SUMÁRIO

SMIT

Inicialização
Encerramento

Segurança

Gerenciamento de Usuários

Gerenciamento de Grupos

Volumes Lógicos

Sistemas de Arquivos

PAGINAÇÃO
MEMÓRIA VIRTUAL

BACKUP RESTORE

Ambiente Operacional

Instalação

Processos

Subsistemas

Contabilização de Uso de Recursos

1

SMIT

SMIT

- **Ferramenta de gerenciamento do sistema**
- **Fácil de usar**
- **Poderosa**
- **Cria registros das sessões de trabalho (arquivos smit.log e smit.script)**

NOTAS:

O **SMIT** é a principal ferramenta do administrador de sistemas **AIX** para o gerenciamento do sistema. Esta ferramenta apresenta uma interface orientada a tarefas para executar os diversos comandos necessários à administração do sistema. O usuário é conduzido passo a passo pelos diversos menus até chegar à tarefa desejada. O usuário não precisa memorizar comandos complexos e as suas (em alguns casos) dezenas de opções. Além disto tudo, o **SMIT** cria dois arquivos de vital importância: o arquivo `smit.log` e o arquivo `smit.script`. O arquivo `smit.log` contém um registro da sessão, ou seja, são registrados todos as telas pelas quais se navegou, todos os comandos emitidos e finalmente, todos os resultados obtidos. O arquivo `smit.script` contém apenas os comandos que foram emitidos. Este arquivo contém exemplos valiosos de comandos e suas sintaxes. Um outro uso possível do arquivo `smit.script` é a duplicação da configuração de uma máquina (usuários, endereços de rede, definição de impressoras, etc,). Para fazer isto basta digitar:

```
ksh smit.script
```

Desta forma seriam executados todos os comandos empregados na configuração do sistema antigo. É evidente que para que isto funcione é necessário que o administrador ou administradores, usem sempre o SMIT para qualquer alteração nos parâmetros do sistema operacional.

SMIT
Interface ASCII

```
                System Management

Move cursor to desired item and press
Enter

Installation and Maintenance
Devices
Physical and Logical Storage
Security&Users
Diskless Workstation Management
Communications Applications & Serv-
ices
Spooler (Print Jobs)
Problem Determination
Performance & Resource Scheduling
System Environments
Processes and Subsystems
Applications
Using SMIT (information only)

F1=Help  F2=Refresh  F3=Cancel
 F8=Image
F9=Shell F10=Exit    Enter=Do
```

NOTAS:

Uso

Para invocar o programa SMIT digite:

- **smit**
- **smit *FastPath***

Para sair:

- **F10**
- **ESC+0**

NOTAS:

Existem várias maneiras de se invocar o programa smit. Se se estiver usando um terminal do tipo **hft**, pode-se invocar a interface **Motif** (default) ou a interface **ASCII**. Prevalece aí a preferência de cada um. Se todavia, se estiver usando terminal do tipo vt100 ou compatível, IBM3151, ou outros baseados em caracteres, a interface default, como é óbvio, é a interface **ASCII**.
Para invocar o smit digite na linha de comandos:

```
smit
```

e será apresentado então o menu principal do programa smit. O programa smit aceita também argumentos que o direcionam para um submenu (FastPath). Por exemplo, para criar a definição de uma impressora digite:

```
smit mkprt
```

Neste caso é apresentado o submenu `Add a Printer,` que guia o usuário através dos passos necessários para criar a definição de uma impressora.

Para sair do smit basta digitar, a partir de qualquer tela, **F10** ou **ESC+0.** A ação a ser tomada vai depender do tipo de terminal (existência de suporte a teclas de função).

Uso

Para ativar a interface ascii digite:

smit -C

ou

smitty

NOTAS:

Para invocar o programa smit usando a interface ASCII, digite:

```
smit -C
```

ou

```
smitty
```

Ambas as opções suportam argumentos *(FastPath)*

Teclas de Função

F1	**ESC+1**	**Help**
F2	**ESC+2**	**Refresh**
F3	**ESC+3**	**Cancel**
F4	**ESC+4**	**List**
F5	**ESC+5**	**Undo**
F6	**ESC+6**	**Command**
F7	**ESC+7**	**Edit**
F8	**ESC+8**	**Image**
F9	**ESC+9**	**Shell**
F10	**ESC+10**	**Exit**
ENTER	**ENTER**	**Do**

NOTAS:

As funções podem ser invocadas pressionando-se ou as teclas de função ou a tecla ESC seguida de um número indicativo da função.

F1: Fornece mais informações sobre o tópico onde o cursor se encontra

F2: Redesenha a tela, eliminando caracteres estranhos à aplicação. Por exemplo, mensagens da console podem ser direcionadas para a tela. A tecla F2 elimina a mensagem e apresenta uma tela limpa.

F3: Retorna à tela anterior. Quando pressionada a partir do menu principal esta tecla encerra a sessão

F4: Exibe uma lista de opções disponíveis para o campo sobre o qual se encontra o cursor.

F5: Exibe o conteúdo original do campo

F6: Exibe o comando que o SMIT está montando

F7: Apresenta o texto ressaltado em um campo para edição. Faz seleções individuais em listas. Por exemplo, esta tecla permite a seleção de programas a serem instalados no menu de instalação de programas

F8: Exibe o parâmetro a ser usado com a opção *FastPath* para o menu ou tela corrente. Salva também uma imagem da tela no arquivo smit.log para impressão posterior.

F9 Entra em uma shell. É solicitada uma confirmação para execução do comando.

ENTER Executa o comando construído pelo SMIT ou salva as entradas selecionadas em uma tela

Simbologia

Símbolo	Significado
*	Preenchimento obrigatório. Aparece à esquerda do campo do nome ou do prompt
#	Campo de preenchimento numérico
X	Campo deve ser preencido com número hexadecimal
/	Forneça o nome de um arquivo
++	Uma lista de opções está disponível. Para exibir as opções, pressione a tecla F4 ou <ESC>+0
[ ]	Delimitação de campos
<	Mais texto à esquerda do campo visível
>	Mais texto à direita do campo visível

NOTAS:

2

Inicialização
Encerramento

O processo de Boot

Durante o boot o sistema testa o hardware, carrega e executa o sistema operacional e configura os periféricos. Para realizar o boot são necessários os seguintes recursos:

- **Imagem de boot**
- **Acesso aos filesystems root e /usr**

NOTAS:

Tipos de Boot de Sistema

- **Disco rígido**
- **Standalone**
- **Via rede**

NOTAS:

Uma máquina é inicializada para operações normais com a chave na posição *NORMAL*. A chave é posicionada na posição *SERVICE* para se rodar diagnósticos.

O boot standalone pode ser feito a partir de disquetes, fitas ou CDROM com a chave da máquina na posição *SERVICE*. No modo de manutenção o administrador pode instalar ou atualizar novos produtos e rodar diagnósticos.

O boot via rede é iniciado remotamente através da rede. A máquina deve estar com sua chave na posição *NORMAL*. Um ou mais servidores de arquivos podem fornecer os arquivos necessários para o boot.

Operações efetuadas durante o boot

- **Configuração dos dispositivos**
- **Inicialização do software básico**
- **Montagem e disponibilização dos sistemas de arquivos**

NOTAS:

Durante o boot efetuado a partir do disco rígido, a imagem de boot é encontrada no disco local criado quando da instalação do sistema operacional. Durante o processo de boot, o sistema configura todos os dispositivos da máquina e inicializa o software básico necessário para a operação normal do sistema. No final do processo os sistemas de arquivos são montados e disponibilizados para uso.

O mesmo se aplica a clientes diskless. Estas máquinas também necessitam de uma imagem de boot e acesso aos arquivos do sistema operacional. Clientes diskless não possuem um disco local e obtêm todos os arquivos necessários através da rede.

Boot a partir de fita

- Certifique-se de que o sistema está desligado
- Coloque a chave na posição `secure`, ligue a estação e espere que apareça o código `200` no indicador do LED.
- Insira a fita na unidade
- Vire a chave para a posição `service`
- Ao aparecer o código `c31` escolha a console de acordo com as instruções da tela
- No menu `Installation and Maintenance` selecione a opção 4.
- No prompt do sistema digite `getrootfs` para que seja exibida a lista de discos rígidos do sistema
- Para acessar o sistema de arquivos digite `getrootfs hdiskxx`

NOTAS:

System run levels

- **Estes níveis identificam o estado do sistema e definem quais processos são inicializados**
- **Ao final do boot do sistema, o *run level* ¢ lido a partir da entrada initdefault do arquivo `/etc/inittab`**
- **O *system run level* pode ser alterado com o comando `telinit`**
- **O arquivo `/etc/inittab` cont¢m um registro para cada processo que define em quais nÁveis do sistema este processo rodarÀ**
- **Durante o boot do sistema o comando `init` l¨ o arquivo `/etc/inittab` para determinar quais processos serÐo inicializados**
- **Para identificar em que nível o sistema está rodando examine o arquivo `/etc/.init.state`**

NOTAS:

Ao final da inicialização do sistema, o comando `init` assume o *run level* definido na entrada `initdefault` do arquivo `/etc/inittab`. O sistema opera neste *run level* at¢ que receba um sinal para alterÀ-lo.

São os seguintes os *run level*s definidos:

0-9	**Quando o comando `init`** muda para *run level*s entre 0 e 9, ele mata todos os processos do *run level* corrente e reinicializa os processos associados com o novo *run level*
0-1	**Reservados para uso futuro**
2	*run level* default
3-9	**Podem ser definidos de acordo com as preferências dos usuários**
a,b,c	**Quando o comando `init`** solicita uma mudan§a para os *run levels* **a, b ou c, ele não mata os processos do** *run level* corrente; ele simplesmente inicializa os processos associados com os novos *run levels*
S,s,M,m	**Solicita ao comando `init` que coloque o sistema em modo de manutenção**
N	**Envia um sinal que impede que processos sejam** *respawned*
Q,q	**Solicita ao comando `init`** que releia o arquivo **`/etc/inittab`**

Encerramento do Sistema

- **shutdown**

  shutdown -m +2

- **halt**

- **fasthalt**

NOTAS:

O comando halt escreve os dados no disco rígido e para o processador. A máquina não reboota. Este comando deve ser executado apenas pelo usuário root e nunca deve ser usado quando outros usuários estiverem logados. Use este comando quando a máquina tiver que ser desligada para algum tipo de manutenção. O comando halt registra o horário de shutdown usando o comando syslogd e coloca um registro no arquivo /usr/adm/wtmp.

O comando shutdown encerra o sistema operacional de várias formas. Durante a execução do shutdown os usuários são notificados através do comando wall do horário em que o sistema será encerrado através de uma mensagem do próprio sistema e de uma mensagem opcionalmente fornecida por quem invocou o comando. Após o horário especificado o comando shutdown invoca o comando killall para encerrar todos os processos remanescentes, roda o comando sync para limpar todos os blocos de disco residentes na memória e finalmente desmonta todos os sistemas de arquivos.
Opções:

-c	**Não verifica os sistemas de arquivo durante reinicialização do sistema**
-d	**Traz o sistema para do modo distribuído para o modo multiusuário**
-F	**Fast shutdown**
-h	**Encerra o sistema completamente, semlhante ao halt**
-k	**Não executa o shutdown**
-m	**Traz o sistema para o modo de manutenção**
-r	**Reinicializa o sistema após o encerramento**

3

Segurança

Introdução

- Sistemas Unix não foram projetados com preocupações com segurança
- Sistemas instalados com poucos ou nenhum mecanismos de segurança
- Sistema desenvolvido por programadores para programadores
- Conexão em redes agravou o problema
- A difusão maior do uso de sistemas Unix está tornando os sistemas mais seguros

NOTAS:

Violações de Segurança

- ***Internet Worm***
- **Espionagem**
- ***Christmas Virus***

NOTAS:

Objetivos de um Sistema de Segurança

- **Integridade da informação**
- **Privacidade da informação**
- **Disponibilidade da informação**

NOTAS:

Este item descreve os objetivos da segurança de computadores e os mecanismos empregados para a consecução dos mesmos. São abordadas também as ameaças à segurança e como estas podem servir como base para definição da política de segurança de sistemas.

OBJETIVOS:

A segurança de sistemas computacionais é muito semelhante a outros tipos de segurança. O seu objetivo é a proteção de informações armazenadas nos computadores. A segurança da informação objetiva:

- Manter a integridade da informação:
  O valor de qualquer informação depende de sua acuracidade. Se mudanças não autorizadas são efetuadas, esta informação perde algum ou todo o seu valor.
- Privacidade das informações:
  O valor de muitas informações é dependente de seu sigilo.
- Disponibilidade da informação:
  A informação deve sempre estar imediatamente disponível a quem dela necessita.

Mecanismos de segurança

- **Controle de acesso**
- **Auditoria**

NOTAS:

Diferentes mecanismos são usados para se proteger as informações, tais como:

- Mecanismos de controle de acesso:
  Impedem a violação da segurança. Estes mecanismos protegem a informação permitindo ou negando o acesso a recursos baseado em dados como por exemplo a identidade do usuário.
- Mecanismos de auditoria:
  Identificam violações de segurança. A informação é protegida pelo registro de acesso a recursos e outros eventos relevantes de forma a que todos as ações de usuários no sistema sejam auditáveis.

Tanto o controle de acesso quanto a auditoria dependem de uma administração e autenticação de usuários segura. A autenticação de usuários consiste em se verificar a identidade do usuário quando este loga no sistema e na configuração do sistema operacional, dispositivos e usuários de modo a que o sistema possa ser operado de maneira confiável.

Controle de Acesso

- **Físico**
- **Discrecionário**

NOTAS:

Existem dois tipos de controle de acesso:

- Controle de acesso físico:
  Provê uma forma física de controle de acesso. O computador está fisicamente protegido por, por exemplo, uma sala fechada.
- Controle de acesso discrecionário:
  Forma mais comum de controle de acesso. Com esta forma de controle de acesso, o proprietário dos dados pode garantir a terceiros o direito de ler ou alterar os seus dados. Cada um destes usuários por sua vez pode extender a outros estes direitos.

Auditoria

- **Auditoria física**
- **Auditoria de sistemas**

NOTAS:

Existem dois tipos de auditoria:

- Auditoria física:
  Semelhante ao controle de acesso físico. O computador está protegido fisicamente, por exemplo, em uma sala trancada. Qualquer tentativa de se invadir a sala do computador resulta em danos irreparáveis. Estes danos acionam a segurança do sistema. Como a identidade do invasor é desconhecida, a auditoria física não fornece dados completos.
- Auditoria de sistemas:
  Cria um registro de todos os eventos relevantes no tocante à segurança do sistema. Estes eventos incluem acesso à informação, login de usuários e acessos administrativos. Cada evento inclui a identidade do usuário que a causou, de forma a possibilitar que o auditor de sistemas identifique sinais potenciais ou reais de violação da segurança.

Autenticação de Usuários

- A identidade de um usuário pode ser estabelecida através de:
- Informações que somente o usuário possui, como por exemplo, senhas e informações pessoais
- Algo que apenas o usuário possua, como por exemplo, chaves ou cartões magnéticos

NOTAS:

Administração Segura de Sistemas

- **O administrador de um sistema seguro deve:**
- **Definir as características de segurança do sistema, dispositivos, etc.**
- **Definir as características de segurança de seus usuários**

NOTAS:

O administrador de um sistema seguro deve definir as características de segurança do sistema, dispositivos de entrada e saída e dispositivos de armazenamento disponíveis no sistema. Isto inclui permissão de execução de programas e permissão de leitura e gravação de arquivos de configuração.

O administrador deve também definir as características de segurança de seus usuários. Isto inclui os direitos de acesso e privilégios no sistema, como cada usuário deve ser auditado, e como a autenticação de cada um deles deve ser efetuada.

Ameaças à Segurança

Ameaças à segurança da informação derivam de três tipos diferentes de comportamento:

- **Descuido**
- **Má fé**
- **Ataques deliberados**

NOTAS:

Três tipos distintos de comportamento podem comprometer os dados de um sistema computacional:

- Descuido:
  A segurança é freqüentemente comprometida devido ao descuido de usuários autorizados ao sistema. Se um usuário é descuidado com sua senha de acesso do sistema, qualquer outra medida de segurança é ineficaz para impedir o acesso a dados confidenciais.
- Má fé:
  Muitos problemas de segurança são causados por usuários autorizados que exploram o sistema à procura de informações desprotegidas.
- Ataques deliberados:
  Um indivíduo que tenha em mente a penetração em um sistema irá estudá-lo para detectar pontos fracos em sua segurança e planejar ataques deliberados com o propósito de explorar estas falhas.

