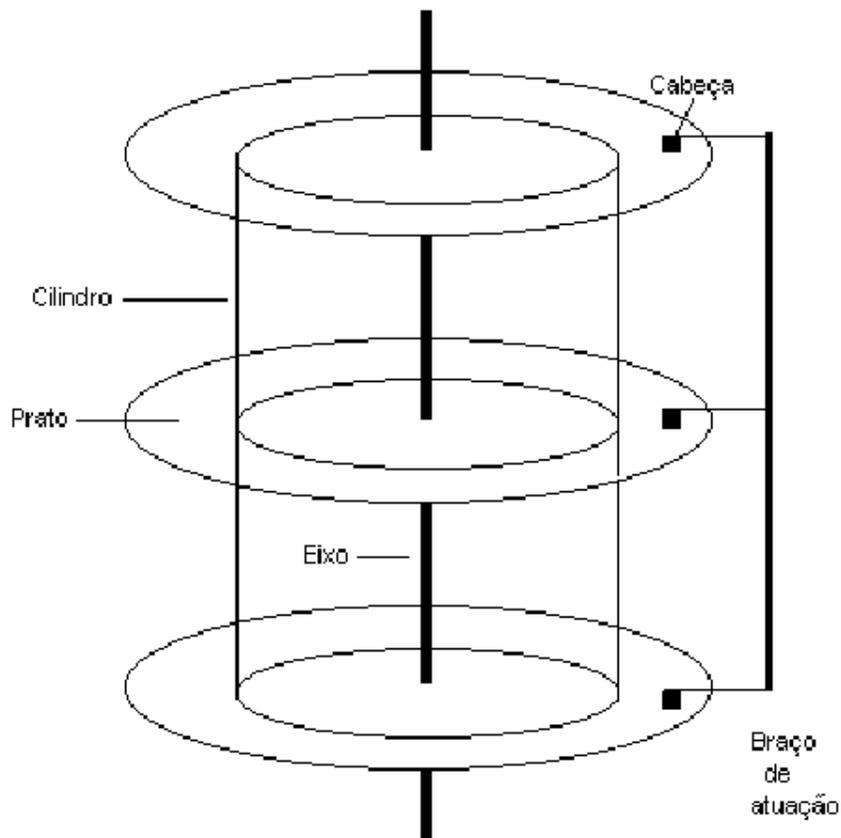
- É o nome lógico definido no sistema de arquivo que indica o driver de dispositivo no kernel.

## NOTAS

Como o nome indica, a controladora de discos é o dispositivo que controla a unidade de disco. A controladora se encarrega de detalhes tais como operações de baixo nível no disco, checagem de erros, movimentar os cabeçotes de leitura e gravação, transferência de dados e disposição dos dados no disco. A controladora de disco provê a interface da unidade de disco com o resto do sistema.

O driver de dispositivo (device driver) é o software que opera a controladora. Este software reside no kernel. Um dispositivo cujo software não houver sido incorporado ao kernel não poderá ser usado.

## 2.2. ESTRUTURA DO DISCO

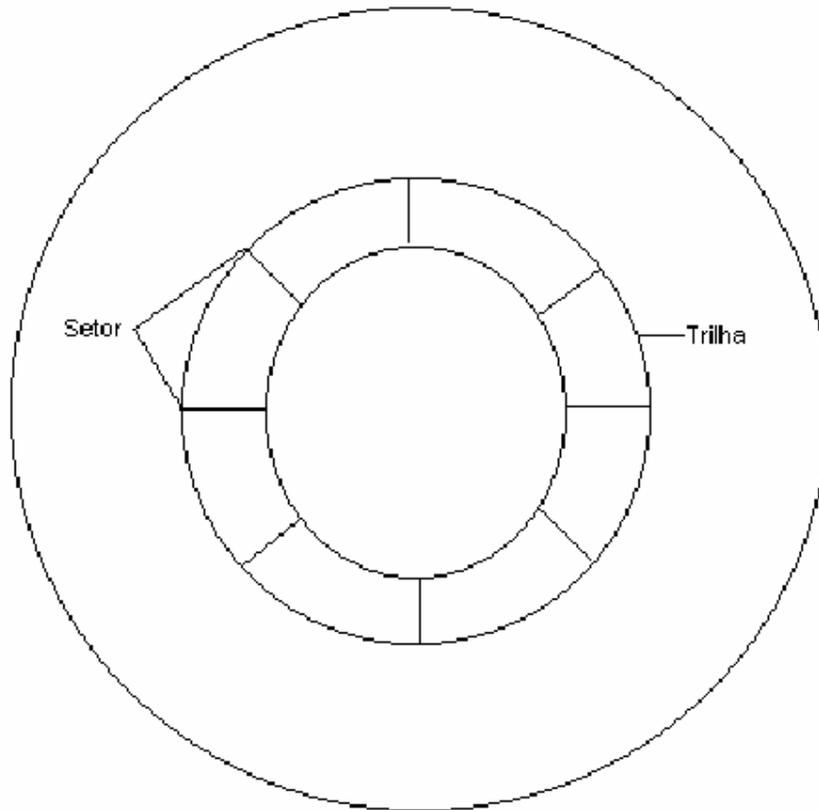


### NOTAS

Um disco rígido é fisicamente composto por uma série de pratos ligados a um eixo. A informação contida nestes pratos é processada pelas cabeçotes de leitura e gravação. Estes cabeçotes são acoplados ao braço atuador. Quando lendo ou gravando dados, os cabeçotes são movidos conjuntamente pelo braço atuador. O movimento radial é chamado "seeking", que é uma das razões pelas quais as especificações de disco frequentemente mencionam o "seek time". Outros fatores permanecem constantes, quanto mais rápido o "seek time", mais rápido é o acesso aos dados no disco.

A compressão da estrutura do disco pode ser útil na sua formatação, pois podem ser necessárias informações como o número de cilindros, cabeças, setores por trilha, velocidade de rotação.

## 2.3. PRATO DO DISCO



### NOTAS

O prato do disco é dividido em trilhas, cilindros e setores. A trilha é a porção do disco que passa debaixo de uma cabeça simples estacionada durante a rotação do disco, formando um anel de 1 bit de largura. O cilindro é composto de um conjunto de trilhas descrita por todas as cabeças em uma simples posição de procura. Cada cilindro é equidistante do centro do disco. A trilha é dividida em segmentos de setores, os quais são a unidade básica de armazenamento.

SunOs usa o sistema de arquivos de Berkeley com grupo de cilindros. Um grupo de cilindro compreende 32 ou menos cilindros (default é 16) no disco. Cada grupo de cilindros tem uma cópia redundante do superbloco, espaço para inodes, lista de blocos livres e, lista de uso de blocos de dados dentro grupo de cilindro. Através dos grupo de cilindro se reduz a soma do movimento de cabeça, em requerida em média para acessar um arquivo. O inode descrevendo o arquivo, e os dados para o arquivo, estão igualmente na mesma área física do disco. A posição do superbloco redundante dentro de cada grupo de cilindro é variada, de tal modo que eles não residam no mesmo prato. Isso ajuda a assegurar que você pode recuperar as informações no caso da perda do superbloco primário.

## 2.4. PARTIÇÕES no Solaris

```
#format
partition> print
Current partition table (original sd3):
Part Tag          Flag* Cylinders   Size          Blocks
0   root           wm     0-51        18.28MB       (52/0/0)
1   swap           wu    52-143      32.34MB       (92/0/0)
2   backup         wm    0-1150     404.65M       (151/0/0)
3   unassigned     wm     0           0              0
4   unassigned     wm     0           0              0
5   -              wm   144-627    170.16MB      (484/0/0)
6   usr            wm   648-1026   140.27MB      (399/0/0)
7   home           wm  1027-1150  43.59MB       (124/0/0)
```

Flag indica writeable (w), e mountable (m) e umountable(u).

### NOTAS

SunOs 5.X (Solaris 2.X) dividiu os nomes dos devices em três espaços: físico, lógico, SunOS compatível. Mas para a administração normalmente são utilizados os nomes lógicos.

Os nomes lógicos dos discos contém o número da controladora, o número "target" (alvo) se o disco está no "bus" (portadora), o número do disco e o número do "slice" (partição). O slice root é 0, slice 1 é swap, slice 2 cobre o disco inteiro. Cada device de disco tem uma entrada nos diretórios /dev/dsk e /dev/rdisk para os devices de disco bloco e "raw" respectivamente.

Assim, o nome lógico /dev/sd0a do SunOS 4.1.X será no Solaris 2.X:  
/dev/dsk/c0t0d0s0

**onde:**

**c0** número da controladora

**t0** número de alvo (target)

**d0** número do disco

**s0** número do slice (partição)

Este nome é na verdade um link simbólico para:

/devices/sbus@1,f8000000/esp@0,8000000/sd@0,0:a (nome físico do device).

Foi implementado o comando prtvtoc no lugar do dkinfo do SunOS 4.1.X.. Este comando é utilizado da seguinte maneira:

```
prtvtoc /dev/rdisk/c0t0d0s2
```

## 2.5. SISTEMAS DE ARQUIVOS E PARTIÇÕES

### SISTEMAS DE ARQUIVOS

Estrutura de arquivos e diretórios

Blocos de 8192 bytes com fragmentos de 1024

- Superblocos, blocos de grupo de cilindros, inodes, blocos de dados
- Bloco de boot
- Criado pelos comandos `mkfs` ou `newfs`
- Precisa ser ``montado".
- Reparado pelo `fsck`

### PARTIÇÕES

- Blocos do disco (setores de 512 bytes)
- Cilindros, trilhas, setores
- Chamados de dispositivos "raw".

### NOTAS

Partições e sistema de arquivos são duas maneiras diferentes de ver a organização de um disco. Para alguns procedimentos como backups e restores o uso de partições é mais conveniente. A partição é o dispositivo de disco lógico.

O sistema de arquivo é uma estrutura de dados que contém arquivos e diretórios. Os sistemas de arquivos (um por partição) são criados pelos comandos `mkfs` ou `newfs`. Problemas no sistemas de arquivo podem ser corrigidos pelo comando `fsck`.

## 2.6. LAYOUT DO DISCO

LABEL
ÁREA DE BOOT (bootstrap)
SUPERBLOCO PRIMÁRIO
BLOCO DE RESUMO DO GRUPO DE CILINDROS
TABELA DE INODE
ÁREA DE BLOCOS DE DADOS
BACKUP DO SUPERBLOCO
BLOCO DE RESUMO DO GRUPO DE CILINDROS
TABELA DE INODES
ÁREA DE BLOCOS DE DADOS

### NOTAS

O label fica no primeiro setor da primeira partição. Os próximos 15 setores contêm a área de boot. Seguindo o label na partição root e em todas as outras partições vêm as séries de grupos de cilindros. Cada grupo de cilindro contêm um bloco de resumo do grupo de cilindros, uma tabela de inodes, e a respectiva área de blocos de dados.

O superbloco primário é mantido na memória de cada grupo de cilindro guarda uma cópia. O tamanho default para os blocos de dados é 8192, dividido em 8 fragmentos de 1024 cada.

O bloco de resumo do grupo de cilindros guarda: o número de inodes e blocos de dados; ponteiros para o último bloco, fragmento e inode usado; o número de fragmentos livres; o mapa de inodes usados, o mapa de inodes livres.

Um disco pode conter mais de um sistema de arquivos que o UNIX interpreta como um hierarquia de diretórios e arquivos. O UNIX interpreta as solicitações para criar, ler, gravar e apagar arquivos e as executa acrescentando e apagando entradas na lista de inodes e blocos livres.

Os diretórios contêm apenas duas informações sobre cada arquivo: o nome e o número do inode.



## **3. Startup e Shutdown**

## 3.1. BOOTING

### Auto teste, memória

#### Identificação

- Modelo
- Tipo de teclado
- Hostid
- Endereço ethernet

#### Pesquisa o barramento à procura do dispositivo de boot

- SCSI (sd0)
- Network (le0)
- PROM lê a trilha de boot (boot block)
- Carrega o programa de boot (/boot)
- programa de boot lê o kernel

## NOTAS

Quando a máquina é ligada segue uma seqüência determinada inicialmente pela PROM da placa de CPU. Se todos os testes iniciais forem completados com sucesso, o controle é passado então ao dispositivo de boot. No Sun0s o dispositivo de boot é por default a partição a do drive 0, mas isto pode ser reprogramado através da EEPROM. Se o drive definido não existe ou se o programa de boot (/boot) não for encontrado, o sistema tentará fazer a carga através da rede.

O startup a partir do dispositivo de boot procede em estágios até que o Kernel esteja carregado na memória. Neste ponto o controle é passado para o kernel.

O kernel exhibe informações a respeito de seu tamanho e histórico. Ele então pesquisa o barramento para confirmar se os dispositivos especificados no kernel realmente existem. Se o dispositivo não existe, então o kernel o ignora. Cada dispositivo que precisa ser testado e que não existe atrasa a carga do sistema. Conseqüentemente, uma das razões pelas quais se recomenda a configuração do kernel é diminuir o tempo de boot.

O dispositivo root, swap e dump são identificados.