Este último tipo de comportamento representa a maior ameaça à segurança das informações, mas problemas causados pelos dois primeiros itens não devem ser subestimados.

Administração de Segurança

É dever do administrador configurar os seguintes aspectos de segurança:

- **Controle de acesso**
- **Procedimentos de identificação e autenticação**
- **Trusted Computing Base (TCB)**
- **Auditoria de usuários**

NOTAS:

A boa administração de um sistema é vital para a manutenção da segurança de recursos de informação em um sistema computacional. A versão 3 do AIX é baseada em conceitos que permitem estabelecer e manter controles de acesso adequados juntamente com ferramentas de auditoria. É responsabilidade do administrador configurar os seguintes aspectos de segurança:

- Controle de acesso
- Procedimentos de controle e autenticação
- Trusted Computing Base (TCB)
  Parte do sistema responsável por assegurar as politícas de segurança de informação do sistema. Todo o hardware do sistema está incluído no TCB, mas o administrador do sistema deve se preocupar principalmente com os componentes de software do TCB. O TCB compreende:
  - O kernel (sistema operacional)
  - Arquivos de configuração que controlam a operação do sistema
  - Qualquer programa que rode com privilégios de alterar o kernel ou os arquivos de configuração

  O administrador de sistemas pode definir quais programas devem pertencer ao TCB e deve ser cuidadoso em acrescentar apenas programas plenamente confiáveis, ou seja, programas que foram totalmente testados, que tenham seu código fonte examinado ou cuja fonte seja confiável.

  O administrador de sistemdireitos de acesso
- Auditoria de usuários

Administração de Segurança

Uma vez configurado o sistema o administrador será capaz de:

- **Controlar o acesso a recursos de informação, terminais e dispositivos de entrada e saída**
- **Configurar senhas de acesso**
- **Instalar e configurar o TCB e pacotes de software**
- **Auditar ações de usuários**

NOTAS:

Administração de Usuários

- **Criação**
- **Definição de atributos**

NOTAS:

A administração de usuários consiste na criação de usuários e grupos, na definição de seus atributos e como serão autenticados. Os usuários são os agentes primários de um sistema. Os seus atributos controlam seus direitos de acesso, ambiente, como são autenticados, e como/quando/onde suas contas podem ser acessadas.

Os grupos são uma unidade de acesso discrecionário (DAC - *Discrecionary Access Control*). Os grupos possuem uma identificação e são compostos de membros e administradores. O criador de um grupo é normalmente o primeiro administrador.

O sistema operacional suporta os atributos padrão dos usuários normalmente encontrados nos arquivos `/etc/passwd` e `/etc/group`, como por exemplo:

- Informações de autenticação como a senha
- Credenciais:
  Especificam o identificador do usuário, grupo principal, e identificação do grupo secundário
- Ambiente:
  Especifica o diretório home e o ambiente de shell

O sistema operacional permite um maior controle, se desejado, com atributos estendidos. As informações de segurança podem também ser mantidas fora do acesso público.

Alguns usuários e grupos podem ser definidos como administrativos. Estes usuários e grupos podem ser criados e modificados apenas pelo usuário *root*.

Controle de Contas

A cada conta é associado um conjunto de atributos. Estes atributos são criados a partir de valores default quando se define uma conta e podem ser alterados quando se desejar

NOTAS:

Cada usuário de um sistema computacional possui uma conta, à qual são associados um conjunto de atributos. Estes atributos são criados a partir de valores default no momento da definição de uma conta através do comando mkuser. Os atributos de uma conta podem ser alterados através do comando chuser.

O conjunto completo de atributos está definido nos arquivos
`/usr/lib/security/mkuser.default`
`/etc/security/user`
`/etc/security/limits`

Estes defaults podem ser alterados diretamente nos arquivos acima. A maior parte dos defaults estão definidos de modo a refletirem o uso padrão.
OBS.: A atribuição de um valor igual a zero no arquivo `/etc/security/limits` implica que o atributo em questão não possui limites.

Identificação e Autenticação

- **Definem a identidade do usuário**
- **Usuários identificados basicamente por uma conta e uma senha**
- **Senhas são armazenadas separadamente das definições das contas**
- **Senha de acesso: todo cuidado é pouco**

NOTAS:

A identificação e autenticação estabelecem a identidade de um usuário. Para acessar o sistema o usuário deve fornecer o nome de sua conta (identificação) e sua senha de acesso (autenticação). Em um sistema seguro todas as contas devem ter senhas ou estarem invalidadas. Se a senha estiver correta, o usuário ganha acesso à sua conta adquirindo os acessos e privilégios a ela inerentes.

Todas as contas destinadas a uso normal devem possuir senhas. Uma pessoa que conheça a senha de uma conta pode logar naquela conta usufruindo de todos os direitos e privilégios da mesma. Os grupos também podem ter senhas, e um usuário que não seja definido em determinado grupo ainda assim pode usufruir dos privilégios deste se conhecer sua senha.

Como a senha é a única proteção de cada conta, é de vital importância que os usuários sejam cuidadosos na seleção e guarda de suas senhas. A maior parte das tentativas de invasão de sistemas começa por tentativas de se adivinhar as senhas. A versão 3 do AIX provê um nível adicional de proteção armazenando as senhas separadamente da definição das contas e dos grupos. Somente o usuário root deverá ter acesso às senhas encriptadas e a outros dados de segurança armazenados nos arquivos `/etc/security/passwd` e `/etc/security/group`. Com este acesso restrito às senhas, um invasor não poderá decifrar as senhas com um programa que tente passar por todas as senhas possíveis e prováveis.

É ainda possível adivinhar as senhas tentando logar em uma conta por meio de várias tentativas. Se a senha é trivial, tais ataques podem ser bem sucedidos. Conseqüentemente, é importante educar seus usuários sobre a forma correta de se escolher suas senhas.

As senhas a seguir são obviamente triviais e podem ser adivinhadas sem maiores esforços:

- Palavras que podem ser encontradas em um dicionário
- Senhas que incluam nomes ou sobrenomes do usuário ou de seus parentes próximos
- Senhas que incluam o mês corrente do ano

- Senhas baseadas em algum atributo do usuário

Segurança das Contas

- **Senhas**
  - **Seleção adequada**
  - **Políticas para escolha**
  - **Verificação periódica da segurança**
- **Datas de expiração**
- **Contas** *guest*
- **Contas sem senha**
- **Contas de grupo**
- **NIS**

NOTAS:

A segurança de sistemas Unix pode ser dividida em três áreas principais. Segurança de contas consiste basicamente em 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 dados importantes trafeguem criptografados; a segurança dos sistemas de arquivos, consiste em impedir que usuários não autorizados, tanto do sistema ou invasores, consigam acesso a 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 normalmente é fácil visto que muitas instalações não removem o acesso de pessoas que já deixaram a organização, não verificam a segurança de contas em vigor, com senhas fáceis de serem adivinhadas

Restrições às Senhas

- **Podem ser aplicadas às senhas as seguintes restrições:**
- **Número mínimo de semanas entre a troca de senhas**
- **Número máximo de semanas sem troca de senhas**
- **Número mínimo de caracteres alfabéticos que a senha deve conter**
- **Número mínimo de caracteres não alfabéticos que a nova senha deve conter**
- **Número máximo de vezes que um caracter pode aparecer na nova senha**
- **Número mínimo de caracteres na nova senha que devem ser diferentes dos caracteres na senha antiga**

NOTAS:

O gerenciamento adequado das senhas somente pode ser conseguido através da educação dos usuários. A versão 3 do AIX, entretanto, fornece algumas facilidades que podem ajudar, ou mesmo forçar, a escolha de senhas não triviais. Estas restrições se encontram no arquivo `/etc/security/login.cfg`, e são exercidas sempre que uma nova seja é definida para um usuário ou grupo. Observe que todas as restrições se aplicam ao sistema e não a usuários individuais.

Use o seu bom senso na definição destas regras de modo a impedir que um usuário tenha que escolher senhas tão complicadas que impeçam a memorização, fazendo com que tenham que ser escritas, comprometendo assim a segurança do sistema. Cumpre lembrar que a segurança das senhas é sempre de responsabilidade do usuário. Restrições simples, associadas com orientação e auditorias periódicas é a melhor política.

Restrições às senhas em `/etc/security/login.cfg`

minage	Número mínimo de semanas que devem se passar antes que seja permitida a troca de senhas
maxage	Número máximo de semanas que podem se passar antes que a troca de senhas seja solicitada pelo sistema
minalpha	Número mínimo de caracteres alfabéticos que uma nova senha deve conter
minother	Número mínimo de caracteres não alfabéticos que cada senha deve conter. Obs: O tamanho mínimo de uma senha no sistema é equivalente à soma dos campos minalpha e minother. O tamanho máximo da senha é de 8 caracteres
maxrepeats	Número máximo de vezes que um caracter pode aparecer em uma nova senha
mindiff	Número mínimo de caracteres na nova senha que precisam ser diferentes de caracteres na senha antiga

NOTAS:

A tabela abaixo fornece os valores recomendados, default e máximos para cada uma das restrições que podem ser aplicadas às senhas. Os valores default são aplicados se a stanza `pw_restrictions` não estiver definida no arquivo `/etc/security/login.cfg`, ou se um atributo particular não estiver definido. Ambientes seguros devem usar os valores recomendados, e mesmo sistemas apenas usados para trabalhos pessoais devem exigir senhas de algum tipo, especialmente se estiverem ligados a algum tipo de rede.

Parâmetro	Sugerido	Default	Máximo
minage	**1**	**0**	**52**
maxage	**8**	**0**	**52**
minalpha	**5**	**0**	**8**
minother	**2**	**0**	**8***
mindiff	**3**	**0**	**8**
maxrepeats	**1**	**0**	**8**

* O valor máximo de minother é (8 - minalpha)

Segurança da Rede

Trusted hosts

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

Terminais seguros

***Network File System* (NFS)**

- `/etc/exports`
- `/etc/netgroup`
- **Restrição de acesso para o superusuário (*root*)**

***File Transfer Protocol* (*FTP*)**

- ***tftp***

NOTAS:

Segurança da Rede

- **Mail**
- **Cuidado na instalação de programas públicos**
- **Finger**
- **Modems e servidores de terminais**
- **Firewalls**

NOTAS:

Segurança do Sistema de Arquivos

Permissões dos arquivos

- Cada diretório ou arquivo possui três grupos de permissões:
  - dono do arquivo
  - grupo a que pertence o dono do arquivo
  - demais usuários
- Cada grupo de permissões pode especificar os seguintes atributos:
  - ***read***
  - ***write***
  - ***execute***
  - ***setuid***
  - ***setgid***
  - ***sticky***

NOTAS:

A última linha de defesa contra invasores do sistema 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 idênticos 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 lhe permite acessá-lo.
- Acesso de gravação (*write*):
  Se este bit estiver ligado, o arquivo ou diretório pode ser modificado. Em se tratando de um diretório, a 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 arquivos na fila de 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 bit, com a difer-

ença que o programa será executado com as permissões do grupo ao qual pertence

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

Segurança do Sistema de Arquivos

- **Nunca utilizar ou permitir que se utilize em qualquer sistema Unix shell scripts com o *setuid bit* ligado**
- ***sticky* bit em diretÂrios**
- **Setgid bit em diretórios**
- **umask**
- **Arquivos encriptados**
- **Dispositivos**

NOTAS:

Shell scripts que possuem os bits setuid ou setgid 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 sistemas Unix foi acrescentado um novo significado ao *sticky* bit quando aplicado a diretÂrios. Quando este bit ¢ ativado em um diretÂrio, usuÀrios nÐo podem apagar ou renomear arquivos pertencentes a outros usuÀrios. Isto ¢ tipicamente Ãtil no caso de diretÂrios como `/tmp`. Normalmente o diretório `/tmp` possui acesso universal para gravação, permitindo que qualquer usuário remova arquivos pertencentes a qualquer outro usuário. Ativando o *sticky* bit no diretÂrio `/tmp`, os usuÀrios podem remover apenas seus prÂprios arquivos. Para ativar o *sticky* bit em um diretÂrio, 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 que 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 seguinte linha:

```
umask 022
```

para impedir a criação acidental de arquivos pertencentes ao superusuário com permissão 777.

A segurança de dispositivos é uma questão importante em sistemas Unix. Arquivos de dispositivo são usados por vários programas para acessar dados nos discos rígidos 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 arquivos `/dev/kmem`, `/dev/mem` e `/dev/drum` não devem ter permissão de leitura universal.

- Os dispositivos de disco, tais como `/dev/sd0a`, `/dev/rx1b`, etc., devem pertencer ao usuário *root* e grupo *operator,* e devem possuir modo 640.
- Com 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 utilizando no momento. Ao desconectar-se, o terminal volta a pertencer ao *root.*

Segurança é sua responsabilidade

Existem diversas ferramentas para aperfeiçoar a segurança de sistemas Unix. Embora estas ferramentas estejam disponíveis na maior parte dos sistemas, normalmente não são utilizadas. É tarefa do administrador de sistemas despender o tempo e o esforço necessário para disponibilizar estas ferramentas e, desta forma, proteger o sistema de acessos não autorizados.

NOTAS:

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 de cada usuário do sistema. A mensagem impressa no terminal a cada vez que um usuário loga

```
Last login: Sat Mar 10 10:50:48 from ccvax.unicamp.br
```

usa a data armazenada no arquivo `lastlog`. A data do último login relatada pelo comando `finger` também usa estes dados. Os usuários devem ser alertados a inspecionar esta data para certificarem-se de que não foi efetuado nenhum acesso não autorizado às suas 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`:

```
%who
queiroz  tty0c   Mar  13  13:54
ivete    tty14   Mar  13  12:15
sandra   tty15   Mar  13  08:16
zanini   ttyp1   Mar  9   07:03  (cesar.unicamp.br)
```

Para cada usuário é exibido o userid, o terminal sendo utilizado e o computador remoto(se o login foi efetuado via rede).

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

```
%who /usr/adm/wtmp
queiroz     tty0c     Mar    13     13:54
            tty0c     Mar    13     15:34

ivete       tty14     Mar    13     11:49
            tty14     Mar    13     12:15
```

Uma linha que contem o *userid* indica a hora em que o usuário logou; uma linha que não contem o userid indica a hora em que o usuário desconectou-se do sistema. Infelizmente a saída deste comando é raramente exibida como acima. Se vários usuários logaram ao mesmo tempo os tempos de *login* e *logout* ficam misturados e precisam ser ajustados manualmente.

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` exibe toda a informação contida no arquivo.

O arquivo `acct` registra a execução de cada comando no sistema, quem o executou, quando e quanto tempo gastou. Esta informação é registrada cada vez que um comando é completado, mas apenas se o kernel foi compilado com a opção SYSACCT ativada.

O arquivo `acct` pode ser examinado através do comando `lastcomm`. Se invocado sem argumentos toda a informação do arquivo é exibida. O comando `lastcomm` aceita como argumentos o nome de um comando, de um usuário ou de um terminal.

A monitoração da segurança da rede é mais difícil, porque existem inúmeros mecanismos que um invasor pode utilizar para penetrar no sistema. Existem entretanto alguns programas que auxiliam nesta tarefa.

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 que foram enviadas pelo programa que as gerou.

De particular interesse são as mensagens geradas pelos programas `login` e `su`. Sempre que alguém loga como *root* o comando `login` registra esta informação. Normalmente, logar diretamente como *root* deve ser desencorajado, visto ser difícil identificar quem está realmente utilizando a conta. Uma vez que esta possibilidade tenha sido desabilitada identificar violações de segurança se resume a analisar o arquivo de mensagens procurando as mensagens geradas pelo comando `su`.

Outro tipo de evento a ser monitorado são pessoas tentando insistentemente logar em determinada conta e não conseguindo. Após três tentativas o programa login não mais permite que a pessoa tente novamente.

O programa `su` registra o sucesso ou o fracasso da operação. Estas mensagens podem ser usadas para verificar se os usuários estão compartilhando suas senhas bem como identificar um invasor que conseguiu uma conta e está tentando utilizar outras.

O comando `showmount` pode ser usado em um servidor NFS para exibir o nome de todas as máquinas que estão montando algum de seus diretórios. Se invocado sem opções o programa simplesmente exibe uma lista de todos os computadores. Com as opções `-a` e `-d` a saída é mais útil. A opção `-a` faz com que o comando `showmount` liste todos as combinações de computadores e diretórios:

```
cesar.unicamp.br:/pub/pub2
fee.unicamp.br:/pub
wuarchive.wustl.edu:/simtel20
```

Será exibida uma linha para cada diretório montado por uma máquina. A opção `-d` faz com que seja exibida uma lista de todos os diretórios que estão montados por alguma máquina.