## 3.2. BOOTING (cont.)

- Kernel inicializa o sistema
- Procura dos dispositivos conhecidos pelo Kernel
- Inicializa o processo `init`
- `Init` lê o arquivo `inittab`(Solaris 2)
- Executa os scripts RC
- Sistema em modo multiusuário

### NOTAS

O kernel invoca o `init` no último estágio do boot. `Init` dá um `fork` para o background para rodar durante a existência do sistema. Depois que o `init` é carregado, ele invoca os shell scripts RC na seqüência definida por cada sistema. Estes scripts executam várias tarefas necessárias para colocar o sistema em modo multiusuário.

Entre estas tarefas está a verificação da consistência dos sistemas de arquivos. Se problemas sérios são detectados pelo `fsck`, o processo de boot pode ser impedido de continuar até que seja feita uma verificação manual dos sistemas de arquivos. Neste caso o sistema é carregado apenas em modo de leitura e o comando `fsck` pode ser utilizado. Entretanto, normalmente as inconsistências são corrigidas automaticamente pelo `fsck`.

Os scripts providenciam a montagem dos discos e os daemons padrão, depois carregam daemons de rede e sistemas de arquivos remotos. Por fim inicializa o sistema em modo multiusuário.

## 3.3. INTERRUPÇÕES NO SISTEMA

<STOP> A

Aborta temporariamente o sistema nos monitores de máquinas *Sun*

Ok go

Retoma as atividades do sistema

Ok sync

Sincroniza os discos e recarrega o sistema (reset)

### NOTAS

As máquinas *Sun* com monitores de mapa de bits e teclado aceitam a combinação das teclas <Stop> e "a" como um sinal para parar o sistema. A tecla BREAK é usada em sistemas que usam terminais ASCII como console. Estas interrupções jogam o console para o nível PROM que aceita vários comandos. O comando "go" (continue) permite a recuperação do sistema paralisado.

Este tipo de interrupção não avisa aos usuários que o sistema será encerrado. Além disto, os sistemas de arquivos não são sincronizados, o que poderá acarretar inconsistências que poderão necessitar correção manual com o programa fsck.

O comando sync tenta sincronizar os discos e gera um core. Este procedimento deve ser adotado no lugar de simplesmente desligar o sistema após um interrupção com STOP-A ou BREAK. O comando sync tenta rebotar automaticamente após sincronizar os sistemas de arquivos.

## 3.4. PARADA DO SISTEMA

### **Halt**

- Executa imediatamente
- Retorna ao prompt da PROM (Sun)
- Sincroniza os discos

### **Reboot**

- Executa imediatamente
- Sincroniza os discos
- Recarrega o sistema
- Verifica as partições com fsck

### **fasthalt e fastboot**

- Similares as anteriores, mas não executam fsck
- quando o sistema é recarregado.
- Presentes apenas se o pacote de compatibilidade com BSD (SUNWucb) estiver instalado

## NOTAS

Os comandos halt, fasthalt e fastboot sincronizam os discos antes de parar o sistema.

Os comandos fasthalt e fastboot criam um arquivo chamado /fastboot. Quando este arquivo existe, o script /etc/rc não roda o comando fsck. Este procedimento só é recomendado em casos especiais, como por exemplo, teste do kernel do sistema. Uma vez que a máquina esteja novamente pronta para ser usada pelos usuários, rodar o comando fsck.

## 3.5. PARADA DO SISTEMA (cont.)

### Shutdown

- Sincroniza os discos
- Alerta os usuários e clientes NFS
- Impede logins

### Exemplos:

```
shutdown 1400          (encerra o sistema às 14:00 h)
shutdown +15          (encerra o sistema em 15 minutos)
shutdown -h +30       (encerra o sistema com halt)
shutdown -r +10       (encerra e recarrega o sistema)
shutdown now          (encerra o sistema imediatamente)
shutdown -k 1000      (simula o shutdown do sistema)
```

## NOTAS

O comando shutdown é um procedimento ordenado de encerramento que alerta os usuários e clientes NFS. Este comando aceita opções, tempo de encerramento e mensagens a serem enviadas aos usuários.

As opções determinam o modo como o sistema será encerrado. A hora de encerramento e uma instrução para rebootar podem ser incluídas na linha de comando.

No caso do Solaris 2.X há diferentes versões do comando shutdown. A versão residente em /usr/sbin/shutdown possui opções diferentes da citadas ao lado.

Usa-se o comando como:

```
shutdown [-y] [--g segundos] [-i nível_do_sistema]
```

### Exemplo:

```
shutdown -y -g300 -i0
(leva o sistema para o nível 0 (derruba) em 5 minutos sem perguntas)
```

## 3.6. OPÇÕES DE BOOT

### Forma Geral

Ok boot device(controller#, unit #, file#) pathname args

### Boot default

Ok boot

### Boot em modo single user

Ok boot -s

### Boot através da rede

Ok boot le0

## NOTAS

O boot pode ser efetuado com várias opções. Estas opções incluem a controladora a partir de onde fazer o boot e o arquivo a ser usado. As máquinas Sun podem fazer o boot a partir de:

- Qualquer partição lógica do disco
- Um sistema NFS cujo nome seja fornecido pelo daemon

O monitor PROM informa quais os dispositivos de boot se você entrar com o comando help boot. Esta lista fornece a ordem pela qual os dispositivos são testados. Não há nenhuma garantia que um dispositivo listado esteja presente ou funcione.

Cada Unix possui um procedimento próprio de startup, e suas peculiaridades devem ser procuradas nos manuais específicos.

# **4. INSTALAÇÃO DE SISTEMA OPERACIONAL**

## 4.1. INSTALAÇÃO DE Solaris 2.x

Fazer o boot a partir de um cdrom  
Identificar o Sistema (fornecer hostname, IP e máscara)  
Será executado o SunInstall (instalação)  
Escolha entre instalação Initial ou UPGRADE  
Selecione o software desejado  
Para um servidor, configurar os clientes  
Particione os discos de acordo com o software  
Começar a instalação

# **5. CONFIGURAÇÃO DO KERNEL**

## 5.1. PORQUE CONFIGURAR O KERNEL

- Obter maior velocidade
- Reduzir uso de memória
- Adequar-se à configuração do sistema
- Acrescentar novos dispositivos de hardware
- Acrescentar novos softwares

### NOTAS

O kernel que vem junto com o sistema é configurado para suportar todos os dispositivos de uma determinada arquitetura. Na realidade, nenhuma máquina terá conectada a ela todos estes dispositivos a um só tempo. O código do kernel ocupa espaço na memória. Para reduzir o tamanho do kernel, elimine os módulos que não são necessários, reduzindo assim memória para uso de programas e conseqüentemente, aumentando a performance do sistema.

Durante o boot, o sistema pesquisa cada dispositivo do kernel para confirmar sua existência. Se o dispositivo não responde, o sistema desiste após algum tempo e passa para o próximo dispositivo. Para eliminar este período de espera requerido pelo timeout, configure o kernel de forma a que apenas os dispositivos existentes no sistema sejam testados.

Hardware adicional - acréscimo de discos, placas, mudanças para monitores coloridos - requerem suporte do kernel. À medida que o sistema cresce, o kernel deve ser reconfigurado para suportar o novo hardware que lhe é acrescentado. Isto também pode se aplicar ao software. Programas aplicativos podem incluir drivers de dispositivos que precisam ser incluídos no kernel. Estas informações são normalmente encontradas na documentação dos distribuidores independentes de software.

Para reconfigurar o kernel em equipamentos **Sun** utilizamos o comando  
Ok boot -r

## 5.2. Configuração do KERNEL

### SunOS 4.x.x

```
cd /usr/kvm/sys/sun4c/conf
cp GENERIC MEU_KERNEL
vi HOSTNAME (modificar o necessário)
config MEU_KERNEL
cd ../../MEU_KERNEL
make
mv /vmunix /vmunix.old
cp vmunix /
reboot
```

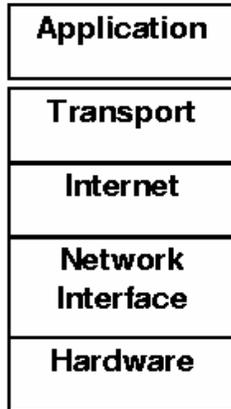
### NOTAS

No caso do novo kernel não funcionar faça o boot com:  
Ok boot vmunix.old

# 6. TCP/IP

## 6.1. Arquitetura TCP/IP

### Conceptual Layer



## NOTAS

Uma arquitetura de rede é definida pelas **camadas** ou níveis que a compõem, pela **interface** entre essas camadas e pelas regras de comunicação entre camadas de mesmo nível em máquinas distintas, regras estas conhecidas como **protocolo**.

O objetivo da divisão em camadas é permitir a modularização do software, permitindo que as alterações sejam localizadas e transparentes aos outros níveis não afetados.

Os módulos de software de protocolo em cada máquina podem ser representados como camadas empilhadas. Cada camada cuida de uma parte do problema.

Existem duas regras importantes para o entendimento da divisão do software de rede em camadas:

- 1- A camada inferior fornece serviços à camada superior.
- 2- O protocolo de nível N no nó destino tem que receber o mesmo objeto enviado pelo protocolo de nível N no nó origem

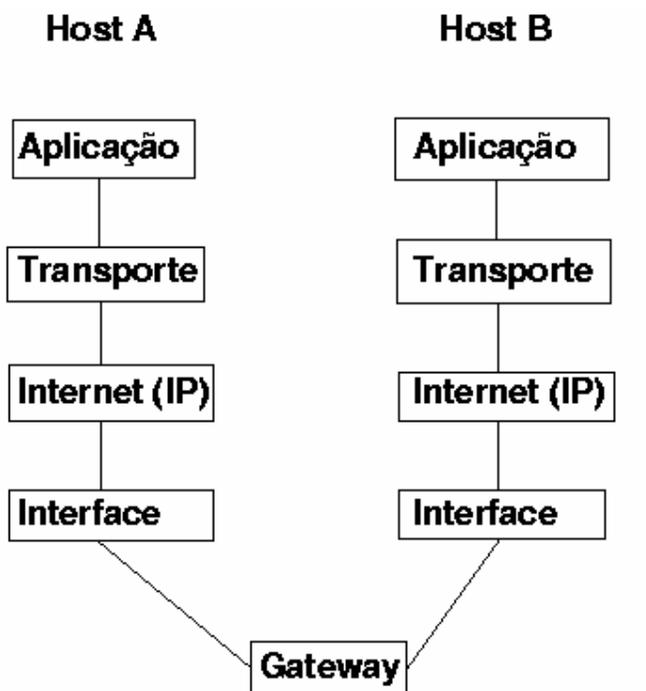
A arquitetura TCP/IP possui apenas 4 níveis:

- nível 1: interface de rede
- nível 2: Internet ou camada IP
- nível 3: transporte
- nível 4: aplicação

O nível 1 lida com o meio de comunicação, utilizando endereços físicos. os níveis 2 e 3 são incorporados ao sistema operacional. O nível 4 pode ser escrito por usuários. os níveis 2, 3 e 4 usam endereços IP.

O diagrama conceitual da primeira figura mostra a camada Internet entre uma camada de protocolo superior e uma camada de interface de rede. A segunda figura é mais realística e mostra que o software IP pode se comunicar com múltiplos protocolos de nível superior e múltiplas interfaces de rede.

## 6.2. Fluxo de Mensagens



### NOTAS

Transferir um mensagem de uma aplicação numa máquina para outra aplicação em outra máquina significa transferir a mensagem para sucessivas camadas inferiores na máquina do transmissor, transferir a mensagem através da rede e transferir para sucessivas camadas superiores na máquina destino.

Nas máquinas intermediárias o datagrama que chega através de uma camada de interface de rede sobe até a camada IP que o roteará de volta para uma outra rede através de sua outra interface de rede. Este datagrama será capturado por outra máquina intermediária e passará pelo mesmo processo até chegar à máquina destino. Essas máquinas "intermediárias" são chamadas GATEWAYS na nomenclatura Internet.

Conforme a mensagem passa por diferentes camadas de rede e é encapsulada por diferentes protocolos, ela assume diferentes nome como pacote, datagrama, e frame.

## 6.3. Endereços Ethernet

- Endereço único no mundo, estabelecido pelo IEEE
- 6 bytes
- Codificado por hardware

Ex. 00-00-1D - 00-26-A3  
3 bytes 3 bytes

### **Broadcast**

Recebido por todas as estações no mesmo segmento de rede

Ex. FF-FF-FF-FF-FF FF

## NOTAS

Cada placa Ethernet possui um endereço físico definido na fábrica.

Quando um endereço Ethernet é utilizado como endereço destino num pacote, este só será decodificado pela estação que possuir aquele específico endereço.

O endereço broadcast é utilizado por certos protocolos para comunicação com todos os nós da rede, quando não se conhece qual nó que pode atender uma solicitação.