Deve ser verificado que apenas máquinas locais montem os diretórios exportados e que apenas diretórios normais estejam sendo montados.

Monitoração da Segurança

Segurança 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 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. O comando

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

localiza todos os arquivos do sistema com os bits setuid ou setgid ligados.

A saída deste comando deve ser analisada para determinar se não existe algum arquivo suspeito na lista.

O comando

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

identifica todos os arquivos com permissão de escrita universal.

O comando

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

identifica arquivos que não pertencem a nenhum usuário ou a nenhum grupo.

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 novamente e compare-o com o arquivo mestre:

```
# diff MasterChecklist Currentlist
```

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

Checklist

- Faça backups
- Rode os comandos grpck, pwdck e usrck diariamente
- Verifique os dados do sistema de account
- Rode o comando errpt ao menos semanalmente
- Certifique-se de que o sistema de registro de erros esteja sempre ativo

NOTAS:

Conheça seu sistema

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 atividades suspeitas

NOTAS:

Comandos úteis

- **ps**
- **who**
- **w**
- **ls**

NOTAS:

Ferramentas Úteis

- **npasswd**
- **cops**
- **tripwire**
- **watcher**
- **tcpwrapper**
- **kerberos**

NOTAS:

Leituras Recomendadas

- ***Unix System Administration Handbook***
  Evi Nemeth, Garth Snider, Scott Seebass
  Prentice Hall
- ***Unix Operating System Security***
  F.T.Grammp/R. H. Morris
  AT&T Bell Labs Technical Journal
- ***Password Security: A Case History***
  Robert Morris/Ken Thompson
  Communications of the ACM
- ***On the Security of UNIX***
  Dennis M. Ritchie
- ***The Cuckoo's Egg***
  Clifford Stoll
  Doubleday
- ***Improving the Security of your Unix System***
  David A. Curry, Systems Programmer
  ftp.unicamp.br:
  /pub/security/info/security.doc.tar.z

NOTAS:

4

Gerenciamento de Usuários

Criação de Usuários

smitty mkuser

```
                    Create User

Type or select values in entry fields
Press Enter AFTER making all desired changes

[TOP]                                [Entry Fields]
* User NAME                           []        +
  User ID                             []        #
  LOGIN user?                          true     +
  PRIMARY group                       []        +
  Group SET                           []        +
  ADMINISTRATIVE groups               []        +
  SU groups                           [ALL]     +
  HOME directory                      []
  Initial PROGRAM                     []
  User Information                    []
  Another user can SU to user?         true     +
  User can RLOGIN?                     true     +
  TRUSTED PATH?                        nosak    +
  Valid TTYs                          [ALL]
  AUDIT classes                       []
  PRIMARY Authentication method       [SYSTEM]
  SECONDARY Authentication method     [NONE]
  Max FILE size                       [2097151]
  Max CPU time                        [-1]
  Max DATA segment                    [262144]  #
  Max STACK size                      [65536]   #
  Max CORE file size                  [2048]    #
  Max physical MEMORY                 [65536]   #
  File creation UMASK                 [022]
  EXPIRATION date (MMDDhhmmyy)        [0]
```

NOTAS:

User NAME	**Obrigatório. Deve conter 8 bytes ou menos. Não pode iniciar por -, +, ou ~. Não pode conter ":" ou usar as palavras ALL ou default.**
ADMINISTRATIVE User	**Opcional. Indica o status administrativo do usuário. Apenas o usuário *root* pode alterar este atributo.**
User ID	**Opcional. Identificação do usuário. Se não fornecido este valor é atribuído pelo sistema**
LOGIN User	**Opcional. Indica se o usuário pode acessar o sistema usando o comando `login`**
PRIMARY Group	**Opcional. O grupo principal ao qual o usuário irá pertencer. O grupo default é *STAFF***
Group set	**Opcional. Os grupos aos quais este usuário pertence**
ADMINISTRATIVE Groups	**Opcional. Os grupos dos quais este usuário será o administrador**
SU Groups	**Opcional. A lista de grupos que podem usar o comando `su` para mudarem para a conta especificada. O caracter "!"em frente a um grupo o exclui da lista. Se este atributo não for especificado todos os grupos podem acessar esta conta através do comando `su`.**
HOME Directory	**Opcional. O caminho completo do diretório de trabalho do usuário. Por default o sistema atribui o valor `/u/`*Usuário.***
Initial PROGRAM	**Opcional. O programa a ser rodado quando uma sessão é iniciada. O default é `/bin/ksh`.**
User Information	**Opcional. Informações sobre o usuário.**

Another user CAN SU to user	**Opcional. Indica se um outro usuário pode acessar esta conta com o comando `su`.**
User CAN RLOGIN	**Opcional. Indica se esta conta pode ser acessada através do comando `rlogin`.**
Trusted Path	**Opcional. Indica o status do caminho confiável para este usuário.**
Valid TTYs	**Opcional. A lista de terminais que este usuário pode utilizar.**
AUDIT classes	**Opcional. As classes de auditoria para as quais este usuário será auditado.**
PRIMARY Authentication Method	**Opcional. Métodos primários para autenticar este usuário. Valores válidos são SYSTEM e NONE.**
SECONDARY Authentication Method	**Opcional. Métodos secundários para autenticação deste usuário. Não existem métodos secundários de autenticação de usuários disponíveis no momento.**
Max FILE size(*|)**	**Opcional. Tamanho do maior arquivo que este usuário pode criar.**
Max CPU time(*)	**Opcional. A maior quantidade de tempo de CPU, em segundos, que o usuário pode utilizar.**
Max DATA segment(*|)**	**Opcional. O maior segmento de dados para processos do usuário.**
Max STACK size(*|)**	**Opcional. Tamanho do maior segmento de pilha que um processo deste usuário pode criar.**
Max CORE File size(*|)**	**Opcional. Tamanho do maior arquivo de core que um processo do usuário pode criar.**
Max Physical MEMORY(*|)**	**Opcional. A maior quantidade de memória física que processos deste usuário podem alocar.**
File creation UMASK	**Opcional. Permissões que um arquivo não terá ao ser criado pelo usuário.**
EXPIRATION Date	**Data de expiração desta conta.**

*Valores menores ou iguais a zero implicam na inexistência de restrições.
** Especificado em unidades de blocos de 512Kbytes

Alteração de atributos

- **smitty chuser**

```
                Change User Attributes

 Type or select a value for the entry field
 Press Enter AFTER making all desired changes

                                    [Entry Fields]
 * User NAME                      []              +
```

```
F1=Help      F2=Refresh     F3=Cancel   F4=List
Esc+5=Reset  Esc+6=Command  Esc+7=Edit  Esc+8=Image
Esc+9=Shell  Esc+0=Exit     Enter=Do
```

NOTAS:

Alteração de atributos

```
                Change User Attributes

Type or select values in entry fields
Press Enter AFTER making all desired changes

[TOP]                              [Entry Fields]
* User NAME                            queiroz
  User ID                           [1020]    #
  LOGIN user?                        true     +
  PRIMARY group                     [sys]      +
  Group SET                         [sys,staff]
  ADMINISTRATIVE groups             []       +
  SU groups                         [ALL]    +
  HOME directory                    [/u/queiroz]
  Initial PROGRAM                   [/bin/ksh]
  User Information                  [Sup. Unix]
  Another user can SU to user?          true    +
  User can RLOGIN?                      true    +
  TRUSTED PATH?                         nosak   +
  Valid TTYs                          [ALL]
  AUDIT classes                       []
  PRIMARY Authentication method       [SYSTEM]
  SECONDARY Authentication method     [NONE]
  Max FILE size                       [2097151]
  Max CPU time                        [-1]
  Max DATA segment                    [262144] #
  Max STACK size                      [65536]  #
  Max CORE file size                  [2048]   #
  Max physical MEMORY                 [65536]  #
  File creation UMASK                 [022]
  EXPIRATION date (MMDDhhmmyy)        [0]
```

NOTAS:

Listar Usuários

smitty lsuser

```
                COMMAND STATUS

Command: OK     stdout: yes     stderr: no

Before command completion, additional
instructions may appear bellow

[TOP]

root  0               /
daemon                1           /etc
bin   2               /bin
sys   3               /usr/sys
adm   4               /usr/adm
uucp  5               /usr/lib/uucp
guest 100             /usr/guest
nobody                -2          /
lpd   104             /
queiroz               1020        /u/queiroz
sandra                1001        /u/sandra
ana   1002            /u/ana
[MORE..83]


F1=Help     F2=Refresh     F3=Cancel   F4=List
Esc+5=Reset Esc+6=Command  Esc+7=Edit  Esc+8=Image
Esc+9=Shell Esc+0=Exit     Enter=Do
```

NOTAS:

Este comando exibe uma lista de todos os usuários no sistema juntamente com seus UIDs e diretórios de trabalho.

Listar Atributos

```
% lsuser queiroz
queiroz id=1020 pgrp=supsof
groups=supsof,sys,usr
home=/u/queiroz shell=/ksh gecos=Sup
UNIX
```

```
# lsuser queiroz
queiroz id=1020 pgrp=supsof
groups=supsof,sys,usr
home=/u/queiroz shell=/ksh gecos=Sup
Unix login=true su=true rlogin=true
daemon=true admin=false sugroups=ALL
tpath=nosak ttys=ALL expires=0
auth1=SYSTEM auth2=NONE umask=22
fsize=2097151 cpu=-1 data=262144
stack=65536 core=2048 rss=65536
time_last_login=781614408
time_last_unsuccess-
ful_login=781534614
tty_last_login=pts/4 tty_last_unsuc-
cessful_login=pts/27
host_last_login=paris host_last_un-
successful_login=paris
unsuccessful_login_count=0
```

NOTAS:

Listar Atributos

```
# lsuser -f queiroz
queiroz:
      id=1020
      pgrp=supsof
      groups=supsof,sys,usr
      home=/home/cesar/supsof/queiroz
      shell=/ksh
      gecos=Sup UNIX
      login=true
      su=true
      rlogin=true
      daemon=true
      admin=false
      sugroups=ALL
      tpath=nosak
      ttys=ALL
      expires=0
      auth1=SYSTEM
      auth2=NONE
      umask=22
      fsize=2097151
      cpu=-1
      data=262144
      stack=65536
      core=2048
      rss=65536
      time_last_login=781614408
      time_last_unsuccessful_login=781534614
      tty_last_login=pts/4
      tty_last_unsuccessful_login=pts/27
host_last_login=paris
      host_last_unsuccessful_login=paris
      unsuccessful_login_count=0
```

NOTAS:

Listar Atributos

```
$ lsuser -f queiroz
queiroz:
        id=1020
        pgrp=supsof
        groups=supsof,sys,usr
        home=/u/queiroz
        shell=/ksh
        gecos=Sup. UNIX
```

```
$ lsuser -a id groups queiroz
queiroz id=1020
groups=supsof,sys,usr
```

```
$ lsuser -a -f id groups queiroz
queiroz:
        id=1020
        groups=supsof,sys,usr
```

NOTAS:

Remoção de Usuários

smitty rmuser

```
                Remove a User from the System

Type or select a value for the entry field
Press Enter AFTER making all desired changes

                                      [Entry Fields]
* User NAME                              []      +
  Remove AUTHENTICATION Information      Yes     +

F1=Help      F2=Refresh     F3=Cancel   F4=List
Esc+5=Reset  Esc+6=Command  Esc+7=Edit  Esc+8=Image
Esc+9=Shell  Esc+0=Exit     Enter=Do
```

NOTAS:

Se voce desejar remover a password do usuário e outras informações usadas para autenticação contidas no arquivo `/etc/security/passwd`, selecione `Yes` no campo `Remove Authentication Information`.

Para efetuar esta operação sem utilizar o smit utilize o comando `rmuser`. O comando `rmuser` com a opção `-p` remove o usuário e todas as informações de autenticação.

Por exemplo, para remover o usuário joao e todas as suas informações de autenticação emitir, como *root*, o comando:

```
# rmuser -p joao
```

Alteração de senhas

smitty passwd

```
                Change User Password

Type or select a value for the entry field
Press Enter AFTER making all desired changes

                                    [Entry Fields]
* User NAME                              [joao]
+

F1=Help      F2=Refresh     F3=Cancel   F4=List
Esc+5=Reset  Esc+6=Command  Esc+7=Edit  Esc+8=Image
Esc+9=Shell  Esc+0=Exit     Enter=Do
```

```
Changing password for “joao”
Enter root’s password or joao’s Old password:
joao’s New password: ********
Enter the new password again:
```

NOTAS:

Nos sistemas AIX existem duas maneiras de se trocar a password de um usuário: através dos comandos `passwd` e `pwdadm`. O comando `passwd` simplesmente efetua a troca da senha do usuário ao passo que o comando `pwdadm` ativa o atributo *ADMCHG*, que força o usuário a trocar sua password na próxima vez que acessar o sistema ou quando sua conta for acessada através do comando `su`.

Definição de Atributos Default

- **Atributos default são armazenados no arquivo `/usr/lib/security/mkuser.default`**
- **mkuser.default:**

```
 user:
        group = staff
        groups = staff
        prog = /bin/ksh
        home = /u/$USER
        auth1 = SYSTEM
        auth2 = NONE
admin:
        group = system
        groups = system
        prog = /bin/ksh
        home = /u/$USER
```

NOTAS:

Os atributos default empregados na criação de novos usuários são armazenados no arquivo `/usr/lib/security/mkuser.default`. Estes valores são lidos pelo comando `mkuser` e utilizados a menos que sejam fornecidos outros na linha de comando. Para definir ou alterar estes valores, este arquivo deve ser editado.

Bloqueio de acesso

O acesso ao sistema pode ser bloqueado negando-se ao usuário o uso de comandos que são usados para se logar no sistema. Dois tipos de login podem ser controlados:

- **Logins locais através do comando login**
- **Logins remotos através dos comandos rsh, rlogin e telnet**

OBS.: Este tipo de controle não é válido para ambientes NIS. A restrição somente se aplicaria ao NIS master.

NOTAS:

Bloqueio de acesso

- **Impedir que o usuário `joao` acesse o sistema usando os comandos telnet, rlogin, rsh:**

  `#chuser login=no rlogin=no joao`

- **Restituir ao usuário `joao` o direito de acesso:**

  `#chuser login=yes rlogin=yes joao`

- **`smitty chuser`**

NOTAS:

Arquivos - 1

- **/etc/passwd**

```
root:!:0:0::/:/bin/ksh
daemon:!:1:1::/etc:
ze:!:155:100:Jose:/u/ze:/bin/csh
...
```

- **/etc/security/passwd**

```
yazmin:
        password = Y7mhg9qU3uIpU
        lastupdate = 780752101
backup:
        password = bsy04R5phqdKM
        lastupdate = 780950764
        flags = ADMCHG
```

- **/etc/group**

```
system:!:0:root
staff:!:1:moretti,daemon
bin:!:2:bin,root,backup
```

NOTAS:

Arquivos - 2

- **/etc/security/user**

```
default:
        admin = false
        login = true
        su = true
        daemon = true
        rlogin = true
        sugroups = ALL
        ttys = ALL
        auth1 = SYSTEM
        auth2 = NONE
        tpath = nosak
        umask = 022
        expires = 0
root:
        admin = true
        login = true
        rlogin = false
        sugroups = sys
```

NOTAS:

Arquivos - 3

- **/etc/security/login.cfg**

```
pw_restrictions:
        maxage = 0
        minage = 0
        minalpha = 0
        minother = 0
        minalpha = 0
        minother = 0
        mindiff = 0
        maxrepeats = 8
port:
        sak_enabled = false
        herald = “login:”
        aliases =
usw:
shells=/bin/sh,/bin/bsh,/bin/csh
,/bin/ksh,/bin/tsh,/usr/bin/sh,/
usr/bin/bsh,/usr/bin/csh,/usr/bi
n/ksh,/usr/bin/tsh,/usr/mbin/sh,
/usr/mbin/bsh,/usr/mbn/csh,/usr/
mbin/ksh,/usr/mbin/tsh
   maxlogins = 0
```

NOTAS:

5

Gerenciamento de Grupos

Criação de Grupos

```
# smit mkgroup
```

```
                    Add Group

Type or select values in entry fields
Press Enter AFTER making all desired
changes

  Group NAME                   []
  ADMINISTRATIVE group?         false  +




F1=Help     F2=Refresh      F3=Cancel     F4=List
Esc+5=Reset Esc+6=Command   Esc+7=Edit    Esc+8=Image
Esc+9=Shell Esc+0=Exit      Enter=Do
```

```
# mkgroup [-a|-A] finance
```

NOTAS:

Alteração de Atributos

```
# smit chgroup
```

```
        Change Group Attributes

Type or select values in entry fields
Press Enter AFTER making all desired changes

  Group NAME                [sys]



F1=Help     F2=Refresh      F3=Cancel     F4=List
Esc+5=Reset Esc+6=Command   Esc+7=Edit    Esc+8=Image
Esc+9=Shell Esc+0=Exit      Enter=Do
```

```
        Change Group Attributes

Type or select values in entry fields
Press Enter AFTER making all desired changes

                             [Entry Fields]
  Group NAME                [sys]
  Group ID                  [3]              #
  ADMINISTRATIVE group       true           +
  USER list                 [sys,joao]      +
  ADMINISTRATOR list        []


F1=Help     F2=Refresh      F3=Cancel     F4=List
Esc+5=Reset Esc+6=Command   Esc+7=Edit    Esc+8=Image
Esc+9=Shell Esc+0=Exit      Enter=Do
```

NOTAS:

A alteração dos atributos de um grupo pode também ser feita diretamente na linha de comandos através do comando chgroup:

chgroup users=joao,ze,maria finance

O comando acima define como membros do grupo finance os usuários joao, ze e maria.

chgroup users=joao,ze adms= finance

O comando acima retira a usuária maria do grupo finance e remove todos os administradores do grupo.