**OBS.: Não serve para localização de máquinas na rede.**

## 6.4. Endereçamento IP

- Distribuídos pelo SRI-NIC
- Representados em notação decimal
- Intervalo de 0.0.0.0 a 255.255.255.255. Número de 32 bits representado em 4 porções de 8 bits escritas em decimal. Ex: Endereço IP 0xFEDA5E67 é representado como 254.218.94.103
- Máscara separa parte "rede" da parte "máquina" e segue o mesmo padrão numérico do número IP
- Endereço de rede e de máquina com valor 0 são inválidos
- 127.0.0.0 é reservado para teste de loopback
- Qualquer porção do endereço formada por 1's é considerada um broadcast

### NOTAS

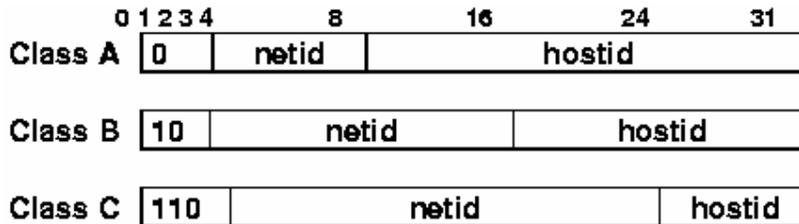
Os endereços Internet são administrados pelo NIC (Network Information Center, SRI Internacional).

Como os endereços IP codificam a rede e a máquina dentro da rede, eles não especificam uma máquina, mas sim uma conexão à rede.

Problema: se uma máquina muda de uma rede para outra o se endereço IP deve mudar.

O endereço IP broadcast é mapeado ao broadcast do hardware.

## 6.5. Endereçamento IP (cont)



### NOTAS

Gateways usam o endereço de rede para o roteamento de datagramas IP.

A parte "máquina" do endereço identifica unicamente uma estação dentro de uma rede específica.

O endereço Internet é um número de 32 bits que representa tanto a rede quanto a máquina dentro da rede. Existem três tipos de endereços: classes A, B, C. O primeiro, segundo ou terceiro bits determinam o tipo de rede.

**Classe A** - O primeiro bit é 0; os próximos sete bits representam o número da rede e variam de 1 a 126. Os últimos 24 bits representam o número da máquina.

**Classe B** - Os dois primeiros bits são 1 e 0 respectivamente. Os bits de 3 a 8 variam entre 128 a 191 e incluem os próximos 8 bits que variam de 1 até 254, perfazendo um total de 14 bits para o número da rede. Os últimos 16 bits representam o número da máquina.

**Classe C** - Os primeiros três bits são 1, 1, e 0 respectivamente. Os bits de 4 a 8 variam de 192 a 223 e incluem os próximos dois campos de 8 bits (cada um deles variando de 1 a 254), perfazendo um total de 21 bits para o número da rede. Os últimos 8 bits representam o número da máquina.

## 6.6. Máscara e Subnetting

### Subnetting

Permite que se divida a porção de máquina em 2 partes:

- mais bits para rede
- menos bits para máquinas

Utiliza-se uma nova máscara para identificar a nova parte de redes

A nova porção do endereço usada para rede é conhecida como subnet

```

143      106      1      45
10001111 01101010 00000001 00101101
11111111 11111111 00000000 00000000

```

Classe B ---> usando máscara 255.255.0.0

```

143      106      1      45
10001111 01101010 00000001 00101101
11111111 11111111 11111111 11000000

```

Classe B ---> máscara 255.255.255.192

## NOTAS

Máscara 255.255.0.0 possui 65.534 máquinas na mesma rede.

Máscara 255.255.255.192 possui 1022 subredes com 62 máquinas cada.

Fórmula genérica para calcularmos quantas máquinas ou subredes temos disponíveis utilizando determinada máscara:

$$X = 2^n - 2$$

Onde n é o número de bits utilizados para especificar a parte "rede" ou a parte máquina do endereço IP.

## 6.7. Máscara e Subnetting (Exemplo)

**Dado o número IP 143.106.1.45**

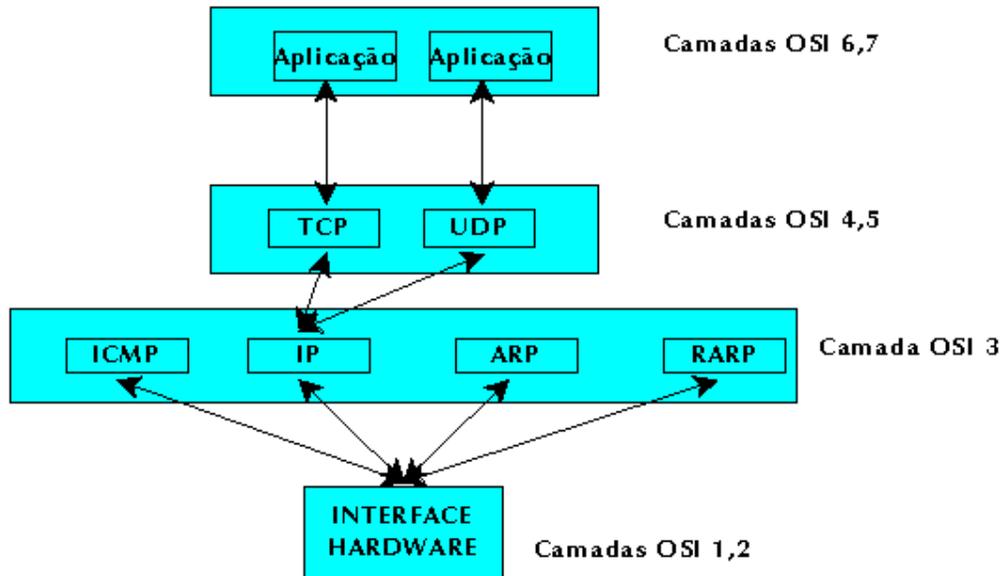
**Dada a máscara 255.255.0.0**

- Endereço de rede: 143.106.0.0
- Broadcast nesta rede: 143.106.255.255

**Dada a máscara 255.255.255.192**

- Endereço de rede: 143.106.1.0
- Broadcast nesta rede: 143.106.1.63

## 6.8. Protocolos da Família TCP/IP



### NOTAS

Os repetidores trabalham no nível 1 OSI (ao nível de sinais elétricos).

As pontes (bridges) operam no nível 2 OSI. Examinam endereços destino de todos os frames e tomam decisões quanto à necessidade de retransmiti-los para uma determinada rede.

Os roteadores lêem as informações até o nível 3 (endereçamento IP), apenas nos frames a eles endereçados.

## 6.9 ROTEAMENTO IP

### Decisão:

- Baseada em tabelas de rotas que ficam no kernel do S.O.

### RIP (Router Internet Protocol)

Tabela pode ser estática ou dinâmica

### Dinâmica:

- `routed - etc/gateways`

### Hops

- Número de gateways até a rede específica

Comando `netstat -r` exibe a tabela de roteamento

### Exemplo:

```
#netstat -r

Routing Table:
  Destination          Gateway                Flags  Ref    Use  Interface
-----
localhost             localhost             UH      0    450   lo0
telnet32.telnet.com.br 192.9.200.69         UGH     0     4
telnet33.telnet.com.br 192.9.200.220        UGH     0     0
200.246.118.251        192.9.200.220        UGH     0     0
200.246.118.249        atlantic              UGH     0     8
172.16.10.0           172.16.10.1         U       4     0    le0
192.9.200.0           wall                  U       4    442   le0:1
200.246.118.0         wall.cimcorp.com.br  U       2    108   qe0
BASE-ADDRESS.MCAST.NET wall                  U       4     0    le0:1
default               one.telnet.com.br    UG      0   4665
#
```

## NOTAS

Quando se diz "roteie de acordo com a tabela" pretende-se dizer: pegue o endereço do gateway que está na tabela, que necessariamente pertence à alguma rede da máquina onde está o pacote, use ARP para traduzir esse endereço IP para um endereço físico, encapsule esse datagrama num frame e envie para a interface de rede adequada.

## 6.10. Aplicações de Rede

- Processos clientes X servidores
- Servidores estão sempre ativos esperando conexões
- Clientes são criados assincronamente

### Identificação da conexão

- IP origem
- IP destino
- Porta origem
- Porta destino

**Porta destino: Well-known ports**

**Porta origem: alocação dinâmica**

### Arquivos

```
/etc/services
/etc/inetd.conf
```

## NOTAS

Nas aplicações de rede existe sempre um processo cliente numa máquina que dispara uma conexão, e um processo servidor em outra que tem que estar preparado para aceitar várias conexões simultâneas. Para isso existe para cada tipo de processo servidor o conceito de um processo "master" que aceita novas conexões e cria processos escravos do mesmo tipo para lidar com cada conexão. O processo "master" nunca morre (exceto em condições excepcionais) e o processo cliente tem uma duração finita.

Esses processos servidores são chamados "daemons" na linguagem UNIX e seus nomes terminassem com a letra "d". Existe um daemon genérico chamado inetd que aceita conexões "em nome" de vários tipos de processos, como telnet e finger.

Surge a questão: como identificar uma conexão unicamente, tanto na máquina origem quanto na máquina destino? Da explanação anterior fica claro que apenas os IP de destino e origem não bastam. Solução: associar a cada aplicação ( ou serviço ) um número de porta padronizado. Assim por exemplo, telnet tem o número 23; ftp, 21 e SMTP, 25. O melhor meio de pensar numa porta é como numa fila. Quando uma aplicação negocia com o sistema operacional uma determinada porta, o S.O. cria uma fila interna para armazenar as mensagens que chegam.

Porém isso não é suficiente: e se na máquina origem existir mais de um telnet com a mesma máquina destino? Solução: associar a cada cliente na máquina origem um número de porta único, obtido dinamicamente. Para isso em cada máquina existe um processo chamado portmap que "escolhe" uma porta disponível. Os números de porta até 1024 são reservados para portas pré definidas ( alguns valores são reservados para aplicações criadas por usuários ) e acima deste valor a alocação é dinâmica.

## 6.11. TELNET

Estabelece uma sessão de login remoto interativa

### **Utiliza porta 23 TCP**

O cliente ignora caracteres de controle com exceção de um que funcionará como escape  
Permite independência de tipos de terminais

**Desvantagem:** ineficiência. Cada caracter transmitido força várias trocas de contexto de processos dentro do sistema operacional local e remoto

**TN3270** - TELNET COMO EMULAÇÃO DE TERMINAL IBM 3270

## NOTAS

Protocolos confiáveis como TCP permitem que se faça uso interativo de máquinas remotas. A aplicação telnet faz com que usuários estabeleçam uma sessão de login com outra máquina que não aquela onde estão fisicamente conectados e tenham acesso a todos os comandos disponíveis na máquina remota, utilizando a porta de protocolo 23 para comunicação entre os processos. O processo cliente tem que ignorar o funcionamento de todos os caracteres de controle locais, como CTRL/C, CTRL/Z

## 6.12. FTP - File Transfer Protocol

- Filosofia Cliente-Servidor
- Acesso interativo
- Usuário especifica máquina destino, username e password
- Eficiente
- Transferência nem sempre pode ser revertida
- Duas portas de protocolo:
  - Processo de controle (baseado em telnet)
  - Processo de transferência de dados
- FTP anonymous para arquivos públicos

### NOTAS

Dentro da filosofia cliente-servidor, o usuário chama um programa cliente para iniciar a transferência. Nessa chamada o usuário especifica o computador remoto onde o arquivo desejado reside e uma autorização para obter acesso (username e password). O cliente então contata o servidor na máquina remota e requisita uma cópia do arquivo. Uma vez transferido, o usuário pode encerrar o software cliente.

Transferência entre máquinas heterogêneas é mais complicada, pois cliente e servidor têm que entrar em acordo com relação ao proprietário do arquivo, proteção de acesso e formato do dado. Porém, nem sempre assim a transmissão inversa é possível. Por exemplo, pode ser impossível converter o ponto flutuante de uma máquina para outra sem perder a precisão.

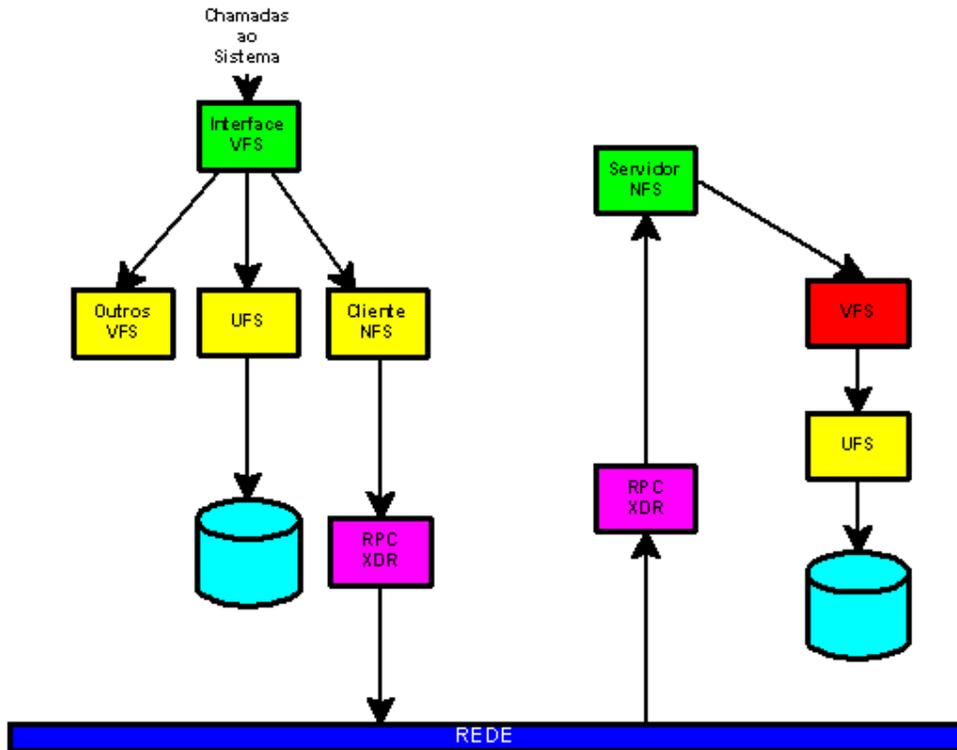
Como outros servidores, o servidor `in.ftpd` aceita várias conexões simultâneas. Um único processo servidor espera conexões e cria processos escravos para lidar com cada uma delas. Contudo o processo escravo não faz tudo sozinho. Ele apenas gerencia a parte do controle (porta 21) e usa um outro processo separado (porta 20) para trabalhar com a tarefa de transferência de dados. Para cada transferência o FTP estabelece uma nova conexão na porta 20. Quando a conexão de controle é encerrada, a sessão termina.