Listar Informações sobre Grupos (1)

```
# smit lsgroup
```

```
        Command Status

Before command completion, additional
instructions may appear below.

[TOP]

system   0          root,backup,ze
staff    1          paulo,joao,maria
bin      2          bin,root
sys      3          paulo,joao,ze,maria
adm      4          adm,root,bin
[MORE...34]

F1=Help     F2=Refresh     F3=Cancel      Esc+6=Command
Esc+8=Image Esc+9=Shell    Esc+0=Exit
```

NOTAS:

Listar Informações sobre Grupos (2)

 # lsgroup -a id users ALL

```
system id=0 users=root
staff id=1 users=moretti,daemon
bin id=2 users=bin,root,backup
sys id=3 users=sys,queiroz,san-
dra,ivete
adm id=4
...
```

 # lsgroup -a -f users ALL

```
system:
        id=0
        users=root,backup
staff:
        id=1
        users=moretti,daemon
```

NOTAS:

Remoção de Grupos

```
# smitty rmgroup
```

```
        Remove a Group from the System
Type or select values in entry fields
Press Enter AFTER making all desired changes

  Group NAME                  [finance]



F1=Help     F2=Refresh      F3=Cancel      F4=List
Esc+5=Reset Esc+6=Command   Esc+7=Edit     Esc+8=Image
Esc+9=Shell Esc+0=Exit      Enter=Do
```

```
# rmgroup finance
```

NOTAS:

Os usuários pertencentes a um grupo que foi deletado não são removidos do sistema. Caso o grupo removido seja o grupo primário de um usuário o grupo não poderá ser removido. Todos os usuários que tenham o grupo em questão como grupo primário terão que ser alterados para passar a pertencer a um outro grupo primário.

Apenas o usuário pode remover um grupo administrativo ou um grupo ao qual pertençam administradores.

6

Volumes Lógicos

Logical Volume Storage

Método transparente para usuarios e aplicativos de dividir e alocar espaço de armazenamento nos discos do sistema

Volumes Físicos (Physical Volumes)

Discos conectados ao sistema com informações de configuração e identificação nele gravadas

Grupos de volumes (Volume Groups)

Um grupo de volumes é um conjunto de 1 a 32 volumes físicos de tamanho e tipo variados

Partições Físicas (Physical Partitions)

Cada volume físico é particionado em unidades de espaço contíguas e de igual tamanho, denominadas partições físicas

Volumes Lógicos (Logical Volumes)

Conjunto de partições lógicas onde reside o sistema de arquivos

Partições Lógicas (Logical Partitions)

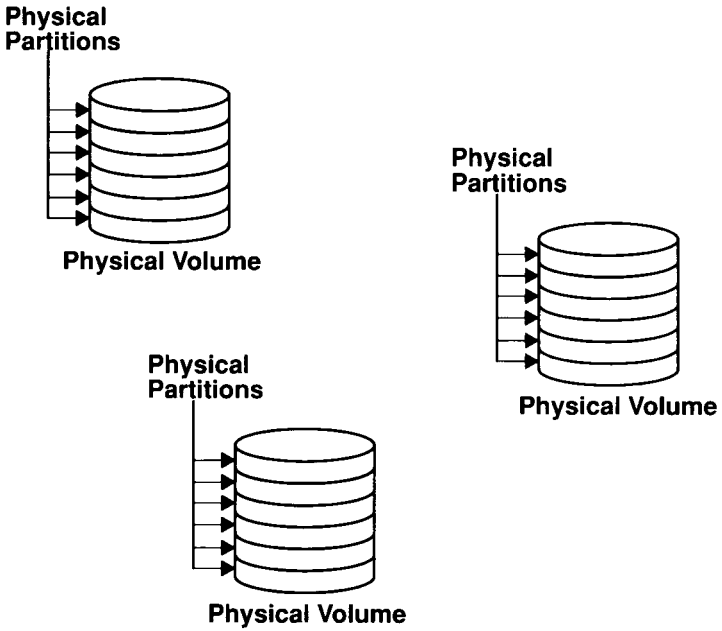
Um volume lógico é composto de partições lógicas. As partições lógicas podem conter de uma a tres partições físicas, dependendo do número de cópias especificadas para o volume lógico

Espelhamento (Mirroring)

Facilidade oferecida pelo sistema operacional de se ter mais de uma cópia de um volume lógico

Vantagens do Armazenamento Lógico

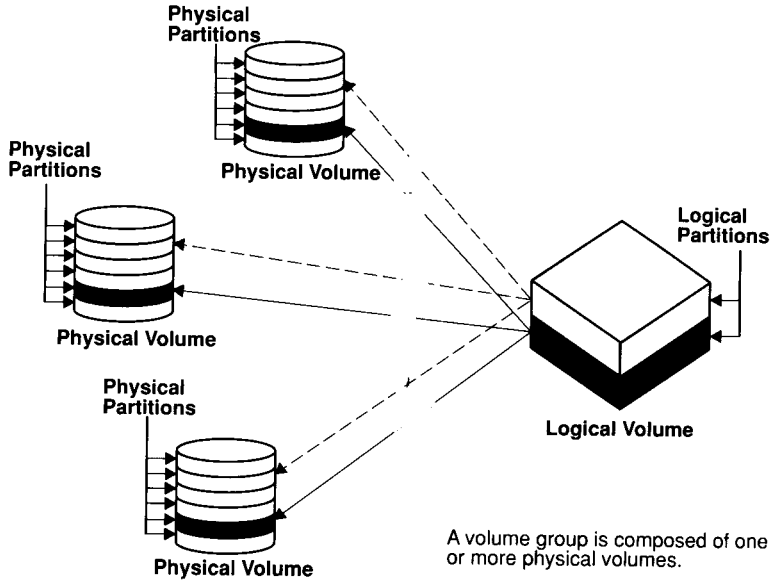
- Facilidades para expansão do sistema de arquivos



- Suporte a duplicação de dados (mirroring)
- Dados de um mesmo sistema de arquivos podem ser distribuídos por vários discos físicos
- O sistema operacional usa volumes lógicos distintos para finalidades específicas tais como paginação, log de journal, dados de boot, etc.

Logical Volume Manager

Conjunto de comandos do sistema operacional, bibliotecas de subrotinas e outras ferramentas que permitem criar e controlar volumes lógicos



Componentes do LVM

- **LVDD**
  Logical Volume Device Driver
- **LVM subroutine interface library**

varyonvg
varyoffvg

Comandos utilizados para ativar e desativar grupos de volumes definidos para o sistema

Comandos Importantes (1)

- chvg

```
% chvg -a y vg03
```

- mkvg

```
% mkvg -s 1 hdisk3 hdisk5 hdisk6
```

- exportvg

```
exportvg vg02
```

Comandos Importantes (2)

- importvg

```
% importvg -y bkvg hdisk07
```

- reorgvg

```
% reorgvg vg02 lv03 lv04 lv07
```

- syncvg

```
% syncvg -v vg04 vg05
```

Comandos Importantes (3)

- chpv

```
% chpv -v r hdisk03
% chpv -v a hdisk03
% chpv -v n hdisk03
```

- lspv

```
% lspv hdisk3
% lspv -p hdisk5
% lspv
```

Comandos Importantes (4)

- migratepv

```
% migratepv -l lv02 hdisk1 hdisk6
```

- mklv

```
% mklv -c 2 vg02 1
% mklv -c 3 -u 2 -s n vg03 9
% mklv -a c -t paging -b n vg04 5
% mklv vg03 15 hdisk5 hdisk6
```

Comandos Importantes (5)

- chlv

```
% chlv -e m lv01
% chlv -t copy lv03
% chlv -p r lv03
% chlv -t paging -u 10 lv03
```

- rmlv

```
% rmlv -f lv05
```

Comandos Importantes (6)

- rmfs

```
% rmfs /test
```

- mklvcopy

```
% mklvcopy lv01 3
```

- rmlvcopy

```
% rmlvcopy lv03 2
```

Comandos Importantes (7)

- **extendlv**

```
% extendlv lv05 3
```

- **cplv**

```
% cplv lv03
% cplv -v vg02 lv03
```

- **reducevg**

```
% reducevg vg01 hdisk1
% reducevg -d -f vg01 hdisk1
```

- **extendvg**

```
% extendvg vg3 hdisk3 hdisk8
```

Sistemas de Arquivos

- **Um sistema de arquivos é uma estrutura hierárquica de arquivos e diretórios.**
- **Algumas tarefas, para serem executadas mais eficientemente, devem ser realizadas sobre os sistemas de arquivos.**
- **O sistema de arquivos nativo é chamado de *journalled file system*. Esta técnica impede inconsistências no sistema de arquivos em caso de encerramento anormal do sistema.**
- **Um sistema de arquivos reside em um único volume lógico.**
- **Precisam estar montados para serem acessados.**

NOTAS:

Um sistema de arquivos é uma estrutura hierárquica (árvore de arquivos) de arquivos e diretórios. Este tipo de estrutura se assemelha a uma árvore invertida com as raízes no topo e os galhos no fundo. Esta árvore de arquivos usa diretórios para organizar dados e programas em grupos, permitindo o gerenciamento de vários diretórios e arquivos a um só tempo.

Algumas tarefas são desempenhadas mais eficientemente sobre um sistema de arquivos do que sobre cada diretório dentro do sistema de arquivos.

O sistema de arquivos nativo é denominado *journaled file system*. Este tipo de sistema de arquivos usa técnicas de *journaling* para manter a sua consistência estrutural. Isto impede danos ao sistema de arquivos em caso de término anormal do sistema.

Um dado sistema de arquivos reside em um único volume lógico. O comando `mkfs` (*make file system*) ou o SMIT (comando *smit mkfs*) cria um sistema de arquivos em um volume lógico. Todo arquivo e diretório pertence a um sistema de arquivos dentro de um volume lógico.

Para se acessar um sistema de arquivos, este deve estar montado sobre um diretório (*mount point*). Quando vários sistemas de arquivos são montados, é criada uma estrutura de diretórios semelhante à imagem de um único sistema de arquivos. Existe uma estrutura hierárquica com uma única raiz. Esta estrutura inclui os sistemas de arquivos base e quaisquer outros sistemas de arquivos que sejam criados.

Sistemas de Arquivos

/dev/hd1	**/home**
/dev/hd2	**/usr**
/dev/hd3	**/tmp**
/dev/hd4	**/**
/dev/hd9var	**/var**

NOTAS:

Na versão 3.2, a árvore de diretórios padrão foi significantemente reestruturada para facilitar o gerenciamento de máquinas diskless e dataless.

Cada sistema de arquivos reside em um volume lógico separado, montados pelo sistema operacional durante a inicialização. Esta configuração é útil para auxiliar em tarefas de gerenciamento de sistema tais como backup, restore e reparos, porque uma parte da árvore de arquivos é isolada das demais.

Arquivos locais e remotos podem ser acessados através do comando `mount`. Desta forma o sistema de arquivos fica disponível para atividades de leitura ou leitura e gravação a partir do sistema local. Para montar ou desmontar arquivos o usuário precisa pertencer ao grupo `system`. Os sistemas de arquivos definidos no arquivo `/etc/filesystems` são montados automaticamente durante a inicialização do sistema. Sistemas de arquivos tanto remotos quanto locais podem ser desmontados usando-se o comando `umount`, a menos que um usuário ou processo os esteja acessando.

Tipos de Sistemas de Arquivos

- **Journaled File System(JFS)**
- **Network File System(NFS)**
- **CD-ROM File System(CRFS)**

NOTAS:

A versão 3 do AIX suporta vários tipos de sistemas de arquivos:

- **JFS** (*Journaled File System*)
  o sistema nativo do AIX.
- **NFS** (*Network File System*)
  um tipo de sistema de arquivos que permite que arquivos residentes em máquinas remotas sejam acessados como se residissem na máquina local
- **CRFS** (*CD-ROM File System*)
  um tipo de sistema de arquivos que permite que o conteúdo de um CD-ROM seja acessado atravs das interfaces normais dos sistemas de arquivos (*open, read, close*).

Comandos de Gerenciamento de Sistemas de Arquivos

- **chfs**
- **crfs**
- **lsfs**
- **rmfs**
- **mount**

NOTAS:

Existem vários comandos para gerenciar sistemas de arquivos, independente de seu tipo. O arquivo `/etc/filesystems` controla a lista de sistemas de arquivos que a seguinte lista de comandos pode manipular:

`chfs`	Altera as características do sistema de arquivos
`crfs`	Adiciona um sistema de arquivos
`lsfs`	Exibe as características de um sistema de arquivos
`rmfs`	Remove um sistema de arquivos
`mount`	Torna um sistema de arquivos disponível para uso

Tarefas de Gerenciamento de Sistemas de Arquivos

- **Alocação de espaço para sistemas de arquivos em volumes lógicos**
- **Criação de sistemas de arquivos**
- **Tornar os sistemas de arquivos disponíveis para os usuários**
- **Monitoração do uso de espaço em disco**
- **Backup dos sistemas de arquivos**
- **Manter a consistência dos sistemas de arquivos**

NOTAS:

Um sistema de arquivos é uma estrutura completa de diretórios, que inclui o diretório raiz e diretórios e arquivos abaixo dele. Os sistemas de arquivos são confinados a um único volume lógico.

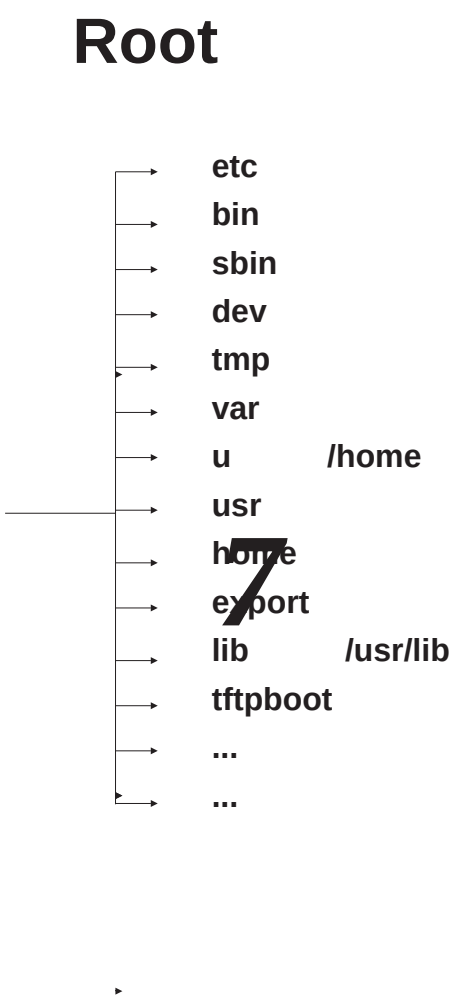
Comandos Úteis

- **backup**
- **dd**
- **df**
- **fsck**
- **mkfs**
- **mount**
- **restore**
- **umount**

NOTAS:

Os seguintes comandos são usados regularmente para gerenciar sistemas de arquivos:

`backup`	Realiza backups totais ou incrementais de sistemas de arquivos
`dd`	Copia dados diretamente de um dispositivo para outro
`df`	Informa o total de espaço usado e livre em um sistema de arquivos
`fsck`	Verifica um sistema de arquivos e repara inconsistências
`mkfs`	Cria um sistema de arquivos de determinado tamanho em um volume lógico especificado
`mount`	Torna um determinado sistema de arquivos disponível para o sistema
`restore`	restaura arquivos de um backup
`umount`	Torna um sistema de arquivos indisponível para o sistema



NOTAS:

O sistema de arquivos *root* está no topo da hierarquia. Ele abriga os arquivos e diretórios críticos para a operação do sistema, incluindo o diretório de dispositivos e programas para a inicialização do sistema. O sistema de arquivos *root* também contém *mount points* onde os demais sistemas de arquivos são montados.

A seguinte lista fornece informação sobre o conteúdo de alguns subdiretórios do sistema de arquivos root:

`/etc` Contém arquivos de configuração que variam de máquina para máquina, tais como os arquivos `/etc/hosts` e `/etc/passwd`.O diretório `/etc` contém os arquivos geralmente usados na administração do sistema. A maior parte dos comandos que residiam no diretório /etc residem agora no diretório /usr/sbin. Por compatibilidade, entretanto, o diretório /etc contém links simblicos para as novas localizações. Por exemplo, o arquivo `/etc/chown` na verdade um link para `/usr/sbin/chown.`

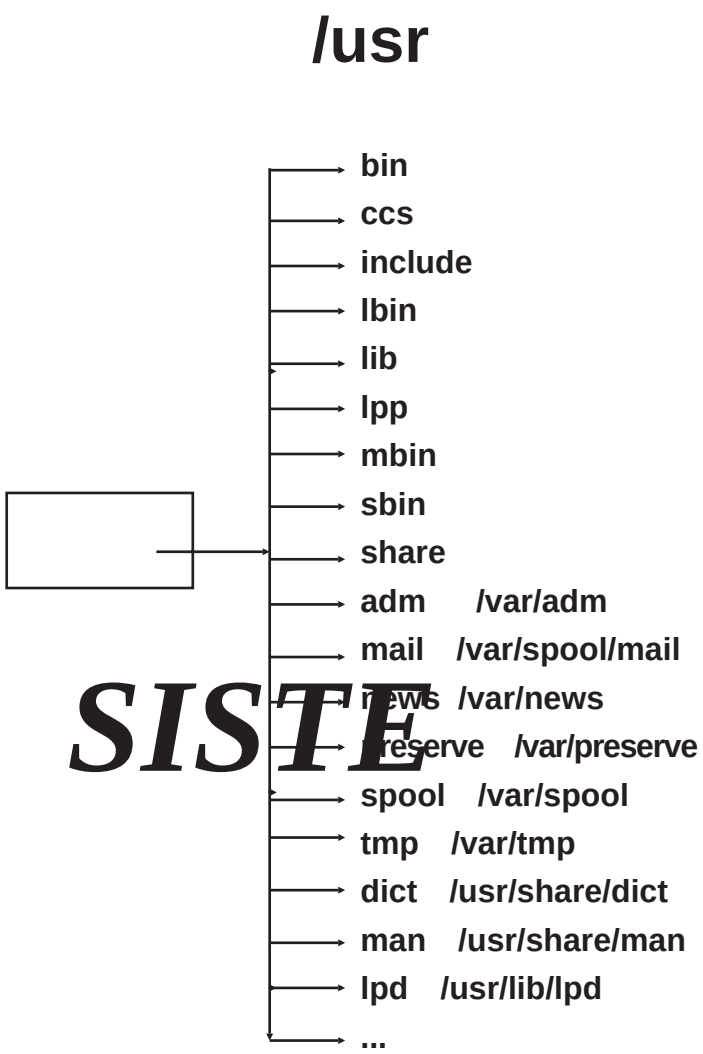
`/bin` um link simblico para o diretório `/usr/bin`. Em sistemas UNIX mais antigos, o diretório `/bin` continha comandos de usuários que agora residem no diretório `/usr/bin.`

`/sbin` contém arquivos necessrios para o boot da máquina e para montar o sistema de arquivos /usr.

`/dev` contém nós de dispositivo para arquivos especiais para dispositivos locais. O diretório `/dev` contém arquivos especiais para fitas, impressoras, partições de discos e terminais

`/tmp` *Mount point* para um sistema de arquivos que contém arquivos temporários gerados pelo sistema. O sistema de arquivos /tmp é um diretório vazio.

`/var` *Mount point* para um sistema de arquivos que contém arquivos que variam de máquina para máquina. O sistema de arquivos `/var` configurado como um sistema de arquivos independente pois os arquivos que abriga tendem a crescer.

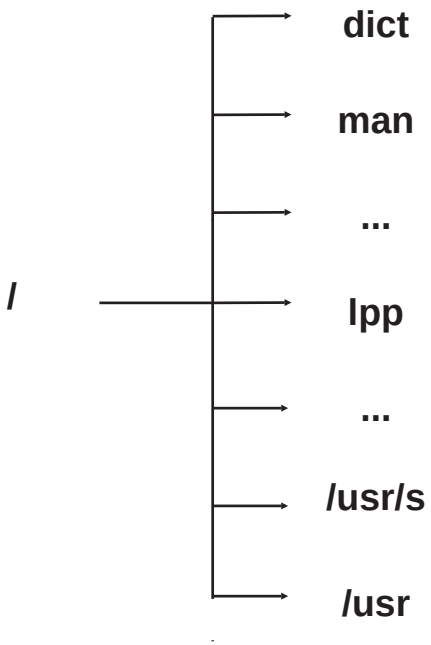


SISTE

NOTAS:

`/usr`	contém arquivos que não mudam e que podem ser compartilhados por máquinas diferentes.
`/u`	Link simblico apontando para o diretório /home
`/home`	*Mount point* para um sistema de arquivos contendo dados de usuários. Em uma máquina standalone o diretório /home est contido em um sistema de arquivos separado que é montado sobre o diretório `/home` do sistema de arquivos raiz. Em uma rede um servidor pode conter arquivos de usuários acessíveis a partir de várias máquinas. Neste caso, o diretório `/home` do servidor montado remotamente sobre um diretório `/home` local.
`/export`	contém diretórios e arquivos usados por clientes remotos.
`/lib`	Link simblico para o diretório `/usr/lib`
`/tftpboot`	contém as imagens e informações de boot para clientes diskless.

/usr/share



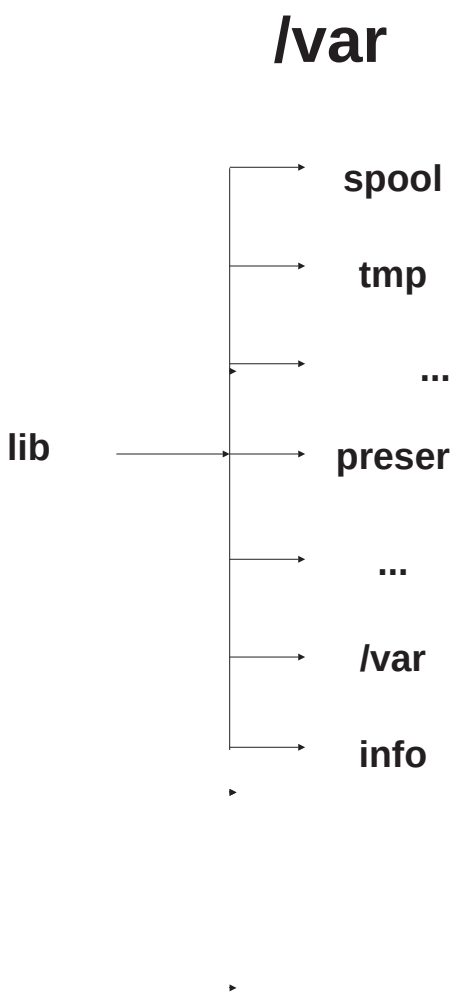
NOTAS:

O diretório `/usr/share` contém arquivos texto compartilháveis, independentemente de arquitetura. O conteúdo deste diretório pode ser compartilhado por todas as máquinas, independentemente da arquitetura de hardware.

Em uma arquitetura mista, clientes diskless montam o diretório `/usr` de um servidor sobre o seu próprio diretório `/usr` e montam de um outro servidor o diretório `/usr/share`. Os arquivos debaixo do diretório `/usr/share` estão contidos em um ou mais pacotes instalados separadamente. Desta forma, um nó pode ter alguns pacotes do diretório `/usr` instalados localmente ao mesmo tempo em que monta de um servidor o diretório `/usr/share`.

Descrição dos diretórios:

`/usr/share/man`	Man pages
`/usr/share/dict`	dicionário e seus índices
`/usr/share/info`	arquivos do banco de dados do InfoExplorer
`/usr/share/lib`	arquivos de dados independentes de arquitetura, inclusive terminfo, lear, tmac, me e macros
`/usr/share/lpp`	contém dados e informação sobre produtos opcionalmente instalavéis no sistema



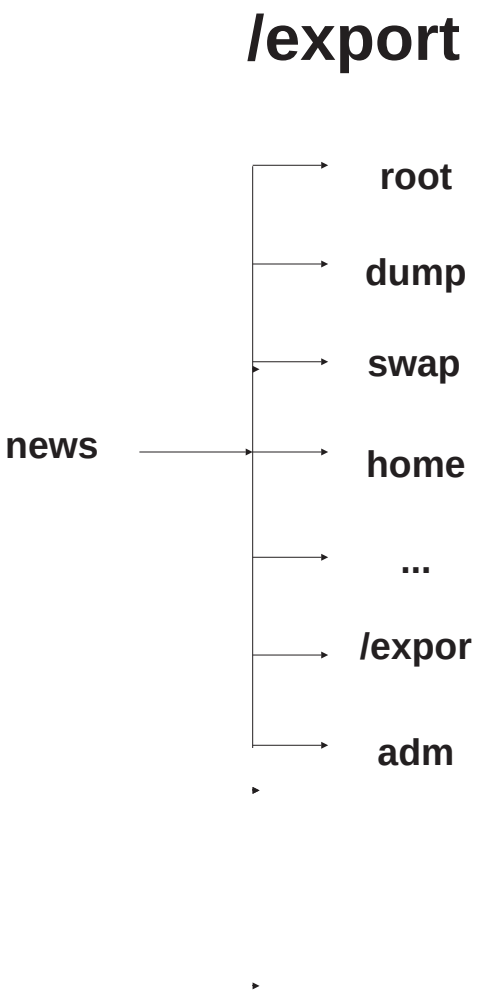
NOTAS:

O sistema de arquivos /var tende a crescer visto que contém sudiretórios e arquivos de dados que são usados por aplicações tais como contabilização, mail e spool de impressão. Se as aplicações no em seu sistema fazem uso intensivo do sistema de arquivos /var, o seu tamanho terá que ser aumentado além do tamanho default alocado quando da instalação.

Os arquivos abaixo devem ser monitorados periodicamente:

```
/var/adm/wtmp
/var/adm/ras/errlog
/var/adm/ras/trcfile
/var/adm/snmpd.log
```

`/var/adm`	contém arquivos do sistema de contabilização de recursos
`/var/news`	contém news
`/var/preserve`	contém arquivos salvos de sessões interrompidas; semelhante ao diretório `/usr/preserve` de versões anteriores do sistema operacional
`/var/spool`	contém arquivos sendo processados por programas como mail
`/var/tmp`	contém arquivos temporários; semelhante ao diretório `/usr/tmp` de versões anteriores. O diretório `/usr/tmp` é hoje um link simbólico para o diretório `/var/tmp`



NOTAS:

O diretório /export é a localização default de recursos de clientes no servidor. Como os clientes montam estes recursos sobre sua árvore de diretórios, estes recursos parecem estar, para os clientes, no lugar normal em sua árvore de diretórios. São os seguintes os subdiretórios mais importantes sob /export:

`/export/root`	Este diretório é montado sobre o sistema de arquivos root do cliente. Os diretórios do cliente estão localizados sob o diretório /export/root.
`/export/exec`	(Shared Product Object Tree - SPOT) Este diretório é montado sobre o sistema de arquivos /usr do cliente. SPOTs são versões do /usr armazenadas in diretório /export/exec e possuem nomes que refletem o nível do release do sistema operacional. Por default, o nome é RISCAIX
`/export/share`	Este diretório é montado sobre o diretório /usr/share do cliente. A localização default é /export/share/AIX/usr/share
`/export/home`	Este diretório é montado sobre o sistema de arquivos /home do cliente.
`/export/swap`	Este arquivo recebe o nome do cliente e por default é colocado no diretório /export/swap.
`/export/dump`	Sistemas standalone usam um disco local como dispositivo de dump; clientes diskless utilizam um arquivo com seu hostname colocado no diretório /export/dump.
`Microcode`	Este diretório contém microcódigo para dispositivos físicos. A localização default é /export/exec/RISCAIX/usr/lib/microcode

Criação de Sistemas de Arquivos

- **smit crjfs**

```
           Volume Group Name

Move cursor to desired field and press Enter

     rootvg
     usrvg


F1=Help             F2=Refresh              F3=Cancel
F8=Image            F10=Exit                Enter=Do
/=Find              n=Find Next
```

```
             Add a Journalled File System

Type or select values in entry fields
Press Enter AFTER making all desired changes

                                          [Entry Fields]
  Volume group name                         rootvg
* SIZE of filesystem (in 512-byte blocks   []
* MOUNT POINT                              []
  Mount AUTOMATICALLY at system restart     no
  PERMISSIONS                               read/write
  Mount OPTIONS                            []
  Start disk accounting                     no




F1=Help        F2=Refresh        F3=Cancel      F4=List
Esc+5=Reset    F6=Command        F7=Edit        F8=Image
F9=Shell       F10=Exit          Enter=Do
```

NOTAS:

Neste procedimento o sistema operacional cria e atribui um nome ao volume lógico que irá conter o sistema de arquivos. O volume lógico é dimensionado de modo a poder conter os dados do sistema de arquivos. Se mais espaço for necessário posteriormente, use o comando chfs. Este comando irá acrescentar as partições lógicas necessárias e então extenderá o sistema de arquivos às partições lógicas acrescentadas.

Alterações de Sistemas de Arquivos

```
                File System Name

Move cursor to desired field and press Enter

[TOP]
/
/usr
/home
/var
/tmp
/blv
/
[BOTTOM]

F1=Help             F2=Refresh              F3=Cancel
F8=Image            F10=Exit                Enter=Do
/=Find              n=Find Next
```

```
 Change/Show characteristics of a Journaled File System
Type or select values in entry fields
Press Enter AFTER making all desired changes

                                          [Entry Fields]
  Volume group name                         rootvg
* SIZE of filesystem (in 512-byte blocks   [8192]
* MOUNT POINT                              [/]
  MOUNT GROUP                              [bootfs]
  Mount AUTOMATICALLY at system restart     yes
  PERMISSIONS                               read/write
  Mount OPTIONS                            []
  Start disk accounting                     no

F1=Help        F2=Refresh        F3=Cancel      F4=List
Esc+5=Reset    F6=Command        F7=Edit        F8=Image
F9=Shell       F10=Exit          Enter=Do
```

NOTAS:

Diminuição do Tamanho de um Sistema de Arquivos

- Faça um backup do sistema de arquivos
- Remova o sistema de arquivos
- Recria o sistema de arquivos no tamanho desejado
- Restaure os arquivos

NOTAS:

Diminuição do Tamanho do Sistema de Arquivos /usr

- Remova os arquivos dispensáveis de /usr
- Certifique-se de que todos os sistemas de arquivos do rootvg estão montados
- # mkszfile
- Editar /.fs.size e altere o tamanho da entrada do /usr
- # chdev -l rmt0 -a block=512 -T
- Desmonte os sistemas de arquivos que não pertencem ao rootvg
- # varyoff uservg
  # exportvg uservg
- # mksysb /dev/rmt0
- Restaure o sistema com a fita gerada
- # importvg uservg
  # varyonvg uservg

NOTAS:

Verificação dos Sistemas de Arquivos / e /usr

- Identifique um disco do rootvg

```
# lspv
hdisk0 0000008600019 07c  rootvg
hdisk1 00027168dfcb11b3  localvg
```

- Insira o disquete ou fita na unidade
- Encerre o sistema

```
# shutdown -r
```

- Selecione modo de manutenção (Maintenance Mode)
- Importe e ative o rootvg

```
# getrootfs hdisk0 sh
```

- Rode o comando fsck

```
# fsck -y /dev/hd4
```

NOTAS:

O comando fsck requer que os sistemas de arquivos a serem verificados estejam desmontados. Em geral, os sistemas de arquivos `/` e `/usr` não podem ser desmontados de um disco de boot de sistema. Se se deseja rodar o comando fsck nos sistemas de arquivos `/` ou `/usr`, deve-se fazer o boot do sistema a partir de fita ou disquete.