Para a parte de controle, o FTP usa o protocolo telnet simplificado, sem negociação de opções.

Para prover acesso irrestrito a arquivos públicos, muitas instalações usam FTP anonymous, o que significa que o usuário não precisa de conta ou password para ter acesso. O username usado é "anonymous" e a password é o endereço da pessoa (para fim de contabilidade). O acesso fica restrito a apenas um filesystem.

# **7. Network File System (NFS)**

## 7.1. Network File System



Serviço de rede que permite o compartilhamento transparente de sistemas de arquivos ou diretórios entre os nós de uma rede.

Implementado usando RPC (Remote Procedure Call), cujos protocolos são descritos usando XDR (External Data Representation)

## 7.2. Servidor NFS

- Exporta sistemas de arquivos para tornar seu uso transparente a aplicações de clientes NFS
- Lê ou grava arquivos em resposta a pedidos de clientes NFS
- Não mantém informação relativa a arquivos por clientes
- Pode servir a clientes NFS ou a outros servidores
- Não faz cache de pedidos de write de clientes NFS

### NOTAS

Arquivos de configuração:  
/etc/dfs/dfstab

```
#      place share(1M) commands here for automatic execution
#      on entering init state 3.
#
#      share [-F fstype] [ -o options] [-d "<text>"] <pathname> [resource]
#      .e.g,
#      share -F nfs -o rw=engineering -d "home dirs" /export/home2

share -F nfs -o ro /
share -F nfs -o ro /usr
share -F nfs -o ro /opt
share -F nfs -o ro /var
share -F nfs -o ro /usr/openwin
```

## 7.3. Cliente NFS

- Monta sistema de arquivos compartilhados pelo servidor NFS
- Arquivos são lidos ou gravados através de pedido ao servidor NFS
- Mantém toda informação a respeito de seus arquivos abertos
- Pode utilizar os serviços de vários servidores NFS
- Pode se comunicar com servidores NFS através de nós na Ethernet
- Realiza cache para gravação

### NOTAS:

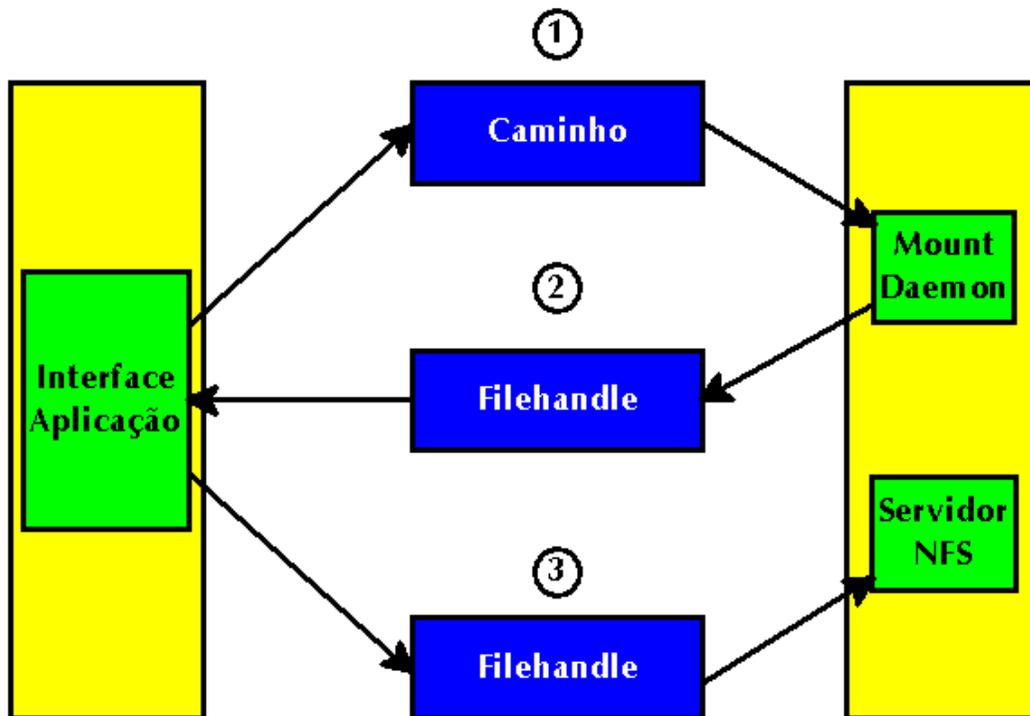
Arquivos:

/etc/vfstab

#device	device	mount	FS	fck	mount	mount
#to mount	to fsck	point	type	pass	at boot	option
#						
#/dev/dsk/cl1d0s2	/dev/rdisk/cl1d0s2	/usr	ufs	1	yes	-
fd -	/dev/fd	fd -	no	-		
/proc -	/proc	proc -	no	-		
/dev/dsk/c0t2d0s1	-	swap -	no	-		
/dev/dsk/c0t2d0s0	/dev/rdisk/c0t2d0s0	/	ufs	1	no	-
/dev/dsk/c0t2d0s6	/dev/rdisk/c0t2d0s6	/usr	ufs	1	no	-
/dev/dsk/c0t2d0s4	/dev/rdisk/c0t2d0s4	/var	ufs	1	no	-
/dev/dsk/c0t2d0s7	/dev/rdisk/c0t2d0s7	/export/	ufs	2	yes	-
/dev/dsk/c0t2d0s5	/dev/rdisk/c0t2d0s5	/opt	ufs	2	yes	-
/dev/dsk/c0t2d0s3	/dev/rdisk/c0t2d0s3	/usr/openwin			ufs	2
						yes
atlantic:/work	-	/atlantic	nfs	-	yes	-
swap -	/tmp	tmpfs -	yes	-		

## 7.4. Interação Cliente/Servidor

- No SunOS o arquivo `/etc/exports` determina os sistemas de arquivos a serem exportados, no Solaris é `/etc/dfs/dfstab`
- servidor inicializa o daemon do NFS (`in.nfsd`) e o daemon para mount (`mountd`)
- daemon `mountd` do servidor retorna um indicador (`file handle`) para o diretório ou sistema de arquivos solicitado
- `file handle` do cliente é colocado na tabela de mounts do kernel
- Todas as referências posteriores são passadas ao daemon `nfsd` rodando no servidor usando o `file handle` do cliente
- `read ahead/write behind` do NFS é gerenciado pelo `i/o` daemon



1. Cliente envia o caminho de um arquivo ou diretório ao servidor NFS
2. O servidor responde retornando o filehandle correspondente
3. O cliente envia o filehandle ao servidor NFS

### NOTAS:

Os arquivos de configuração no SunOS são respectivamente:  
`/etc/exports` no lugar de `/etc/dfs/dfstab`  
`/etc/fstab` no lugar de `/etc/vfstab`

## 7.5. Exportação de Sistemas de Arquivos

### Opções:

-ro                      exporta o sistema de arquivo como read-only  
-rw=host1:host2        read-only para a maioria e read-write para os hosts especificados

### NOTAS

No SunOS as definições dos arquivos a serem exportados são colocadas no arquivo `/etc/exports`. Durante a inicialização da máquina o script `rc.network` lê este arquivo e inicia os daemon `in.nfsd`.

No Solaris o processo de exportação é chamado `sharing`, dado pelo comando `share`. O arquivo `/etc/dfs/fstab` fica no lugar do `/etc/exports` do SunOS.

### Sintaxe do comando `share`:

```
share [-F nfs [-o opções] [-d descrição] <diretório>]
```

## 7.6. Importação de Sistema de Arquivos

O comando `mount` possui duas opções básicas que são `rw` (read/write) e `ro` (read only). A sintaxe das opções para `mount nfs` para sistemas Sun são:

```
Solaris      mount -F nfs [-o opções] HOSTNAME:DIRETÓRIO MOUNTPOINT
SunOS       mount -t nfs [-o opções] HOSTNAME:DIRETÓRIO MOUNTPOINT
```

### Opções:

```
ro      Somente Leitura
rw      Leitura e escrita
```

O arquivo `/etc/vfstab` contém uma lista dos sistemas de arquivos a serem montados durante o boot. No SunOS o arquivo que contém os diretórios a serem montados é o `/etc/fstab`.

## NOTAS

As opções de `mount` podem ser especificadas tanto durante a execução do comando `mount` ou no arquivo `/etc/vfstab`. As opções são usadas para especificar como um cliente `nfs` deseja que um sistema de arquivos seja montado. Quando o cliente `nfs` emite um pedido de `mount`, as condições sob as quais o sistema de arquivos ou o diretório é exportado pelo servidor `nfs` tem precedência sobre as opções de `mount` especificadas pelo cliente `nfs`.

O comando `mount` anexa um determinado sistema de arquivos à estrutura de diretórios. O diretório especificado no comando `mount` (mount point) deve existir previamente. Se o diretório possuir algum conteúdo antes do `mount`, este permanecerá oculto até que o sistema de arquivos seja novamente desmontado.

Para que um cliente `nfs` efetivamente monte um sistema de arquivos, o servidor `nfs` precisará estar exportando o mesmo. Se o servidor especificar quem pode importar ou montar o sistema de arquivos solicitado, o cliente precisa estar incluído nesta lista. O servidor `nfs` também precisa possuir os daemons necessários (na maior parte dos sistemas `rpc.mountd` e `nfsd`).

## 7.7. Comandos Informativos

```
# showmount atlantic  
wall.cimcorp.com.br  
vader.cimcorp.com.br  
ioda.telnet.com.br  
developer
```

```
#dfsmount atlantic  
export list for atlantic  
/export/home/coutinho ioda.telnet.com.br  
/pub/pub -ro Everyone
```

```
#dfshares obelix  
export list for atlantic  
/export/home/coutinho  
/pub/pub
```

### NOTAS

O comando `showmount` lista todos os clientes que montaram remotamente um sistema de arquivos na máquina especificada. A opção `-e` exibe a lista de sistemas de arquivos exportados, e a opção `-d` mostra os sistemas de arquivos montados por algum cliente.

No SunOS, no lugar dos comandos `dfsmounts` e `dhshares` utilizam-se as opções `-e` `-d` do `showmount`.

# 8. MAIL

## 8.1. Características do E-mail

### Porque usar e-mail?

- Mais rápido, barato e eficiente
- Mais fácil de armazenar

### Porque não usar e-mail

- Não é seguro nem confiável
- Conteúdo do e-mail pode ser ilegal

### Endereço Internet de e-mail

joao@cimcorp.com.br

## NOTAS

O correio eletrônico (e-mail) possui muitas vantagens sobre o correio tradicional. A chegada ao destino é quase imediata; o correio tradicional pode gastar dias ou mesmo semanas. O uso do correio eletrônico permite que a mensagem chegue ao usuário ou mesmo retorne a você muito rapidamente. Desde que não há uso no papel no processo, o envio e o recebimento de mensagens é muito mais rápido e fácil.

Todavia o correio eletrônico não é seguro. Qualquer pessoa com privilégios de root na máquina pode ler o mail de qualquer um, inclusive mails que não são dirigidos a usuários desta máquina (quando a máquina é um MX para outro domínio). Além disso, se a mensagem retornar por algum motivo, uma cópia dela será enviada ao postmaster (pessoa encarregada de verificar a causa de erros no mail e ajudar a localizar as pessoas). É ilegal também enviar via correio eletrônicos artigos sujeitos a taxaço. A transmissão de informações técnicas para fora de alguns países pode ser ilegal.

Quando se diz que o mail não é confiável, significa que não há certeza que a mensagem vai chegar às mãos da pessoa desejada, nem a que horas ou dia que vai chegar (que pode levar segundos ou dias) e nem há garantia que vai chegar antes de um mail enviado posteriormente.