/etc/filesystems

- Arquivo que descreve as características dos sistemas de arquivos a serem montados pelo sistema
- Exemplo:

```
/:
      dev             =/dev/hd4
      vfs             = jfs
      log             = /dev/hd8
      mount           = automatic
      check           = false
      type            = bootfs
      vol             = root
      free            = true
/home:
      dev             = /dev/hd1
      vfs             = jfs
      log             = /dev/hd8
      mount           = true
      check           = true
      vol             = /home
/pub/pub2:
      dev             = /pub
      vfs             = nfs
      nodename        = rhodes.cna
      mount           = true
      type            = nfs
      options         = bg,retry=60
```

NOTAS:

Mount

```
                    Mount a filesystem

Type or select values in entry fields.

Press Enter AFTER making all desired changes.
[Entry Fields]
FILE SYSTEM name                                    []    +
DIRECTORY over which to mount                       []    +
TYPE of file system                                       +
FORCE the mount?                                     no   +
REMOTE NODE containing the file system              []
      to mount
Mount as a REMOVABLE file system?                    no   +
Mount as a READ-ONLY system?                         no   +
Disallow DEVICE access via this mount?               no   +
Disallow execution of SUID and sgid programs         no   +
      in this file system?




F1=Help             F2=Refresh          F3=Cancel
F4=List             Esc+5=Reset         Esc+6=Command
Esc+7=Edit          Esc+8=Image         Esc+9=Shell
Esc+0=Exit          Enter=Do
```

NOTAS:

O comando `mount` torna sistemas de arquivos, arquivos, diretórios, dispositivos e arquivos especiais disponíveis para uso. O comando `mount` instrui o sistema operacional a conectar um determinado sistema de arquivos a um diretório especificado.

Um usuário pode montar um arquivo ou diretório desde que possua direitos de acesso aos mesmos. Usuários que pertençam ao grupo *system* podem também montar dispositivos e os sistemas de arquivos descritos no arquivo `/etc/filesystems`.

Umount

```
       Unmount a File System

Move cursor to desired item and press Enter.

   Unmount a File System
   Unmount a Group of File Systems



F1=Help     F2=Refresh    F3=Cancel
Esc+8=Image Esc+9=Shell   Esc+0=Exit
Enter=Do
```

```
           Unmount a File System

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                [Entry Fields]

Unmount ALL mounted file systems?   no     +
   (except /, /tmp, /usr)
                -OR-
Unmount all REMOTELY mounted        no     +
   file systems?
                -OR-
NAME of file system to unmount      []     +
REMOTE NODE containing the          []     +
file system(s) to unmount

...
```

NOTAS:

Liberação de espaço em disco

- **find**
- **xargs**
- **du**
- **commit**
- **ptfdir_clean**
- **nfs**

NOTAS:

8

PAGINAÇÃO
MEMÓRIA VIRTUAL

Área de Paginação

- **Espaço reservado no disco rígido utilizado para o armazenamento de informações residentes na memória virtual que não estejam sendo acessadas**
- **o processo de paginação acontece quando a quantidade de memória real disponível é baixa.**
- **o sistema operacional controla a paginação dos processos de acordo com dois niveis conhecidos como** *paging-space warning* **e** *paging-space kill***, que determinam o número de blocos livres do arquivo de paginação utilizados pelos processos em execução.**
- **SIGDANGER - paging-space warning level**
- **SIGKILL - paging-space kill level**

NOTAS:

A área de paginação, também conhecida como área de swap,é um volume lógico com o tipo de atributo igual a *paging.* Este tipo de volume lógico é conhecido como volume lógico de paginação, ou simplesmente área de paginação.

Existe um outro tipo de área de paginação disponível que pode ser acessado através de um dispositivo que usa um servidor NFS. Para um cliente NFS acessar o espaço de paginação residente em um servidor NFS, o servidor precisa criar o arquivo de paginação e exportá-lo para o cliente. O tamanho do arquivo irá representar o tamanho da área de paginação do cliente.

Áreas de paginação são definidas através da criação de novos volumes lógicos de paginação. O aumento das áreas de paginação é definido através da criação de um novo volume lógico de pagina§Ðo.

Quando a quantidade de memÂria real livre baixa, os programas ou dados que nÐo tenham sido usados recentemente sÐo movidos da memÂria real para a Àrea de pagina§Ðo de modo a liberar a memÂria real para outras atividades.

O sistema operacional monitora a quantidade de blocos livres do arquivo de paginação disponíveis para os processos . Se o número destes blocos está abaixo do limite estabelecido pelo nível conhecido como *paging-space warning,* então o sistema envia um sinal chamado **SIGDANGER** para os processos que estão utilizando a memória, para que eles liberem memória, recursos de paginação (área de *stack,* segmentos de memória compartilhada), etc.

Se a situação continuar e o número de blocos livres for menor do que um segundo nível conhecido como *paging-space kill*, então o sistema manda outro tipo de sinal chamado **SIGKILL** para os processos que não trataram o sinal **SIGDANGER**, até que o número de blocos livres de paginação esteja acima deste segundo nível.

Os processos que alocam memória dinâmicamente podem assegurar suficiente espaço de paginação para sua execução monitorando a quantidade de blocos livres de paginação através de rotinas do sistema como *psdanger* , *psmallo.c* e *disclaim* (utilizada no tratamento do sinal **SIGDANGER** para liberar recursos de paginação, memória, etc.).

Criação de Arquivos de Paginação

- **smit pgsp**
- **smit mkps**
- **smit mklv**
- **mkps**

```
# mkps -a -n -s4 meuvolgrp
# swapon -a
# mkps -t nfs amora.cmp.unicamp.br \
/export/swap/swapclient
```

NOTAS:

O sistema operacional AIX 3.2 quando instalado em qualquer máquina possui um arquivo de paginação inicial, denominado *hd6*.Em máquinas onde o número de usuários executando programas que requerem muita memória é grande, apenas este arquivo de paginação geralmente não é suficiente. Nestes casos surge a necessidade da criação de mais arquivos de paginação no sistema.

Os arquivos de paginação devem ser definidos em discos diferentes, mas pertencentes ao *rootvg*. Recomenda-se não utilizar outro *grupo de volumes* a menos que se esteja familiarizado com uso do sistema operacional.

Geralmente o tamanho de um arquivo de paginação é estimado de acordo com a anÀlise da carga do sistema, mensagens de erro, etc, mas pode-se determinar o tamanho do arquivo utilizando-se o valor de 2 vezes a memÂria *RAM* do sistema.

Cada arquivo criado recebe um nome no padrão *paginxx*, onde xx é um número seqüencial.

Há vários modos de se definir um arquivo de paginação:

- utilizando-se o comando **`smit pgsp`** - o comando smit invoca o menu da opcao pgsp, que mostra uma tela com várias opções, das quais deve-se escolher a opção **`Add Another Paging Space .`** Depois disto, deve-se completar as demais opções a respeito do arquivo de pagina§Ðo a ser criado : qual o grupo de volumes, tamanho do arquivo, qual o disco, etc.
- utilizando-se o comando **`smit mkps`** - o comando smit invoca o menu da opção mkps, mostrando a tela posterior ¥ primeira tela do item acima citado, com uma s¢rie de perguntas que devem ser completadas a respeito do arquivo de pagina§Ðo a ser criado : grupo de volumes, tamanho do arquivo, disco, etc.
- utilizando-se o comando *smit mklv* - que apresenta uma tela onde são defindos os parâmetros para a criação de um volume lógico. Este volume lógico deve possuir as seguintes características :
  - ser do tipo "*paging*".
  - ter a opção *bad-block alocation* não setada.

- não ser espelhado.
- localizar-se no meio do disco - principalmente para performance.

- comando **mkps** -
  sintaxe :
  mkps [-a] [-n] [-t lv] -s *n VolumeGroup [disco]*
  Este comando utiliza o comando **mklv** para criar um volume lógico com as características acima citadas.

Parâmetros:

-s n	Especifica o número de partições lógicas a serem adicionadas ao tamanho do arquivo.
VolumeGroup	Especifica em qual grupo de volumes está o disco onde o arquivo será definido.
-a	Especifica a configuração do arquivo de paginação em todos os boots posteriores.
-n	Ativa o arquivo de paginação imediatamente.
-t lv	Especifica o tipo de arquivo de paginação : lv é do tipo volume lÂgico.
disco	Especifica qual o disco onde o arquivo de paginação será criado.

Existe a possibilidade de se criar um arquivo de paginação para uso via NFS por uma máquina cliente. A sintaxe para esta opção ¢ a seguinte:
mkps [-a] [-n] -t nfs SeverHostname ServerFileName
onde *ServerHostname* éo nome da máquina onde o arquivo reside e *ServerFileName* o nome do arquivo existente no host. Este arquivo deve estar exportado na mÀquina host.

Depois de cada um destes comandos é necessário ativar o arquivo definido com o comando swapon nomedoarquivo, ou pelo comando smit. O arquivo aparecerÀ definido no arquivo /etc/swapspaces, quando o par£metro -a do comando mkps for utilizado.

Manutenção de Arquivos de Paginação

- **chps**
- **lsps**
- **rmps**
- **swapon**

NOTAS:

A manutenção de arquivos de paginação ¢ feita atrav¢s dos seguintes comandos :

- **`smit chps`**
  utilizando-se o comando **`smit chps`** - o comando smit invoca o menu da opcao pgsp, que mostra uma tela com várias opções, das quais deve-se escolher a opção **`Change/Show the Characteristics of a Paging Space .`** Depois disto, deve-se completar as demais opções a respeito do arquivo de pagina§Ðo a ser alterado : qual o grupo de volumes, tamanho do arquivo, qual o disco, etc.
- **`chps`**
  Este comando muda as características de um arquivo de paginação.
  `Sintaxe : chps [-s n] [-a {y|n}] PagingSpace`

Parâmetros:

`PagingSpace` Especifica qual o nome do arquivo de paginação a ser alterado.

`-s n` Especifica o número de partições a serem adicionadas ao tamanho total do arquivo. Para mudar o tamanho de um arquivo de paginação utilizado via NFS, deve-se alterar o tamanho do arquivo na máquina servidora e depois na máquina cliente executar o comando swapon para que a mudança seja devidamente reconhecida.

`-a y` Especifica que o arquivo de paginação será ativado durante os próximos *boots* do sistema.

`-a n` Especifica que o arquivo de paginação não será ativado durante os próximos boots do sistema. Esta opção não consegue desativar o arquivo de paginação default do sistema, pois ele está gravado no registro de boot.

Toda alteração feita pelo comando `chps` é automaticamente ativada.

- **`smit lsps`**

Utilizando-se o comando **smit lsps** - o comando smit invoca o menu da opcao pgsp, que mostra uma tela com várias opções, das quais deve-se escolher a opção **Change/Show the Characteristics of a Paging Space .** Depois disto, deve-se completar as demais opções a respeito do arquivo de pagina§Ðo a ser criado : qual o grupo de volumes, tamanho do arquivo, qual o disco, etc.

- **lsps**

Este comando lista as características de arquivos de paginação.

Sintaxe: **lsps [-c | -1] {-a | -t {lv|nfs} | *PagingSpace***}

Parâmetros:

-a	Especifica que as características de todos os arquivos de paginação do sistema serão listadas. Nesta opção os valores do campo *size*, são mostrados em megabytes.
-c	Especifica que a saída do comando será colunada. Nesta opção o tamanho dos arquivos são expressos em valores de partições físicas.
-l	Especifica que a saída do comando deve ser na forma de colunas.
-t lv	Especifica que somente as características dos arquivos do tipo logical volume serão listadas.
-t nfs	Especifica que somente as características dos arquivos do tipo NFS serão listadas.

- **smit rmps**

O comando smit ativa o menu da opção rmps, que mostra uma tela pedindo o nome do arquivo de paginação a ser removido.

- **rmps**

Este comando remove um arquivo de paginação de todos os volumes lÂgicos nos quais o arquivo reside.

Sintaxe : **rmps *PagingSpace***

onde *PagingSpace* é o nome do arquivo de paginação a ser removido.

Para um sistema usando um arquivo de paginação via NFS, é removido o dispositivo e sua definição. Nada muda no servidor.

OBS: Arquivos de paginação ativos no sistema não podem ser removidos. Deve-se primeiro mudar a característica do arquivo para não estar ativo nos próximos boots, através do comando **chps** e depois dar um reboot no sistema. Quando o sistema estiver novamente no ar, pode-se remover o arquivo.

Os arquivos de paginação cuja entrada está definida no registro de boot do sistema, não são removidos desta forma, há procedimentos especiais para se fazer isto.

- **smit swapon**

Este comando ativa o smit com a opção swapon e mostra uma tela onde se deve definir o arquivo de paginação a ser ativado.

- **swapon**

Este comando ativa arquivos de paginação e swap para um sistema. Ele é utilizado na fase de inicialização do sistema para a ativação dos arquivos básicos de paginação. Numa fase posterior à da inicialização do sistema este comando também é utilizado para ativar os demais arquivos de paginação, que é geralmente feito no arquivo `/etc/rc.`

Sintaxe: **swapon -a | *DeviceName***

Parâmetros:

`-a`	Esta opção faz com que todos os arquivos definidos no arquivo `/etc/swapspaces` sejam ativados no sistema.
`DeviceName`	Especifica qual arquivo deverá ser ativado no sistema.

Redução do arquivo paginação *hd6*

```
# lsattr -E -l sys0 -a realmem
# mkps -a -n -S rootvg
# chps -a n hd6
# vi /sbin/rc.boot
troque a linha swapon /dev/hd6
por swapon /dev/paging00
```

```
# bosboot -d /dev/hdisk0 -a
```

- **mude a chave de boot para a posição NORMAL**

```
# reboot
# rmps hd6
# mklv -t paging -y hd6 rootvg 10
# vi /sbin/rc.boot
```

- **troque a linha** *swapon /dev/paging00* **por** *swapon /dev/hd6*

```
# bosboot -d /dev/hdisk0 -a
# swapon /dev/hd6
# chps -a n paging00
# shutdown -r
# rmps pagin00
```

NOTAS:

A redução do arquivo de paginação *hd6* ¢ interessante para aumentar a performance de arqmazenamento do sistema, pois o disco *hdisk0* ¢ um dos mais cheios por armazenar os sistemas de arquivos /(root) e `/usr`. Sugere-se entÐo gerar o arquivo *hd6* em outros discos.

Para verificar a distribuição dos sistemas de arquivos e logical volumes nos discos você pode utilizar o comando :

`lspv -l hdiskX`

O tamanho do arquivo reduzido deve ser no mínimo suficiente para armazenar a memória física do sistema, a qual ¢ determinada pelo comando:

`lsattr -E -l sys0 -a realmem`

Lembre-se de que um arquivo de paginação não pode ser removido se o sistema estiver rodando. Por isto, deve-se seguir os seguintes passos :

- criar um arquivo de paginação auxiliar e ativá-lo no próximo boot
- desativar o arquivo *hd6* no prÂximo boot
- depois do boot, deve-se remover o arquivo *hd6* e criar outro arquivo *hd6* com o espa§o reduzido e ativÀ-lo para o prÂximo boot
- e desativar o arquivo de paginação auxiliar no próximo boot.
- rebootar a máquina

O arquivo *hd6* deve ser removido para outro disco pertencente ao prÂprio grupo de volumes **rootvg**, pois uma referência a este arquivo é criada em vários lugares utilizados durante o boot do sistema e utilizados também pelo comando **`getrootfs`** durante o procedimento de manutenção da máquina. Somente os arquivos de paginação armazenados no rootvg estarão ativos nas fases um e dois do processo de boot.Assim se nenhum arquivo de pagina§Ðo estiver neste grupo de volumes a performance do sistema estarÀ seriamente comprometida.

Sugere-se então que o arquivo *hd6* tenha um tamanho pequeno e outros arquivos de pagina§Ðo sejam criados nos outros discos.

Gerenciamento de Memória Virtual

- **permite implementação de:**
  - **espaço de endereçamento virtual dos processos**
  - **compartilhamento de executáveis**
  - **segmentos de memória compartilhados**
  - **arquivos mapeados**
- **processo de mapeamento da memória virtual**
  - **conceito de páginas**
  - **armazenamento em memória física ou disco**
  - **PagIn**
  - **PageOut ou PageSteal**
  - **tipos de segmentos:**
    - **Working Storage**
    - **Persistent Storage**
    - **Client Storage**

NOTAS:

O procedimento de uso de memória virtual ¢ implementado atrav¢s da cria§Ðo de segmentos maiores que a memÂria fÁsica disponÁvel no sistema. Os segmentos sÐo divididos em unidades de tamanho fixo chamadas pÀginas. Cada pÀgina num segmento pode estar na memÂria fÁsica ou pode estar armazenada em disco at¢ que ela seja requisitada pelo sistema. Toda vez que um processo precisa de uma pÀgina que nÐo esteja na memÂria fÁsica, o procedimento de gerenciamento de memÂria coloca esta pÀgina na memÂria fÁsica. Este procedimento ¢ conhecido como PageIn. E quando a quantidade de memÂria fÁsica nÐo estÀ disponÁvel, as pÀginas dos processos que nÐo estÐo sendo utilizadas sÐo escritas no disco. Este procedimento ¢ conhecido como PageOut ou PageSteal.

Os segmentos de memÂria podem ser dos seguintes tipos:

Working Storage - São os segmentos usados para implementar as Àreas de dados dos processos e os segmentos de memÂria compartilhada. As pÀginas deste segmentos sÐo armazenadas nos arquivos de pagina§Ðo configurados no sistema.

Persistent Storage - São segmentos usados para manipular arquivos e diretÂrios. Quando este tipo de segmento ¢ acessado , as pÀginas sÐo lidas ou escritas dos files systems onde eles estÐo armazenados.

Client Storage - SÐo segmentos usados para implementar alguns sistemas de arquivos virtuais como NFS e CD-ROM. O armazenamento para segmentos de pÀginas de clientes pode ser local ou remoto.

9

BACKUP RESTORE

Por que fazer backups?

- **É uma das tarefas mais importantes do administrador de sistemas;**
- **Prevenção de catástrofes;**
- **Rápida recuperação em caso de problemas de software, humanos e operacionais;**
- **Seguro de vida do administrador.**

NOTAS:

Uma vez que os seus sistemas estejam em uso, a primeira coisa que o administrador deve pensar é na segurança de seus dados. Todos os arquivos e diretórios de um sistema representam um investimento considerável em termos de tempo e dinheiro e, como tal, devem ser adequadamente protegidos contra eventualidades (problemas de hardware, software, falhas humanas, etc.). É uma das tarefas mais importantes, se não a mais importante, do administrador de sistemas.

Métodos de backup

- **backup**
- **cpio**
- **dd**
- **tar**
- **rdump**
- **pax**

NOTAS:

Existem vários programas que o sistema operacional fornece para o backup dos dados. O programa mais freq¡entemente usado para esta finalidade ¢ o programa `backup`. O programa `backup` pode efetuar cópias de sistemas de arquivos, arquivos individuais, diretórios. O programa `backup` oferece como uma de suas opções a realização destes dois tipos de backups (totais ou incrementais). Os demais programas nÐo oferecem esta facilidade, a qual somente pode ser obtida atrav¢s da combina§Ðo com outros programas do sistema Unix (comando `find` , por exemplo).

Meios de armazenamento

- **Disquete 3,5"**
- **QIC-150**
- **Fita 8mm/4mm**
- **Fitas de rolo**

NOTAS:

Existem vários meios diferentes para armazenamento de backups. Os mais difundidos hoje sÐo os disquetes, fitas de rolo, fitas do tipo QIC-150, e fitas de 8mm e 4mm.

Os disquetes são mais usados hoje em dia para backups de arquivos de usuÀrios individuais. A baixa capacidade desaconselha o uso deste meio para armazenamento de grandes quantidades de dados. SÐo usados tamb¢m para transfer¨ncia de dados entre esta§Ñes de trabalho e microcomputadores e vice-versa. O interc£mbio de arquivos DOS para sistemas Unix ¢ facilitado por alguns programas, tanto comerciais quanto de domÁnio pÃblico, que facilitam esta tarefa. Os sistemas Unix normalmente oferecem comandos que facilitam a conversÐo do formato Unix para o formato DOS e vice-versa. As unidades de disquetes normalmente gravam com a densidade de 1.44MB e jÀ sÐo oferecidas em alguns sistemas unidades de 2.88MB.

As unidades do tipo QIC-150 possuem capacidade de armazenamento de 150MB e ainda são bastante comuns. Possuem como pontos negativos o alto preço e a lentidão para armazenamento e recuperação de dados.

As fitas de 8mm e 4mm estão começando a dominar o mercado. São rápidas e podem armazenar grande quantidade de dados. As fitas 4mm de 60 metros podem armazenar em média 1.2GB. As fitas 8mm normalmente usadas em sistemas AIX podem armazenar até 2.5GB de dados. Com o uso de compressão este valor é ainda mais elevado. Este tipo de unidade é o que oferece mais vantagens ao administrador:

- A tarefa de execução de backups fica menos tediosa;
- O backup de todas as mÀquinas da rede pode ser feito em uma sÂ unidade sem a necessidade de troca constante de fitas;
- Pode ser executado sem a supervisão do operador. Tudo que é necessário é a verificação do resultado no dia seguinte;
- O backup pode ser executado em horas de pouca ou nenhuma atividade do sistema, garantindo desta forma a integridade dos dados.

NOTAS:(Continuação)

As fitas de rolo (9 track tape), não são encontradas com freqüência. Podem ser usadas para transferência de dados entre mainframes de diversos fabricantes que não estejam interligados via rede. Podem ler fitas gravadas nas densidades de 800, 1600 e 6250 BPI. Com a densidade de gravação de 6250 BPI podem armazenar, em uma fita de 2400 pés, até 150 MB. São bastante rápidas.

RESTORE

Existem várias maneiras de restaurar dados:

- **restore**
- **rrestore**
- **cpio**
- **tar**
- **dd**

NOTAS:

Uma vez que os dados tenham sido armazenados adequadamente existem várias formas de restaurá-los. Esta forma irá depeder da maneira com que os dados foram arquivados originalmente.

Você precisa saber como os seus dados foram arquivados de modo a poder restaurá-los. Por exemplo, ao usar o comando `backup` você pode especificar que o mesmo seja executado pelos i-nodes ou por nome. Em conseqüência, a restauração dos arquivos tem que ser feita do mesmo modo, ou seja, pelos i-nodes ou pelos nomes.

O programa `restore` é usado para restaurar dados criados pelo programa `backup`.

Estratégias para backup

- **Capacidade de recuperação de desastres;**
- **Verifique os seus backups periodicamente;**
- **Guarde os backups antigos;**
- **Verifique os sistemas de arquivos antes de efetuar o backup;**
- **Faça os backups em horÀrios de pouca ou nenhuma atividade no sistema;**
- **Sempre faça backups do sistema antes de efetuar altera§Ñes substanciais**

NOTAS:

Nenhuma estratégia de backup atende a todos os sistemas. Uma estrat¢gia que ¢ adequada para sistemas com um usuÀrio pode ser imprÂpria para sistemas que atendem dez ou mais usuÀrios. Da mesma forma, uma estrat¢gia adequada para um sistema em que os arquivos sÐo modificados freq¡entemente nÐo se adequa a um sistema em que tais altera§Ñes sÐo raras. Apenas o administrador pode determinar com precisÐo a estrat¢gia que melhor se adequa a cada situa§Ðo. Na escolha de uma estrat¢gia de backup tente levar em considera§Ðo os seguintes fatores:

- Capacidade de recuperação em caso de crash total do sistema:
  - Você consegue recuperar o seu sistema se um disco quebrar? Você conseguirá recuperar o seu sistema se TODOS os discos quebrarem? E se tudo pegar fogo, inclusive os backups? Embora isto seja quase impossÁvel, estes fatores devem ser considerados quando da defini§Ðo da estrat¢gia de backup.
- Verifique os seus backups periodicamente:
  - O meio de armazenamento pode não ser totalmente confiável. Um conjunto de fitas ou disquetes muito grande é totalmente inútil se os dados neles contidos não puderem ser restaurados. Para certificar-se de que os dados em uma fita podem ser lidos use, faça periodicamente a verificação dos mesmo (usando, por exemplo, os comandos `tar -t` ou `restore -T`). Se você usa fitas streamer, você pode usar o comando `tapechk` para executar uma verificação rudimentar da integridade da fita.
- Estabeleça uma política de retenção de fitas
  - Determine um ciclo para reutilização de fitas. Você não deve, entretanto, reutilizar todas as suas fitas. Às vezes se transcorrem meses antes que você ou mesmo algum usuário sinta a necessidade de restaurar algum arquivo importante que tenha sido apagado por engano. Devido a isto backups antigos, dentro de certos limites, devem ser mantidos. Existem vÀrias formas de se fazer isto, que irão depender em grande parte dos recursos, das peculiaridades e das necessidades de cada instalação.

NOTAS:

- Verifique os sistemas de arquivos antes de cada backup:
  - Um backup efetuado a partir de um sistema de arquivos corrompido pode ser inÃtil. Antes de efetuar backups ¢ aconselhÀvel verificar a integridade dos sistemas de arquivos usando o comando `fsck`.
- Faça backups em horÀrios em que o sistema se encontre em estado de mÁnima (ou nenhuma) atividade.
- Faça um backup antes de efetuar altera§Ñes substanciais no sistema
  - É sempre aconselhável fazer um backup antes de efetuar mudan§as de porte no sistema operacional, instala§Ðo de corre§Ñes, mudan§as significativas em programas aplicativos, enfim, tudo o que possa representar uma amea§a ao funcionamento normal do sistema. Em caso de problemas o backup significa a volta a um status original em que tudo estava funcionando a contento.

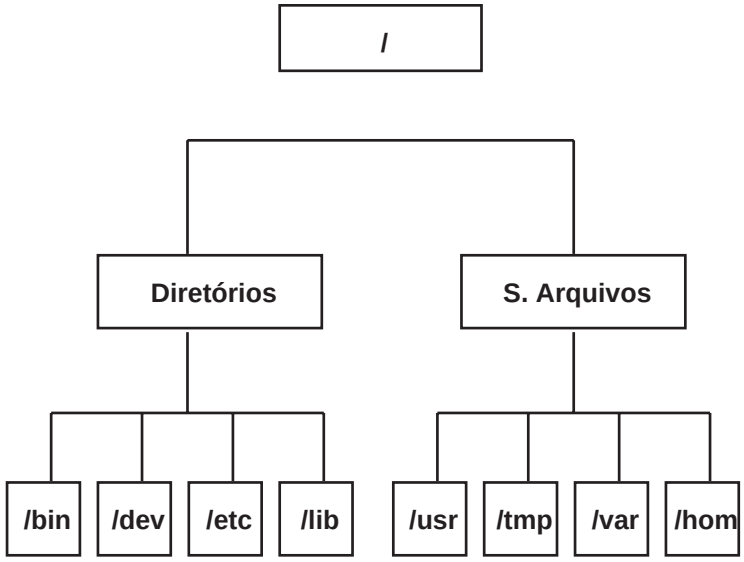
Como desenvolver uma estratégia de backups

Existem duas maneiras de se efetuar backups:

- **totais**
- **incrementais**

NOTAS:

Estrutura dos Sistemas de Arquivos



NOTAS:

É importante entender a diferença entre um sistema de arquivos e um diretório. Um sistema de arquivos é uma seção do disco rígido que foi alocada para abrigar arquivos. Esta parte do disco é acessada montando-se um sistema de arquivos sobre um diretório. Uma vez montado o sistema de arquivos, ele se apresenta ao usuário final como qualquer outro diretório. Devido às diferenças estruturais entre os sistemas de arquivos e diretórios, os dados dentro destas entidades podem ser gerenciados separadamente. A maior parte dos diretórios sobre os quais são montados os sistemas de arquivos são todos criados no sistema de arquivos *root (root filesystem).*

Tipos de Dados

- **Dados do sistema**

- **Dados de usuários**

NOTAS:

Os dados (programas ou texto), podem ser divididos em duas categorias:

- Dados do sistema:
  Compreendem o sistema operacional e suas extensões. Estes dados devem sempre ser mantidos nos sistemas de arquivos do sistema operacional (`/ (root), /usr, /tmp, /var,` etc).
- Dados dos usuários:
  São os dados que os usuários necessitam para desempenhar suas tarefas. Estes dados são normalmente mantidos no sistema de arquivos `/home` ou em sistemas de arquivos especificamente criados para esta finalidade.

Se os dados dos usuários são mantidos em sistemas de arquivos separados, a tarefa de gerenciar backups fica mais fÀcil. Normalmente o backup dos dados dos usuÀrios ¢ feito separadamente dos backups dos dados do sistema por duas razÑes:

- Os dados dos usuários são alterados com mais freqüência do que os dados do sistema. A imagem do backup dos dados dos usuÀrios ¢ tamb¢m muito menor do que a dos dados do sistema.

É mais fácil e rápido restaurar os dados dos usuários quando estes são mantidos separados. A restauração do sistema operacional juntamente com os dados dos usuários requer um tempo e esforço consideráveis. A razão é que o método usado para recuperar o sistema operacional requer um boot do sistema a partir de fita, disquete, ou um outro meio e a instalação do backup do sistema.

backup
(1)

- **smit backup**
  - **indicado para backup de sistemas de arquivos pequenos;**
  - **Não faz backups incrementais.**
- **comando backup**
  - **indicado para o backup de sistemas de arquivos grandes atrav¢s dos i-nodes;**
  - **usado para fazer backups de diferentes nÁveis.**
- **smit mksysb**
  - **indicado para fazer cópia total (cloning) do sistema (rootvg);**
  - **gera fita de boot do sistema.**

NOTAS:

O comando `smit backup` invoca o **smit** com a opção `backup`. Uma tela deve ser preenchida pelo usuário com as informações sobre o sistema a ser armazenado e as opções para o comando `backup`. Este comando é recomendado para o backup de sistemas de arquivos pequenos (`/home`, por exemplo). Backups incrementais nÐo podem ser realizados atrav¢s deste comando.

O comando `backup`, emitido a partir da linha de comandos é indicado para o armazenamento de sistemas de arquivos grandes feitos através dos i-nodes.

O comando `backup` cria cópias de dados em dois formatos; pelos nomes ou pelos i-nodes. Os backups executados pelos i-nodes apresentam a vantagem de que quando os dados sÐo restaurados ¢ feita uma desfragmenta§Ðo do disco, visto que os i-nodes sÐo alocados seq¡encialmente para a cria§Ðo dos blocos de dados dos arquivos.
OBS.: os backups efetuados pelos i-nodes nÐo funcionam corretamente para arquivos cujos UIDs ou GIDs sejam maiores que 65535. Estes arquivos sÐo armazenados com estes valores truncados e, conseq¡entemente, terÐo seus GID ou UID errados ao serem restaurados. O backup por nome funciona corretamente nestes casos.

backup (2)

Para backups a partir da linha de comando:

- **Certifique-se que o dispositivo de backup estÀ online e disponÁvel:**

```
# lsdev -C | pg
```

- **Desmonte os sistemas de arquivos que deseja armazenar:**

```
# umount /home /var
```

- **Rode o comando `fsck`:**

```
fsck /home /var
```

NOTAS:

Ao fazer backups siga os seguintes passos:

- Verifique se o dispositivo utilizado para o `backup` esta online e disponÁvel com o comando `lsdev -C | pg`. Este comando irÀ gerar uma listagem com todos os dispositivos configurados para o sistema. Procure a linha com a identifica§Ðo da unidade de backup e confirme que tudo estÀ correto. Caso contrÀrio, ligue ou conecte o dispositivo, conforme o caso e, da conta *root*, emita o comando `mkdev -l 'rmtx'`, onde *x* representa o dispositivo específico que se quer configurar. O comando `cfgmgr` também pode ser usado com o mesmo fim. Este comando é mais genérico e configura todos os dispositivos acrescentados à máquina após o boot do sistema.
- Desmonte todos os sistemas de arquivos que se deseja armazenar. Para isto use o comando `umount all` ou `umount /home /var`, por exemplo. O primeiro comando irá desmontar todos os sistemas de arquivos que não possuam o atributo `mount=automatic` definido no arquivo `/etc/filesystems`. Esta prática, embora altamente recomendável, é difícil de se conseguir, ou seja, o desmonte dos sistemas de arquivos para a realização de backups se revela inviÀvel na maior parte dos casos. Quando se faz backups de sistemas de arquivos montados o sistema emite uma mensagem de aviso e prossegue com a execu§Ðo. NÐo ¢ uma prÀtica 100% segura mas sÐo rarÁssimos os casos em que um backup esteja totalmente corrompido por ter sido feito com o sistema de arquivos montado. Mas sempre tente fazer os seus backups em horÀrios de pouca atividade, para diminuir os riscos.
- Rode o comando `fsck` para fazer a consistência dos sistemas de arquivos a serem armazenados. A existência de sistemas de arquivos inconsistentes em sistemas AIX é muito difícil de ocorrer devido a implementação de *journalled file systems* (jfs) implementada pela IBM. Mas é sempre recomendável executar este comando (ver a Lei de Murphy para a Informática).