## 8.2. Domínios

- gerenciamento da rede é mais fácil quando os domínios do mail e do DNS são os mesmos
- Precisam ser únicos se a rede possui contato com o mundo exterior
- É recomendável que várias máquinas sejam agrupadas sobre um mesmo domínio, de preferência o mesmo definido para o NIS

### Tipos de domínios Internet:

- com: comercial
- edu: educacional
- gov: governamental
- mil: instituições militares
- org: organizações sem fim lucrativo

## NOTAS

O agrupamento de máquinas em domínios permite que várias máquinas sejam administradas como uma única entidade. É interessante que as várias máquinas de um dado domínio administrativo se identifiquem através de um único domínio de mail e utilizem um único mail relay host. Como existem domínios para mail e NIS, normalmente o domínio para o mail é também usado pelo NIS. Esse grupo de máquinas sob o mesmo domínio devem montar o diretório onde ficam as mensagens incoming (var/spool/mail) do mail relay host deste domínio, para que os usuários as possam ler através de qualquer máquina desse domínio.

Não há necessariamente uma máquina com o nome de um dado domínio de mail. Muitas vezes usuários tentam testar conectividade mail. Muitas vezes usuários tentam testar conectividade de mail através do comando telnet ou ping, o que é um erro. Para um teste efetivo, deve-se procurar saber qual(is) máquinas responsáveis por um dado domínio. Isto é feito via utilitário nslookup com ``set type = MX".

Dentro de um domínio de mail, duas máquinas com o mesmo nome farão com que o mail seja direcionado erradamente. Fora do domínio entretanto o nome das máquinas pode ser duplicado.

## 8.3. Funcionamento do Mail

1. Mensagem é gerada por usuário ou programa usando qualquer interface de mail
2. A interface passa a mensagem para o processo chamado sendmail (porta 25)
3. Mensagem é armazenada em área de spool
4. Processo em background tenta periodicamente enviar a mensagem
5. É registrado um log da última tentativa e armazenado junto com a mensagem
6. Após tempo especificado pelo administrador, as mensagens ainda não enviadas são removidas da área de spool e um mail é enviado ao remetente.
7. Para identificar o destinatário da mensagem, o sendmail expande os endereços através dos aliases definidos e para cada domínio obtido que usa MX

### NOTAS

Normalmente, na maioria dos serviços da Internet, os protocolos enviam pacotes diretamente ao destino, usando timeout e retransmissão de seguimento de pacote caso não retorne nenhum sinal de acknowledgment. No caso do mail, no entanto, o sistema tem que prover uma solução para caso em que a máquina remota ou as conexões até ela falharem.

Para gerenciar esta atividade, sistemas de mail utilizam uma técnica chamada spooling. Quando o usuário envia uma mensagem de mail, o programa que faz a interface de mail com o usuário invoca o sendmail, que coloca uma cópia em uma área privativa de armazenamento. O sistema então inicia uma cópia numa área privativa de armazenamento. O sistema então inicia uma transferência para a máquina remota como uma atividade em background.

O processo em background se transforma em um cliente. Ele mapeia a máquina destino para um endereço IP(usando MX - Mail Exchange records e aliases) e tenta abrir uma conexão TCP com o servidor de mail da máquina destino. Se há sucesso, a mensagem é enviada e guardada numa área de spool do sistema remoto. Após servidor e cliente concordarem que a cópia foi aceita e armazenada, o cliente remove a cópia local. Se a conexão falha por algum motivo, a hora em que foi tentada a transmissão e a causa da falha são armazenadas junto com a mensagem.

Periodicamente essa mensagem vai ser examinada e nova tentativa de transmissão será feita em background. Se após determinado tempo, a mensagem não for entregue (ex:3 dias), a mensagem é retornada ao usuário e ao postmaster, e retirada da área de spool.

#### Obs:

1. MX são utilizados para associar a um dado endereço de mail, endereço de máquina da Internet, e a cada endereço uma prioridade.
2. A sintaxe dos endereços é definida na RFC822.

## 8.4. SMTP - Simple Mail Transfer Protocol

1. Define qual deve ser o formato do diálogo entre os processos sendmail cliente e servidor
2. Inicialmente é estabelecida a conexão, onde as máquinas se identificam
3. Em seguida, é comunicado o endereço do remetente e do(s) destinatário(s)
4. Após cada comando do transmissor o receptor manda um acknowledge
5. texto do mail é transmitido
6. Se há outros mails na fila para o mesmo destino, estes são enviados usando a mesma conexão
7. A conexão é encerrada

### NOTAS

A plataforma TCP/IP especifica um padrão para a troca de mails entre máquinas, chamado SMTP. O protocolo se limita a dizer como os processos de mail (servidor e cliente) se comunicam, isto é, como os processos sendmail devem interagir. Não tem nada a ver com a interface de mail do usuário, nem como o sendmail das máquinas foi configurado.

SMTP é simples. A comunicação entre o cliente e servidor consiste de texto legível. Inicialmente o cliente estabelece uma conexão com a porta de mail da máquina remota (25) e espera o servidor enviar uma mensagem:

```
220 READ FOR MAIL
```

Após isso o cliente envia o comando:

```
-HELLO
```

O servidor responde se identificando.

Uma vez estabelecida a comunicação, o cliente pode enviar uma ou mais mensagens de mail, terminar a conexão ou requisitar ao servidor para trocar os papéis de transmissor e receptor (comando TURN). Após o estabelecimento da conexão, a transação continua com o comando:

```
-MAIL From: <endereço_do_remetente>
```

O receptor responde:

```
-250 OK
```

Após isso são enviados uma série de comandos RCPT que especificam o destinatário da mensagem:

```
_RCPT to: <endereço_do_destinatário>
```

O receptor envia para cada RCPT aceito um acknowledge de sucesso (250) ou um comando de erro do tipo:

```
-550 No such user here
```

Encerrando a troca de endereços, o transmissor envia o comando

```
DATA.
```

O receptor responde com:

```
-354 Enter mail, end with "."
```

texto do mail é enviado e a conexão é fechada com o transmissor enviando para o comando QUIT.

## 8.5. Mail - Arquivos Necessários

**`/usr/bin/mail`**

**`/usr/lib/sendmail`**

**`/etc/sendmail.cf`**

**`/var/spool/mqueue`**

### NOTAS

Existem vários arquivos que precisam estar presentes para que o sistema de correio eletrônico funcione. São eles:

`/usr/bin/mail`

É o mailer original do V7. É usado como uma interface entre `/usr/ucb/mail` e `/usr/lib/sendmail`.

`/usr/lib/sendmail`

É o roteador real do mail. Ele recebe comandos do `/usr/bin/mail` e usa a informação encontrada no arquivo de configuração do `sendmail` para direcionar as mensagens.

`/etc/sendmail.cf`

Arquivo de configuração usado pelo `sendmail`. Define o domínio, o major relay host, opções de funcionamento do `sendmail` e regras de conversão de endereços.

`/var/spool/mail/user_id`

É a caixa de correio eletrônica. Todas as mensagens recebidas são armazenadas no arquivo.

`/var/spool/mqueue`

Mensagens a espera de serem enviadas são armazenadas aqui.

## 8.6. Mail - Outros Arquivos

**/etc/aliases**

**/home\_dir/.mailrc**

**/usr/ucb/newaliases**

**/usr/ucb/biff**

**/usr/etc/in.comsat**

**/etc/syslog.conf**

### NOTAS

**/etc/aliases**

Aliases que valem para todos os usuários. Aliases podem ser usados para agrupar usuários em listas, ou para retransmitir mail.

**/home\_dir/.mailrc**

Pode conter aliases que valem apenas para o usuário dono do diretório.

**/usr/ucb/newaliases**

É usado para iniciar e atualizar o banco de dados de aliases. É na verdade um link simbólico para o programa `/usr/lib/sendmail`.

**/usr/ucb/biff**

É usado para notificar o usuário da chegada de mail. A pessoa que enviou o mail, assunto e a primeira linha da mensagem aparecerão na console da estação de trabalho.

**/usr/etc/in.comsat**

É o daemon de notificação do mail. É utilizado para notificar ao usuário ou ao postmaster da ocorrência de mensagens impossíveis de se entregar.

**/etc/syslog.conf**

Quando o daemon `syslogd` é carregado, lê a configuração do arquivo `syslog.conf` para identificar quais os tipos de eventos devem ser "logados", a partir de que severidade, e quais os arquivos devem ser gerados.

## 8.7. Instalação do sendmail

- Obtenha um arquivo modelo que funcione e copie para o arquivo `/etc/sendmail.cf`
- Altere o arquivo de configuração `/etc/sendmail.cf`
- Encerre o daemon de sendmail e reinicialize-o

```
#ps -ef | grep sendmail
67    ? | 0:15 /etc/sendmail -bd -qlh
178  co S 0:00 grep sendmail
#kill -HUP 67
```

### Opções comuns do sendmail

```
sendmail -bd    (inicialização do daemon)
sendmail -bp    (imprime fila do mail)
sendmail -q     (limpa a fila do mail)
sendmail -v     (trabalha em modo verbose)
```

### NOTAS:

A primeira decisão a tomar é definir o domínio. Consulte o seu gerente de rede.

A segunda decisão é saber se sua máquina enviará as mensagens diretamente ao destino ou se usará um "relay host". Essa decisão é baseada no tamanho, autonomia e grau de experiência do administrador da sua máquina. É sempre mais simples enviar tudo para uma máquina que já possua um corpo administrativo mais equipado. Porém, o administrador da máquina escolhida tem que estar aberto para esse tipo de conexão.

O programa sendmail usa as informações contidas no arquivo `/etc/sendmail.cf` para a sua operação. As decisões tomadas acima terão que ser refletidas nesse arquivo.

A alteração do `sendmail.cf` requer que o daemon seja parado e reinicializado. As mudanças não entrarão em vigor até que isto seja feito.

Programa sendmail pode ser invocado pelo usuário para verificar a fila do mail, com explicação sobre cada mensagem até o momento, e também para forçar o sendmail a limpar a fila.

## 8.8. /etc/aliases

- Informa ao mailer a quem enviar as mensagens de erro.

```
Postmaster:root
```

- Redireciona mensagens para destinatários cujos nomes são normalmente escrito errado:

```
sherri:sherry
```

```
lori:laurie
```

```
rich:rich@geowell
```

- Pode também ser usado para facilitar o envio de mail a um grupo de usuários:

```
suporte:coutinho,tiagog,cleber
```

```
system: :include:/usr/system
```

- Envia mensagens que são processadas por programas:

```
hpc-l:"|catmail -L HPC-L -f"
```

### NOTAS

O arquivo `/etc/aliases` é usado para direcionar mail para usuários especiais, para facilitar a entrega para um grupo de usuários, ou para corrigir erros comuns de nomes escritos erradamente.

O arquivo `/etc/aliases` pode ser usado para enviar mensagens para um grupo de usuários. Os nomes devem constar todos em uma única linha ou referir a um arquivo que conterà lista dos nomes.

Um mailbox pode ser a entrada para um programa que processará o texto e os headers do mail.

# 9. IMPRESSORAS

## 9.1. CONEXÕES

Teste da conexão de uma impressora ASCII

```
# cat /etc/hosts > /dev/cua/a
```

Verificar conexões para impressoras Postscript

```
# tip hardware1
connected
executive
... <verbose message> ...
PS>
disconnecting
```

Para configurar impressoras no *Solaris 2* utiliza-se o programa `admintool`<sup>2</sup> no OpenWindows ou no CDE.

## NOTAS

A configurar uma impressora a primeira coisa a fazer é verificar a conexão. Para testar o cabo, tente enviar dados para a impressora ASCII, o comando `cat` é um bom teste. Use o comando `stty` para estabelecer o baud rate correto da impressora. Se nada é impresso, verifique se o cabo foi feito devidamente. Na maior parte dos casos o cabo deve ser do tipo null modem (DTE para DCE), mas as vezes a porta da impressora é configurada como uma DCE.

Para impressoras PostScript, o comando `tip` enviado para a porta estabelece uma conversação com a impressora. Por exemplo, `executive` obtém uma resposta da impressora. Se a impressora não responder aos comandos, o cabo pode ser problema. O comando `tip` pode ser utilizado também para testar o modem.

---

<sup>1</sup> Hardware pode ser `/dev/bpp0` (porta paralela) `/dev/ttya` (serial A) `/dev/ttyb` (serial B) ou outras.

<sup>2</sup> SunOS: Adicionalmente, deve-se conhecer a entrada adequada da impressora para o arquivo `/etc/printcap` e se a impressora vai precisar de um filtro de saída. A entrada do arquivo `printcap` deve ser fornecida pelo fabricante da impressora.

## 9.2. Comandos de Impressão

### Imprimir um arquivo na impressora ln03

```
% lp -dln03 file
```

### Verificar a fila de impressão na ln03

```
% lpq -Pln03
```

### Remover o job 10 da fila da ln03

```
% lprm -Pln03 10
```

### Programa de controle de impressoras

```
#lpc
```

```
lpc>
```

```
lpc>abort ln03 (abort current job on print ln03)
```

```
lpc>restart all (restart all printer daemons)
```

## NOTAS

O comando lpc é usado freqüentemente para inicializar uma impressora travada. Em tais casos tente primeiramente:

```
#lpc restart nome_da_impessora
```

### Se não funcionar tente então:

```
#lpc stop nome_da_impessora
```

```
#lprm primeiro_job_de_impressão
```

```
#lpc start nome_da_impessora
```

No SunOS o comando de impressão é o lpr.

## 9.3. DESTRAVANDO IMPRESSORAS

### Determinar PIDs do processo lpd

```
# ps -ef | grep lpd
87 ? S 0:02 /usr/lib/lpd
89 ? S 0:05 /usr/lib/lpd
14459 p6 S 0:00 grep
```

### Encerrar os processos

```
# kill 87 89
```

### Remover o lock da porta da impressora

```
# rm /var/spool/*/lock
```

### Reinicializar o daemon de impressão

```
#!/usr/lib/lpd
```

## NOTAS:

O daemon lpd é inicializado pelos scripts de inicialização durante o boot do sistema. Ao entrar no ar, o lpd cria o arquivo lpd.lock no diretório /var/spool.

O primeiro lpd é inicializado em modo "Master" e "Parent", e espera serviços através de comandos de impressão tais como lpr, lpc e lprm.

Quando o trabalho vem de uma impressora, o processo lpd pai cria uma cópia de si mesmo. Isto cria um arquivo de lock chamado /var/spool/nome\_da\_impresora para impedir que os processos lpd sejam criados para aquela impressora. Uma vez que a impressora tenha terminado seu trabalho, o processo lpd filho morre, mas outro pode ser criado se mais serviço aparecer.

Um sistema que possua muitas impressoras conectadas terá um processo lpd Master e um processo lpd filho para cada impressora ativa.

Uma impressora ativa é aquela que possui jobs sendo impressos. Estes processos filhos são criados e removidos pelo sistema para atender a carga de serviço.

# **10. BACKUP E RECUPERAÇÃO**

## 10.1. BACKUPS

Uma das tarefas mais importantes do administrador de sistemas

```
ufsdump / ufsrestore (dump / restore no SunOS)
tar
cpio
dd
```

### NOTAS:

Uma das tarefas mais importantes do administrador de sistemas, é garantir que os dados a ele confiados nunca serão perdidos. Acidentes freqüentemente acontecem e as conseqüências podem variar de um simples arquivos perdido a anos de trabalho jogado fora.

Existem várias utilitários no Unix que permitem que se façam cópias do sistema. A escolha de qualquer um deles vai depender do gosto pessoal de cada administrador.

Abordaremos aqui com mais detalhes os comandos `ufsdump` e `ufsrestore`, usados respectivamente para armazenar e restaurar arquivos.

O comando `ufsdump` permite que se façam backups completos (full) de sistemas de arquivos ou de diretórios. Permite também backup incrementais. Novamente, a decisão de quando se fazer backups completas e incrementais vai depender das particularidades de cada instalação e do administrador de sistemas.

## 10.2. TAR

- É o comando de cópia de diretórios mais utilizado no UNIX
- Cria arquivos de arquivos e possui múltiplos usos

### Sintaxe

```
tar [cxtvf] saída/entrada [diretório]
```

saída/entrada pode ser um device ou um arquivo

-c copia para a saída

-x extrai arquivos da entrada

-t lista a entrada

-v mostra o comando funcionando

-f opção para identificar o dispositivo de i/o

### Exemplos

```
tar cvf /dev/rmt/0 (copia diretório corrente para fita)
```

```
tar tvf backup.tar (lista o conteúdo de backup.tar)
```

## 10.3. CPIO

- Menos utilizados, também permite criar e manipular arquivos de arquivos

### Sintaxe

`cpio` `-[ioc]` `[><]` saída/entrada  
saída/entrada pode ser um device ou um arquivo  
`c` define o nome da saída  
`-i` define o nome da entrada  
`-c` cria um cabeçalho portavel entre sistemas

### Exemplos

```
ls | cpio -oc > /dev/rfd0 (copia diretório corrente para disquete)
cat jtr2.cpio | cpio -ic (extrai arquivos de jtr2.cpio)
cpio -ic < /dev/rfd0 (lê backup em disquete no formato cpio)
```

## 10.4. DD

Trabalha com arquivos, mas é mais usado para trabalhar com backup de partições, disquetes, fitas e conversões

### Sintaxe

```
dd [operador=valor] [operador=valor] ...
```

### Principais operadores

if=file (entrada)

of=file (saída)

bs=tamanho (configura o tamanho do bloco)

conv=opção (faz conversões)

### Exemplos

```
dd if=boot.flp of=/dev/rfd0 (copia imagem para flop)
```

```
dd if=/dev/rmt/0 of=/dev/rmt/1 (duplica fita)
```

```
dd if=teste.txt of=TESTE.TXT conv=ucase
```

## 10.5. PREPARATIVOS PARA O BACKUP

- Antes de iniciar um backup certifique-se que não existem usuários no sistema
- Rode o programa fsck para corrigir inconsistências no sistema de arquivos
- Se existirem usuários logados use os comandos wall, rwall e shutdown para anunciar o encerramento do sistema
- sistema deve estar em modo monousuário durante backups de nível zero
- Não faça backup das áreas de swap

### NOTAS:

O programa fsck verifica a integridade do sistema de arquivos e faz correções aonde possível. Este programa deve também ser rodado sempre que se fizer um restore de um sistema de arquivo completo.

O programa who informa se existem usuários logados no sistema. Se um usuário estiver logado com um longo tempo de ociosidade, provavelmente esqueceu de dar logoff.

Para as paradas programadas aconselha-se avisar o usuário por meio do arquivo /etc/motd (message of the day). A mensagem contida neste arquivo é lida durante o processo de login. O programa wall (warn all) serve para divulgar uma mensagem aos usuários logados no sistema. O programa rwall permite propagar uma mensagem aos clientes de um servidor.

O programa shutdown faz o encerramento do sistema em um tempo predeterminado. Mensagens são enviadas periodicamente aos usuários logados e o intervalo entre as mensagens vai diminuindo à medida que se aproxima o horário de encerramento.

É recomendado que os backups de nível zero sejam realizados em modo monousuário de modo a assegurar o mínimo de inconsistência possível entre o backup e o sistema de arquivos.

Aconselha-se pela mesma razão, que os backups incrementais sejam sempre realizados quando a taxa de utilização do sistema for mínima.

## 10.6. BACKUP DE SISTEMAS DE ARQUIVOS

- A unidade de fita será reconhecida no momento de instalação do Solaris 2.x ou podemos forçar sua detecção com o comando `tapes` ou através de um `boot -r`
- comando padrão usado para backup no Solaris 2.x é o UFS DUMP

### Sintaxe

```
ufsdump [opções] [saída] [sistema_de-arquivos]
```

### Principais opções:

0-9 nível de backup

u atualiza arquivo de log (/etc/dumpdates)

f identifica argumento de saída

### Exemplos

Ufsdump full da partição /home:

```
#ufsdump 0uf /dev/rmt/0 /home
```

Ufsdump em uma fita remota

```
#ufsdump 0uf vader:/dev/rmt/0 /home
```

## NOTAS

O comando `ufsdump` permite a cópia de sistemas de arquivos inteiros.

O nível do dump varia de 0 a 9. O nível 0 é o mais baixo e copia integralmente o sistema de arquivos e o nível 9 é o mais alto. Um dump incremental salva todos os arquivos modificados desde a data do último dump de nível mais baixo.

Um backup de nível 0 (full) precisa ser feito antes que se façam dumps incrementais.

### As opções aceitas pelo comando `dump` são:

**0-9** Este número especifica o nível do dump.

**f** Especifica o dispositivo onde será feita a gravação do dump.

**u** Atualiza o arquivo /etc/dumpdates

**c** Especifica unidade de cartucho.

**b** Blocagem a ser utilizada. O valor default é 20 blocos para unidade de fitas de 1/2" 2 126 para unidades de fita de 1/4". A blocagem default para gravação em densidade de 6250 BPI é de 64 blocos. Quanto maior o fator de blocagem mais rápido será o dump. O tamanho do bloco para fitas é de 512 bytes.

**s** Especifica o comprimento da fita. O default para fitas de 1/2" é de 2300 pés, 700 pés para fitas cartucho de 150M, 6000 pés para fita de 8mm (2 Gbytes) e 13000 pés para fita de 8mm (5 Gbytes).

**D** Especifica a densidade de gravação. O default para fitas de 1/2" é de 1600 bpi, para fitas de 1/4" é de 1000, para fitas de 2 e 5 Gbytes é de 54000 bpi.

Para fazer dumps de sistemas de arquivos em unidades de fita remotas, certifique-se que o arquivo `/.hosts` da máquina remota contém uma entrada com o nome de sua máquina.

## 10.7. BACKUP DE VÁRIOS SISTEMAS DE ARQUIVOS EM UMA FITA

Utilize a opção `norewind`

```
#ufsdump 0uf /dev/rmt/0n /usr
```

Faça o dump seguinte:

```
#ufsdump 0uf /dev/rmt/0 /home
```

### NOTAS

Se for necessário que se faça o dump de mais de um sistema de arquivos utilizando a mesma fita, utilizar a opção `n` (`norewind`) na definição do dispositivo da fita.

Utilizando o device `/dev/null` nenhum dump será realizado, mas pode-se conseguir uma estimativa de consumo de fitas.

O comando `df` também pode ser utilizado para a mesma finalidade.

A opção `norewind` suprime o `rewind` após o encerramento do dump da partição. Assim, o dump da próxima partição será feito a partir do ponto da fita onde termina o primeiro backup.

Entretanto, se o segundo dump não utilizar a opção `norewind`, a fita será rebobinada.

É necessário um comando separado para cada partição.

A opção `u` faz com que o programa `dump` insira um registro no arquivo `/etc/dumpdates` especificando a partição que foi copiada, o nível e a hora do dump.

```
cat /etc/dumpdates
/dev/dsk/c0t0d0s0 1 WedMay 22 04:33:01 1997
/dev/dsk/c0t0d0s3 1 WedMay 22 04:33:01 1997
/dev/dsk/c0t0d0s7 1 Fri Apr 15 18:50:32 1997
```

## 10.8 ESTRATÉGIAS PARA DUMP

### Nível 0 uma vez por mês

SEG	TER	QUA	QUI	SEX
5	5	5	5	3
5	5	5	5	3
5	5	5	5	3
5	5	5	5	3

SEG	TER	QUA	QUI	SEX
3	4	5	6	2
3	4	5	6	2
3	4	5	6	2
3	4	5	6	2

### NOTAS

As duas estratégias mostram a flexibilidade oferecida pelos dumps incrementais. No primeiro exemplo cada dump de nível 5 aumenta de tamanho à medida que se aproxima a sexta feira, quando um backup nível 3 é realizado. No segundo exemplo cada dump incremental é aproximadamente do mesmo tamanho visto que apenas as alterações de um dia são salvas.

O melhor é planejar um modo de se fazer os dumps que utilizem o menor número de fitas possível, certificando-se que todos os arquivos estão em segurança.

## 10.9. RECUPERAÇÃO DE BACKUP

O comando para recuperação de backups feitos com o `ufsdump` é o `ufsrestore`

### Sintaxe

```
ufsrestore [opções] [arquivo_de_backup]
```

Principais opções: `t,i,r,x,v,f`

### Recuperação de backup

```
#ufsrestore xvf /dev/rmt/0 arquivo(s)
Specify next volume#:n
Set owner/mode for `.' [y/n]y
```

### Restore através da rede

```
ufsrestore vader:/dev/rmt/0 arquivo(s)
```

### Restore em fita com múltiplos backup

```
ufsrestore xvs /dev/rmt/0 2
```

## NOTAS

Um usuário pode perder um arquivo, ou sistema de arquivos inteiros devido a falta de cuidado, um programa mal escrito ou com erros, ou, através do uso impróprio de utilitários. Neste caso o administrador de sistemas deve ser capaz de reparar o erro e restaurar os arquivos ou os sistemas de arquivos perdidos. Isto pressupõe a existência de um backup recente em fita magnética. Envolve também descobrir em que fita se encontram os arquivos desejados. A opção do comando `restore` fornece uma listagem de todos os arquivos contidos em uma fita de dump.

### As opções aceitas pelo restore são as seguintes:

- t Lista o conteúdo do arquivo de dump.
- i Invoca o programa `restore` em modo interativo.
- r Restore recursivo. Todos os arquivos na fita são restaurados.
- x restaura apenas os arquivos especificados na linha de comando
- v Roda em modo verboso, exibindo o nome de todos os arquivos restaurados.
- f Especifica o dispositivo de fita magnética. Obrigatório o uso quando usado o comando `restore`.

A organização de fitas de backup é muito importante. Nem sempre a versão mais recente de um arquivo é a requerida. Todas as fitas de dump devem estar numeradas e datadas. O `restore` deve ser feito no mesmo diretório onde se encontrava o arquivo digital. Você pode escolher entre criar um novo diretório ou fazer o `restore` no arquivo `/tmp`. Após os arquivos tiverem sido restaurados copie-os para o lugar correto. Certifique-se também que o modo do arquivo esteja correto (ownership/permissions).

Para restaurar um arquivo de uma fita que se encontra montada em um sistema remoto certifique-se que existe em seu sistema, no arquivo `/.hosts`, uma entrada com o nome do sistema remoto. Isto porque o programa `restore` é rodado na área `root`.

## 10.10. UFSRESTORE INTERATIVO

```
#cd /tmp
#ufsrestore ivf /dev/rmt/0
restore>ls
restore>cd /usr
restore>add arquivo
restore>extract
restore>delete arq
restore>verbose
restore>quit
#
```

## 10.11. UFSRESTORE DE UM SISTEMA DE ARQUIVOS

### **Prepare a partição rodando newfs:**

```
#newfs /dev/rdisk/c0t0d0s7
```

### **Verifique integridade da partição**

```
#fsck /dev/rdisk/c0t0d0s7
```

### **Monte a partição em uma área temporária**

```
#mount /dev/dsk/c0t0d0s7 /home
```

### **Restore o sistema de arquivos**

```
#cd /home  
#ufsrestore rvf /dev/rmt/0  
#rm restoresymtable
```

### **Verifique a integridade do sistema de arquivos**

```
#fsck /dev/rdisk/c0t0d0s7  
#
```

## NOTAS

Certas situações por vezes exigem que um sistema de arquivos seja restaurado em sua integridade. O disco pode ter se danificado a tal ponto que a sua troca seja requerido, a instalação de novos programas pode requerer o redimensionamento do disco, etc.

Lembre-se que no Solares 2.X os nomes dos devices são diferentes do SunOS.

## 11.12. UFSRESTORE DO /ROOT

### **Carregue o sistema a partir do CDROM**

```
Ok boot cdrom -s
```

### **Se houver problemas com o disco, rode o programa format para analisar e corrigir o problema**

```
fsck -y /dev/rdisk/c0t3d0s0
```

### **Utilize o comando newfs para recriar o sistema de arquivos /root**

```
#newfs /dev/dsk/c0t3d0s0
```

### **Verifique o sistema de arquivos criado**

```
#fsck /dev/dsk/c0t3d0s0
```

### **Monte o sistema para fazer o restore**

```
#mount -F ufs -o rw /dev/dsk/c0t3d0s0 /a  
#cd /a  
#ufsrestore rf /dev/rmt/0
```

## 11.13. UFSRESTORE DO /ROOT (Cont.)

### **Remova o arquivo restoresymtable após o final do restore**

```
#rm restoresymtable
```

### **Desmonte o sistema de arquivos**

```
#cd /  
#umount /a
```

### **Instalar o bloco de boot após o ufsrestore do sistema de arquivos root**

```
#cd /usr/lib/fs/ufs/'uname -u'  
#installboot bootblk /dev/rdisk/c0t3d0s0
```

### **Verifique o sistema de arquivos criado**

```
#fsck /dev/rdisk/c0t3d0s0
```

### **Reiniciar o sistema**

```
#reboot
```

# **11. MANUTENÇÃO DE SISTEMA DE ARQUIVOS**

# 11.1. MANUTENÇÃO DE SISTEMA DE ARQUIVOS

- Os sistemas de arquivos podem ver a se danificar devido a:
- Parar o sistema sem sincronizar os discos
- Não verificar e reparar inconsistências nos sistemas de arquivos durante o startup
- Erros no disco
- Falhas de hardware

## 11.2. FSCK

O programa fsck checa:

### Inconsistências nas informações do superbloco

1. Tamanho do sistema de arquivos
2. Números de inodes
3. Contagem de blocos livres
4. Contagem de inodes livres

### Inconsistências no mapa de blocos do grupo de cilindros

1. Blocos livres requisitados por arquivos
2. Total livres+blocos requisitados=total de blocos

### Inconsistências nos inodes

1. Um inode está alocado ou livre, nunca ambos
2. Contagem de links correta
3. Inodes não referenciados

## NOTAS

Inconsistências lógicas são quase sempre descobertas pelo programa fsck ao checar a integridade dos sistemas de arquivos, e são normalmente ocasionadas por quedas do sistema operacional. O programa fsck pode efetuar os reparos tanto interativamente quanto automaticamente (através da opção -p).

O programa fsck checa os sistemas de arquivos identificados no arquivo `/etc/vfstab` como do tipo ufs. O arquivo `/etc/vfstab` pode ser usado para descobrir a ordem em que os sistemas de arquivos são examinados pelo programa fsck.

O programa fsck se baseia no fato de que não devem haver inconsistências nos sistemas de arquivos. Por exemplo, o número de inodes é especificado no superbloco. Qualquer informação que contrarie este número é considerado suspeita. O programa fsck tenta então consertar o sistema de arquivos baseado no tipo de problema encontrado.

Arquivos ou diretórios órfãos são reconectados aos sistemas de arquivos sob o diretório `lost+found`. Cada sistema de arquivos possui um diretório com esse nome.

O programa fsck verifica as seguintes inconsistências:

- Blocos requisitados por mais de um inode, ou requisitado por um inode na lista de inodes livres.
- Blocos requisitados por um inode ou lista de blocos livres fora dos limites do sistema de arquivos.
- Contagem de links incorreta.
- Tamanho de diretórios incorretos.
- Informação sobre inodes incorreta.
- Blocos não identificados em lugar nenhum
- Arquivo apontando para um inode livre, ou um número de inode fora do limite.
- Verificação dos superblocos: mais blocos por inodes do que existem no sistema de arquivos.
- Formato da lista de blocos livres incorreto.
- Total de blocos livres ou inodes livres incorretos.

## 11.3. FSCK (CONT)

O programa fsck verifica, mas não faz correções em um sistema de arquivos ativo

Para verificar a integridade de todos os sistemas de arquivos listados no arquivo /etc/vfstab :

```
#fsck
```

### Exemplos:

Checar o sistema de arquivos /dev/dsk/c0t0d0s0

```
#fsck dev/dsk/c0t0d0s0
```

Checar o sistema de arquivos na partição dev/dsk/c0t0d0s0 usando o superbloco alternado 32

```
#fsck -b 32 dev/dsk/c0t0d0s0
```

Listar todos os superblocos alternados para a partição /dev/rdisk/c0t3d0s0

```
#newfs -N /dev/rdisk/c0t3d0s0
```

## NOTAS

O programa fsck invocado sem nenhum argumento checa todas as partições contidas no arquivo fstab. O programa fsck aceita mount points como argumento e determina através daí o dispositivo correto a checar. A verificação do dispositivo ``raw" é sempre mais rápida.

Se for utilizada opção -y, todas as correções são feitas sem confirmação. Esta opção não é aconselhável, se todos os dados da partição forem importantes. O programa fsck faz as modificações diretamente no disco.

O programa fsck pode ser instruído a utilizar um superbloco alternado. Para determinar os superblocos existentes use o comando newfs com a opção -N <número>. (Muito cuidado, pois o comando newfs sem opção -N apagará todos os dados).

## 11.4. FASES DO FSCK

### Exemplo:

```

** Phase 1 - Check Blocs and Sizes
3788627 BAD I=66
** Phase 2 - Check pathnames
DUP/BAD I=66 OWNER=root MODE 100755
SIZE=76409 MTIME=Mar 3 15:29 1995
FILE=/usr/bin/kawan
REMOVE?y
** Phase 3 - Check connectivity
** Phase 4 - Check Reference Counts
BAD/DUP I=66 OWNER=root MODE=100755
SIZE=76409 MTIME=Mar 3 15:29 1995
FILE=/usr/bin/kawan
CLEAR?y
UNREF FILE I=531 OWNER=root MODE=100600
SIZE=0 MTIME=Feb 28 11:35 1995
RECONNECTY?y
** Phase 5 - Check Free List
137 BLK(S) MISSING
BAD FREE LIST
SALVAGE?y
** Phase 6 - Salvage Free List
1398 files blocks 6546 free

```

## NOTAS

**Fase 1:** Checa Blocos e Tamanhos (cheça inodes procurando inconsistências)

**Fase 2:** Checa Path-Names (cheça diretórios e inconsistências de inodes)

**Fase 3:** Checa Conectividade (cheça que todos diretórios estão conectados no sistema de arquivo)

**Fase 4:** cheça Reference Counts (compara informações de contagem de link da fase 2 e 3, corrigindo discrepância)

**Fase 5:** Cheça grupos de cilindros (cheça blocos livres e o mapa de inodes usados procurando por inconsistências)

No exemplo ao lado, o programa fsck descobriu um inode duplicado para um arquivo chamado /usr/bin/kawan. Neste caso, o superusuário decidiu desconectar o inode e instruiu o programa fsck a prosseguir e limpa-lo. Ao final, o programa descobriu que existem blocos faltando na lista de blocos livres. Ele pergunta então se a lista de blocos livres deve ser salva.

Com a resposta afirmativa o programa fsck prossegue e salva a lista.

Normalmente o programa fsck é executado em cinco fases, mas neste caso existe uma fase adicional.

Algumas vezes, após o fim da execução do programa fsck aparece uma mensagem do tipo:

```
**** BOOT UNIX (NO SYNC) ****
```

Isto acontece quando o sistema de arquivos montado foi modificado. Se o sistema não é reinicializado neste ponto, a modificações feitas pelo programa fsck serão perdidas se os dados no superbloco forem gravados novamente no disco.

Interrompa (STOP-A) e reinicialize o sistema imediatamente. Não espere que o sistema se atualize sozinho.

## 11.5. MONITORANDO ESPAÇO EM DISCO

#df

O comando df (disk free) mostra o tamanho total (não incluindo os 10% adicionais), número de kbytes usados e disponíveis, porcentagem total usada, e mount point de cada sistema de arquivos.

#du

O comando du (disk usage) mostra o número de blocos contidos em cada subdiretório do diretório corrente.

```
#du -a | sort -nr
```

Lista todos os arquivos neste diretório em ordem de tamanho do maior para o menor incluindo o caminho completo e o tamanho em kbytes

### FIND

```
#find / -name '*core' -print
```

Exibe todos arquivos cujos nomes terminem em core

```
#find / -name '*core' -exec rm '{}'\;
```

Remove todos os arquivos cujos nomes terminam em "core"

```
#find / -mtime +90 -print
```

Exibe todos os arquivos não modificados por noventa dias.

```
#find / -name `#*` -mtime +7 -print
```

Exibe todos os arquivos iniciados por "#" cuja data de última modificação exceda 7 dias

## 11.6. /usr/etc/cron e /var/spool/cron/crontabs

```
0,30 * * * * /bin/date > /dev/console
0 0 * * * calender -
15 0 * * * /usr/etc/sa -s > /dev/null
0,30 * * * * /etc/dmesg ->> /usr/adm/messages
```

### NOTAS

O daemon `/usr/etc/cron` é usado para executar comandos em horários pré determinados. Ele lê arquivos contidos no diretório `/var/spool/cron/crontabs`. Estes arquivos pertencem a usuários do sistema; os arquivos determinam quais programas ou scripts serão executados e quando.

Os registros nestes arquivos contêm cinco campos. São eles:

<b>Campo</b>	<b>Valores permitidos</b>
minuto	0-59
hora	0-23
día do mês	1-31
mês	1-12 (Janeiro - Dezembro)
día da semana	1-7 (Segunda - Domingo)

#### Comando

`n` O comando é executado quando o valor do campo é `n`

`n,p,q` O comando é executado quando o valor do campo é `n`, `p`, `q`.

`n-q` O comando é executado quando o valor do campo é `n-p`

`*` O comando é executado para todos os valores possíveis do campo

## 11.7. /usr/bin/crontab e /var/spool/cron

`/var/spool/cron/cron.allow`

Se esse arquivo existir, apenas os usuários nele contidos são autorizados

`/var/spool/cron/cron.deny`

Se o arquivo `cron.allow` não existir, os usuários contidos no arquivo `cron.deny` não são autorizados a executar o comando `crontab`

`#crontab -e`

Edita o arquivo `crontab` do usuário

`#crontab -l`

Lista o arquivo `crontab` do usuário

`#crontab nome_do_arquivo`

Edita o arquivo `crontab` especificado

## 11.8. AT

```
%at 23:30
at>make
at>^D
%atq
Rank Execution Date Owner Jobs# JobName
1st Dec 22, 1992 01:00 aldo 2350 stdin
%atrm 2350
2350 removed
```

## 11.9. USO DE QUOTAS DE DISCO

### Adicionar opção de quota para o sistema de arquivo

```
../c0t0d0s7 /home ufs 1 yes rquota
```

### Criar o arquivo quotas no diretório pai da partição que irá trabalhar com quotas

```
#touch /home/quotas
```

### Configurar quotas para cada usuário

```
#edquota marcelo  
fs /home blocks (soft=2500,hard=4500)  
inodes (soft=80, hard=1250)
```

### Ligar a verificação de quota

```
#quotaon -av
```

### Verificação da quota de disco de um usuário

```
#quota -v marcelo
```

# 12. SEGURANÇA

## 12.1. INTRODUÇÃO

- Sistema operacional Unix não foi projetado com preocupações relativas à segurança
- Sistemas instalados com pouco ou nenhum mecanismo de segurança
- unix foi um sistema desenvolvido por programadores para programadores
- A conexão em rede agravou o problema
- A difusão do uso de sistemas Unix está tornando estes sistemas mais seguros
- As características básicas de segurança do UNIX estão ligadas às contas de usuários, aos sistemas de arquivos e à conectividade.

## 12.2. CONTAS DOS USUÁRIOS

### Os três componentes de uma conta de usuário

- /etc/passwd e /etc/shadow
- diretório home
- arquivos de configuração
  
- .cshrc
- .login
- .profile

Grupos de usuário (/etc/group)

Umask

#umask 022 (resulta arquivos regulares com 644)

## 12.3. Criando conta de usuário

No Solaris 2.x as contas devem ser abertas preferencialmente através do programa de administração (admintool)

**AdminTool: Add User**

**USER IDENTITY**

User Name:

User ID:

Primary Group:

Secondary Groups:

Comment:

Login Shell:  /bin/sh

**ACCOUNT SECURITY**

Password:

Min Change:  days

Max Change:  days

Max Inactive:  days

Expiration Date:    (rdmmmyy)

Warning:  days

**HOME DIRECTORY**

Create Home Dir:

Path:

## NOTAS

Além da abertura de contas o admintool permite modificar vários outros mapas importantes na administração:

- usuários
- grupos
- hosts
- impressoras
- portas serias
- software

No Solaris 2.x as contas de usuários podem ser abertas de duas outras forma elo comandos useradd

## 12.4. Segurança das Contas

### Senhas

- Seleção adequada
- Política para escolha
- Verificação periódica da segurança
- Data de expiração

**Negar contas guest**

**Contas sem senha**

**Contas de grupo**

**Usar preferencialmente NIS+**

## NOTAS

A segurança de sistemas Unix pode ser dividida em três áreas principais. A segurança de contas consiste basicamente impedir que usuários não autorizados acessem o sistema; segurança de rede consiste em não permitir que os pacotes sejam capturados por equipamentos ou programas instalados sem autorização e que os dados importantes trafeguem criptografados; a segurança dos sistema de arquivos, consiste em impedir que usuários não autorizados, tanto do sistema, quanto invasores, consigam acesso aos dados armazenados no sistema.

### Contas:

Uma das maneiras mais fáceis de um intruso ganhar acesso a um sistema, é descobrindo a senha da conta de algum usuário. Isto não é difícil de ocorrer já que muitas instalações não removem o acesso de pessoas que já deixaram a organização, não verificam a segurança das contas em vigor, com passwords fáceis de serem descobertas.

## 12.5. Segurança do Sistema de Arquivos

### Permissões dos arquivos

Cada diretório ou arquivo possui três grupos de permissões:

- dono do arquivo
- grupo do dono do arquivo
- demais usuários

Cada grupo de permissões pode especificar os seguintes atributos:

```
read
write
execute
setuid (4xxx)
setgid (2xxx)
sticky (1xxx)
```

## NOTAS

A última linha de defesa contra invasores são as permissões oferecidas pelo sistema. Cada arquivo ou diretório possui três grupos de bits de permissão a ele associados: um grupo para o usuário ao qual o arquivo pertence, um grupo para os usuários do grupo ao qual o arquivo está associado, e um grupo para todos os demais usuários. Cada grupo contém três grupos de bits que controlam:

### Acesso de leitura (read):

Se este bit estiver ligado, o arquivo ou diretório possui acesso de leitura. Em se tratando de um diretório, esta permissão permite a um usuário ver o conteúdo de um diretório mas não acessá-lo.

### Acesso de gravação (write):

Se este bit estiver ligado, o arquivo ou diretório possui acesso de leitura. Em se tratando de um diretório, esta permissão de escrita implica a capacidade de criar, apagar, ou renomear arquivos. Note que a permissão para remover um arquivo não é determinada pelas permissões do arquivo, mas sim pelas permissões do diretório que o contém.

### Acesso de execução (execute):

Se este bit estiver ligado, o arquivo ou diretório pode ser executado (pesquisado). No caso de um diretório, a permissão de execução implica que os arquivos nele contidos podem ser acessados. Além destas permissões, existe um quarto bit que se ligado atribui as seguintes propriedades aos arquivos e diretórios:

#### setuid:

Este bit, se ativado no grupo de permissões do dono do arquivo, indica que todos os que executarem este programa o estarão fazendo com os privilégios do proprietário do arquivo. Por exemplo, o programa sendmail é setuid root., o que lhe permite gravar mensagens do sistema, o que é vedado a usuários normais. Este bit não possui significado em arquivos não executáveis.

#### setgid:

Este bit atua da mesma forma que o setuid, com a diferença que o programa será executado com as permissões do grupo ao qual pertence.

#### stcky:

O sticky informa ao sistema operacional atuar diferentemente com relação à imagem executável do arquivo. É remanescente de versões antigas do Unix e possui pouco hoje. Diretórios que possuem modo 777 e têm este bit ligado não podem ser removidos (/tmp por exemplo).

## 12.6. Segurança do Sistema de Arquivos

- Nunca utilizar ou permitir que se utilize em qualquer sistema Unix shell scripts com o `stuid` bit ligado
- Sticky bit em diretórios
- `setgid` em diretórios
- `umask`
- Arquivos encriptados
- Dispositivo

### NOTAS

Shell scripts que possuem os bits `setuid` ligados não são seguros, não importa quantas precauções tenham sido tomadas ao escrevê-los. Tais scripts nunca devem ser permitidos em qualquer sistema Unix.

Nas versões mais recentes de sistema Unix (incluindo o Solaris) foi acrescentado um novo significado ao sticky bit quando aplicado à diretórios. Neste caso, os usuários não podem apagar e renomear arquivos pertencentes a outros usuários. Isto é tipicamente útil no caso diretórios com `/tmp`. Normalmente o diretório `/tmp` possui acesso universal para gravação, permitindo que qualquer usuário remova arquivos pertencentes a outro usuário. Ativando o sticky bit no diretório `/tmp`, os usuários podem remover apenas seus próprios arquivos. Para ativar o sticky bit em um diretório, use o comando: `#chmod o+t diretório`

Ao se criar um arquivo normalmente todas as permissões são ativadas. Como isto raramente é o desejado, o valor do `umask` é usado para modificar o grupo de permissões com as quais um arquivo é criado. Ou seja, da mesma forma com o comando `chmod` especifica quais bits devem ser ligados, o comando `umask` especifica quais bits devem ser desligados.

O comando `umask` é especificado nos arquivos `.cshrc` ou `.profile`, dependendo da shell utilizada. A conta `root` deve ter a `umask` definida como `022`, para impedir a criação acidental de arquivos pertencentes ao superusuário com permissão `777`.

A segurança de dispositivo é uma questão importante em sistemas Unix. Arquivos de dispositivo são utilizados por vários programas para acessar dados nos dispositivos ou na memória. Se estes dispositivos não estão devidamente protegidos, o sistema está vulnerável a ataques. A lista completa de dispositivos é muito grande e varia de sistema para sistema. Em linhas gerais, as seguintes normas se aplicam:

Os dispositivos de disco, tais como `/dev/dsk/c0t3d0s0`, etc., devem pertencer ao usuário `root` e `group operator`, e devem possuir modo `640`.

Como muito poucas exceções, todos os outros dispositivos devem pertencer ao usuário `root`. Uma destas exceções são os terminais, que pertencem ao usuário que o estiver utilizado no momento. Ao desconectar-se, o terminal volta a pertencer ao `root`.

## 12.7. Segurança da Rede

### Trusted hosts

- `/etc/hosts.equiv`
- `.rhosts`

### Terminais seguros

### Network File System (NFS)

- `/etc/dfs/dfstab`
- Restrição de acesso para o superusuário (root)

File Transfer Protocol (FTP) e Trivial FTP (tftp) são serviços potencialmente perigosos.

## 12.8. Segurança da Rede

### **Mail**

- Cuidado na criação de alias que enviem mensagens para programas
- Verificar que o programa sendmail não autoriza o comando debug

### **Cuidado na instalação de programas públicos**

- Finger
- Modems e servidores de terminais (ttys)
- Firewalls

## 12.9. Monitoração da Segurança

### Segurança de contas

- lastlog
- utmp
- wtmp
- acct

### Segurança da rede

- syslog
- showmount

## NOTAS:

Uma das tarefas do administrador de sistemas é a monitoração da segurança. Esta tarefa envolve o exame de arquivos de log para detectar acessos não autorizados, bem como a monitoração de falhas de segurança.

As contas devem ser monitoradas periodicamente de modo a verificar dois eventos: usuários que logam quando não devem (por exemplo, tarde da noite ou quando estão de férias) e usuários executando comandos que normalmente não deveriam usar.

O arquivo `/usr/adm/lastlog` registra o login mais recente a cada usuário do sistema. A mensagem impressa no terminal a cada vez que um usuário loga usa a data armazenada no arquivo lastlog: Last login: sat Mar 10 10:50:48 from turing.unicamp.br

A data do último login relatada pelo comando `finger` também usa estes dados. Os usuários devem ser alertados a inspeccionar esta data para certificarem-se de que não foi efetuado nenhum acesso não autorizado às contas e, caso positivo, a alertar o administrador de sistemas para o ocorrido.

O arquivo `/etc/utmp` é usado para registrar quem está logado no sistema no momento. Este arquivo pode ser exibido através do comando `who`.

O arquivo `/usr/adm/wtmp` registra cada as datas de login e logout de cada usuário. Este arquivo também pode ser examinado através do comando `who` (`who /usr/adm/wtmp`).

O arquivo `wtmp` pode também ser examinado manualmente através do comando `last`. Este comando ordena as entradas no arquivo, casando os tempos de login e logout. Se invocado sem argumentos, o comando `last` exhibe toda a informação contida no arquivo.

O arquivo `acct` registra a execução de cada comando no sistema, quem o executou e quanto tempo gastou. O arquivo `acct` pode ser examinado através do comando `lastcomm`.

O `syslog` é um mecanismo que permite que qualquer comando registre mensagens de erro e informativas na console do sistema e/ou em um arquivo. Normalmente mensagens de erro são gravadas no arquivo `/usr/adm/messages` juntamente com a data e hora em foram enviadas pelo programa que as gerou.

## 12.10. Monitoração da Segurança

### Segurança do sistema de arquivos

- find
- arquivos setuid/setgid
- arquivos sem dono
- .rhosts

checklists

backups

### NOTAS

Verificar falhas de segurança no sistema de arquivos é outra tarefa importante do administrador. Primeiramente devem ser identificados os arquivos que podem ser alterados por usuários não autorizados, arquivos que podem involuntariamente dar permissões excessivas a usuários e arquivos que podem involuntariamente dar permissões excessivas a usuários e arquivos que possam fornecer acesso a invasores. É importante também monitorar modificações no sistema de arquivos e possuir mecanismos que permitam a volta do sistema ao estado original.

O comando `find` é um comando de propósito geral para pesquisar o sistema de arquivos. A programação do comando `find` listada a seguir localiza todos os arquivos do sistema com os bits `setuid` e `setgid` ligados. A saída deste comando deve ser analisada para determinar se não existe algum arquivo suspeito na lista.

```
#find / -type f -a \( -perm 0400 -o -perm 0200 \) -print
```

Usando as opções que se seguem, o comando `find` identifica os arquivos com permissão de escrita universal.

```
#find / -perm -2 -print
```

Para identificar arquivos que não pertencem a nenhum usuário ou nenhum grupo:

```
#find / -nouser -o nogroup -print
```

Imediatamente após a instalação de um sistema, deve-se gerar um arquivo que liste a configuração inicial dos arquivos do sistema:

```
#ls -aslgR /bin /etc /usr > Masterchecklist
```

Este arquivo contém uma lista completa de todos os arquivos nestes diretórios. As linhas referentes a arquivos que mudem freqüentemente devem ser removidas do arquivo. O `Masterchecklist` deve ser guardado em um local seguro para evitar adulterações. Para pesquisar alterações no sistema de arquivos, execute o comando acima e compare com o arquivo mestre:

```
#diff Masterchecklist Currentlist
```

Outro aspecto muito importante é a realização de backups freqüentes do sistema de arquivo. Backups não apenas protegem contra falhas de hardware como contra deleções acidentais.

## **12.11. Comandos que podem auxiliar na segurança**

Além dos programas de monitoração existem comandos simples do Unix que podem ser úteis na detecção de violações de segurança. O administrador deve se familiarizar com os comandos executados normalmente no sistema de forma a ser capaz de identificar suspeitas

- `ps`
- `who`
- `w`
- `ls`

## 12.12. Ferramentas Úteis

As ferramentas abaixo estão disponíveis na Internet e podem ser de grande ajuda na melhoria da segurança de um sistema Unix/Solaris

- npasswd (passwords mais seguras)
- crack (verifica passwords)
- cops (identifica riscos em UNIX)
- tripwire (monitora mudanças em binários)
- tcpwapper (controla acesso a rede)
- kerberos (sistema de identificação)

## 12.13. Leitura Recomendada

- Manuais originais de cada Sistema Operacional e de cada equipamento
- Sun System Administration
- Evi Nemeth e Garth Snider e Scott Seebass, "Unix System Administration Handbook", Prentice Hall, 1989.
- David A. Curry "Improving the security of your System"
- Janice Winsor "System Administrator's Guide", MacMillan Technical Publishing, 1998
- Janice Winsor "Advanced System Administrator's Guide", MacMillan Technical Publishing, 1998