backup (3)

- **Execute o comando backup**:

```
backup -5 -upf /dev/rmt0 /home
```

- **Verifique a integridade da fita**

```
restore -T /dev/rmt0
```

NOTAS:

- Execute o comando `restore -t` para verificar a integridade da fita. Caso apareçam mensagens de erro substituir a fita e repetir o `backup`.

restore (1)

- **Exibição do conteúdo da fita:**

```
# restore -Tf /dev/rmt0
```

- **Interativo:**

```
# restore -ivf /dev/rmt0.1 -s 4
```

NOTAS:

O comando `restore -T` exibe o conteúdo da fita. É útil quando se pretende restaurar apenas determinados arquivos, ou mesmo determinar se o arquivo desejado se encontra na fita. A flag **-f** é usada para informar ao comando `restore` a unidade a ser utilizada. Se não for informado este valor, é assumido o default, o dispositivo `/dev/rfd0.`

Muito útil é a opção de restore interativo.Esta opção permite que se selecione os arquivos dinamicamente. Ao se invocar o comando restore com a flag -i o conteúdo da fita é lido e é exibido ao usuário o seguinte prompt:

```
restore>
```

Este comando aceita as seguintes opções:

ls	lista o conteúdo do diretório. Nomes de diretórios são exibidos juntamente com caracter **/** e nomes de arquivos são exibidos juntamente com um *****.
cd	muda o diretório corrente
pwd	exibe o nome do diretório corrente
add [nome]	acrescenta o nome do arquivo ou diretório nomeado à lista dos arquivos a serem restaurados.
delete [nome]	remove o arquivo ou diretório nomeado da lista de arquivos a serem restaurados.
extract	indica que todos os arquivos já foram selecionados (marcados com ***** antes do nome) devem ser restaurados
quit	indica o final da sessão de **restore**

No exemplo ao lado foi usada também a flag **-s**. Esta flag indica que a fita contém mais de um arquivo de `backup` e que deve-se se selecionar o quarto arquivo. O dispositivo no-rewind só pode ser usado quando se especificar a flag **-s.**

restore (2)

- **Restore total:**

```
# restore -rvf /dev/rmt0
```

- **Restore de arquivos selecionados:**

```
# restore -xvf /dev/rmt0 /u/home
```

NOTAS:

A restauração de um sistema de arquivos completo é executada quando se especifica a flag **-r**. É aconselhável se recriar o sistema de arquivos quando se faz o restore total (comando **mkfs**). Para realizar um restore incremental, restaure primeiramente o backup nÁvel 0, e em seguida os backups incrementais. ° criado um arquivo denominado `restoresymtable` no sistema de arquivos sendo restaurado. Este arquivo é necessário para que o comando `restore` realize a restauração incremental de sistemas de arquivos.**NÃO REMOVER ESTE ARQUIVO ATÉ QUE O RESTORE TENHA TERMINADO.**

Para restaurar apenas determinados arquivos utilizar a opção **-x** [nome arquivos/diretório]. Se não forem especificados os nomes dos arquivos, é feito um restore de todos os arquivos contidos na fita.

Para realizar o backup dos dados do sistema, desmonte todos os sistemas de arquivos de usuários, inclusive o **/home**, usando o comando **unmount**. Se estes sistemas de arquivos estiverem sendo utiliizados, você não conseguirá desmontá-los. É aconselhável fazer os backups em horÀrios de baixa utiliza§Ðo do sistema de forma a que os sistemas de arquivos possam ser desmontados; se os sistemas de arquivos dos usuÀrios permanecerem montados eles serÐo armazenados juntamente com os dados do sistema

operacional. Para se certificar de que apenas os sistemas de arquivos do sistema operacional estão montados emita o comando:

mount

Os únicos sistemas de arquivos montados devem ser **/usr**, **/var** e **/tmp** e o resultado do comando mount deve ser:

```
node  mounted     mounted over  date        vfs  options

      /dev/hd4      /           Jun 1 1:45  jfs  rw,log=/dev/hd
                                                 8

      /dev/hd2      /usr        Jun 1 1:45  jfs  rw,log=/dev/hd
                                                 8

      /dev/hd9var   /var        Jun 1 1:45  jfs  rw,log=/dev/hd
                                                 8

      /dev/hd       /tmp        Jun 1 1:45  jfs  rw,log=/dev/hd
                                                 8
```

Emita a seguir o comando mkszfile. Este comando criará no diretório / um arquivo de nome .fs.size. Como o próprio nome diz, este arquivo contém a lista dos sistemas de arquivos montados quando da execução do comando e seus tamanhos. Este arquivo é gravado junto com o backup do sistema e ¢ utilizado durante o restore para fazer a aloca§Ðo dos sistemas de arquivos. Finalmente, ao fim do backup do sistema, montar novamente os sistemas de arquivos que cont¨m os dados dos usuÀrios (/home) e fazer o backup destes com o comando backup.Este backup deve ser executado diariamente ou de acordo com a taxa de atualização de seus arquivos.

Duplicação de Sistemas (Cloning)

- Cópia do sistema inteiro para outras máquinas;
- Em caso de contingências (incêndios, falhas totais de hardware, etc.), a fita backup pode ser levada para outro equipamento sem prejuÁzo das atividades;
- Os dados do sistema são duplicados juntamente com os dados dos usuários;
- Apenas os sistemas de arquivos do grupo de volumes rootvg são armazenados;

NOTAS:

Existem casos em que se faz necessário o `backup` dos dados dos usuÀrios juntamente com os dados do sistema. Este procedimento ¢ tamb¢m conhecido por *cloning***.** Este processo é similar ao descrito na página anterior com a diferença que os sistemas de arquivos contendo dados de usuários não são desmontados.

A fita criada com o comando `mksysb` é usada para dar o boot e o usuário é orientado, por meio de menus, sobre os passos a tomar. Os clones dos sistemas podem ter vários usos:

- Instalação do sistema operacional com características semelhantes em várias máquinas;
- Recuperação rápida em caso de contingências;
- Retorno a um nível estável do sistema operacional após falhas em um processo de upgrade ou de instalação de novas versões;

Apenas os sistemas de arquivos pertencentes ao grupo de volume *rootvg* são armazenados. Os demais sistemas de arquivos devem ser armazenados usando-se o comando `backup`.

Outras Considerações Importantes

- **Qual a freqüência de atualização dos dados?**
- **Qual o número de usuários do sistema?**
- **Quais as dificuldades envolvidas na recriação dos dados?**

NOTAS:

Os dados do sistema operacional não mudam constantemente e, conseqüentemente não precisam ser salvos com freqüência. Os dados dos usuários, entretanto, são alterados quase diariamente e como tal, um cuidado especial deve ser tomado no backup de seus dados.

O número de usuários é outro fator a ser considerado. Este número influi na quantidade de dados modificada e armazenada, o número de fitas (ou qualquer outro meio exigido para o backup) e a freq¡¨ncia com que tempo se farÐo os backups.

Finalmente, caso se necessite recriar os dados, quanto tempo esta tarefa exigirá? É importante levar em conta que certos dados não poderão ser recriados sem a existência de um backup.

Qualquer que seja a estratégia de backup escolhida, ¢ muito importante que ela exista. Os backups devem ser feitos em bases regulares e freq¡entes. A recupera§Ðo em caso de perdas de dados ¢ muitomais difÁcil se nÐo existir uma boa estrat¢gia de backups implementada.

10

Ambiente Operacional

Ambiente Operacional

Conjunto de variáveis que definem ou controlam determinados aspectos da operação do sistema.

Profiles

Arquivos utilizados pela shell para definição do ambiente operacional:

- **/etc/profile**
- **.profile**

NOTAS:

O ambiente operacional é primariamente o conjunto de variáveis que definem ou controlam certas características da operação do sistema. A maior parte destas variáveis é definida durante a inicialização do sistema, sendo que suas definições são lidas do arquivo `/etc/profile` ou definidas por default.

A shell utiliza dois tipos de arquivos de inicialização quando um usuário loga no sistema. Ela avalia os comandos contidos nestes arquivos e então os executa para definir o ambiente operacional. Os arquivos possuem funções similares, com a exceção de que o arquivo `/etc/profile` controla variáveis aplicáveis a todos os usuários ao passo que o arquivo `.profile` permite a customização individual do ambiente.

O primeiro arquivo que o sistema lê durante o login de um usuário é o arquivo `/etc/profile`. Este arquivo controla variáveis tais como:

- variáveis exportáveis
- máscara de criação de arquivos
- tipos de terminal
- mensagens utilizadas para informar o usuário da chegada de novas mensagens de mail.

O administrador do sisema configura o arquivo `/etc/profile` para todos os usuários do sistema. Este arquivo tem acesso permitido apenas para o usuário *root.*

O segundo arquivo que o sistema operacional usa durante o login é o arquivo `.profile` encontrado no diretório *home* do usuário. Este arquivo permite que cada usuário customize o seu ambiente de acordo com suas preferências pessoais. Os comandos contidos neste arquivo suplantam as definições estabelecidas pelo arquivo `/etc/profile`. Entre outras coisas o arquivo `.profile` é usado para:

- definir as shells a serem usadas
- aparência do prompt
- variáveis de ambiente ($PATH, por exemplo)

Shells

- Programa
- Aceita e executa comandos
- Linguagem de programação
- Poderosa e flexível
- Disponíveis com AIX 3.2
  - Bourne Shell
  - C-shell
  - Korn Shell
  - Restrita
  - Trusted

NOTAS:

A shell é a interface entre o sistema e o usuário. É um programa que aceita e executa comandos.

Comandos de shell normalmente executam uma determinada tarefa muito bem. A capacidade de combinar estes comandos para situações específicas dá ao programador um grande controle sobre o sistema.

O sistema AIX 3.2 vem com a Bourne shell, C-Shell, Korn Shell, shell restrita e trusted shell. O usuário pode escolher a shell a que melhor se adapte.

Bourne Shell

- **/bin/bsh**
- **$**
- **Desenvolvida nos Laboratórios Bell**
- **Amplamente utilizada na indústria**
- **Default para sistemas AIX anteriores à versão 3**

C Shell

- **/bin/csh**
- **%**
- **Desenvolvida em Berkeley**
- **Sintaxe semelhante à linguagem C**
- **Possui muitos comandos e variáveis úteis**

NOTAS:

A Bourne Shell é uma shell amplamente utilizada desenvolvida nos laboratórios Bell e recebeu o nome de seu criador.

Como o sistema Unix foi escrito em linguagem C e como era frequentemente utilizado por programadores C, era de se esperar que se criasse uma shell que aceitasse comandos com sintaxe semelhante à da linguagem C.

A C Shell possui funções que não estão disponíveis na Bourne Shell, como por exemplo um histórico dos comandos emitidos com resubmissão e uma função para criação de comandos customizados.

Korn Shell

- **/bin/ksh**
- **$**
- **Possui a sintaxe e as funções da Bourne Shell**
- **Funções da linguagem C**
- **aliases**
- **Controle de tarefas**
- **Histórico de comandos**
- **Edição da linha de comandos**
- **Arquivo de ambiente**

NOTAS:

A Korn shell combina a sintaxe da Bourne Shell com funções adicionais, tal como aliases e histórico de comandos, à semelhança da C Shell.

Os aliases são uma maneira de se criar comandos customizados. O controle de tarefas fornece a facilidade de se suspender e de se retomar tarefas, colocar tarefas em *background* e *foreground*.

O histórico dos comandos executados é mantido em um arquivo designado pela variável **HISTFILE** (default `$HOME/.sh_history`). O tamanho do arquivo é determinado pela variável **HISTSIZE** (default é 128 comandos). A variável **FCEDIT** é usada para definir o editor utilizado para editar comandos. O alias **r** é utilizado para reexecutar comandos.

A edição da linha de comandos pode ser feita usando ou o editor vi ou emacs. "`set -o vi`"ou "`set -o emacs`" especificam qual editor será usado.

Quando as variáveis são definidas, elas estão disponíveis apenas para o processo no qual elas foram definidas a menos que elas sejam exportadas. Da mesma forma, aliases e funções somente estão disponíveis no processo no qual elas foram criadas a menos que algo especial seja feito. De forma a torná-las disponíveis em chamadas subsequentes da shell, coloque definições de aliases e funções em um arquivo, defina a variável **ENV** com o nome do arquivo criado e exporte a variável **ENV.**

Restricted Shell

- /bin/Rsh
- $
- Idêntica à bsh, mas não pode:
  - executar o comando cd
  - set PATH=*valor*
  - Especificar comandos que contenham "/"
  - Redirecionar output (> ou >>)

Trusted Shell

- /bin/tsh
- CTRL-x CTRL-r
- tsh>
- Executa programas/comandos confiáveis
- Subgrupo dos comandos da ksh
- Invocada com SAK (secure attention key)

NOTAS:

As restrições impostas tornam-se efetivas após a execução do arquivo .profile. Isto significa que se um usuário consegue alterar o seu arquivo .profile, ele pode usar funções restritas, como por exemplo o comando cd, de dentro de seu arquivo .profile, logout e login e isto significa que o usuário conseguiu eliminar as restrições. A alteração das permissões de gravação do arquivo não é suficiente, pois o usuário pode apagar seu .profile e criar um outro à sua conveniência. O usuário deve ser colocado em um diretório diferente quando do login.

Normalmente, a shell restrita é invocada fazendo com que o programa seja executado como a shell default do usuário.

A Trusted Shell fornece um ambiente confiável no qual executar tarefas administrativas sem correr o risco de executar programas que possam comprometer a segurança do sistema.

11

Instalação

Roteiro de Instalação

Pode ser feita a partir de:

- **CDROM**
- **Fita**
- **Disquete**
- **Rede**
  - **Servidor de instalação**
- **Backup do sistema**

Tipos de instalação

- *New Installation*
- *Preservation Install*
- *Complete Overwrite Install*

NOTAS:

Roteiro de Instalação (2)

- Obtenha informações da máquina e da rede:
  - Nome da máquina
  - Dispositivo a ser usado para instalação
  - Endereço IP
  - Endereço do servidor
  - Endereço do gateway
  - Conectores de rede (ethernet ou Token-Ring)
  - Máscara de subrede
  - Nameserver
  - Nome do domínio
  - Arquivos montados via NFS

NOTAS:

Roteiro de Instalação (2)

- **Configure o terminal**
  - **Terminal ASCII:**
    Comunicação
    9600 baud
    8 bits
    Sem paridade
    Stop Bits = 1
    Line Control = IPRTS
    Teclado
    Screen = normal
    Row and Column = 24 x 80
    Scroll = jump
    Auto LF = off
    Line Wrap = on
    Forcing insert = line
    Tab = Field
    Operating Mode = echo
    Turnaround character = CR
    Enter = Return
    Return = new line
    New Line = CR
    Send = Page
    Insert Character = Space

NOTAS:

Se o terminal for do tipo IBM 3151, 3161 ou 3164, pressione simultaneamente as teclas CTRL+SETUP para chamar o menu de setup do terminal. Se estiver utilizando algum outro tipo de terminal siga as instruções do fabricante para configurá-lo corretamente.

Roteiro de Instalação (3)

- Ligue a estação
- Coloque a chave na posição `secure`
- Espere que apareça o número 200 no indicador do LED
- Insira na unidade a mídia de instalação (CDROM, fita ou disquete)
- Vire a chave para a posição `Service`
- Aperte rapidamente duas vezes o botão de reset (botão amarelo)
- Selecione a console

NOTAS:

Roteiro de Instalação (4)

```
       AIX 3.2 INSTALLATION AND MAINTENANCE

Select the number of the task you want to perform

>>> 1. Install AIX
    2. Install a system that was created with the SMIT
       “Back Up the System”function or the “mksysb
       command
    3. Install this system for use with a ”/usr"server
    4. Start a limited maintenance shell


Type the number for your selection, then press “Enter”:1
```

```
                  METHOD OF INSTALL

Select the number of the type of Installation you want
to perform:

1  PRESERVATION INSTALL
   Preserves SOME of the data on the destination hard
   disk. Only overwrites the user (/usr), temporary
   (/tmp), and root (/) file systems of the previously
   installed version of AIX

2  COMPLETE OVERWRITE INSTALL
   May overwrite EVERYTHING on the destination hard disk
   - If the destination disk is totally empty, select 2
   - If AIX is already installed on the destination hard
     disk but there is nothing on the disk you want to
     preserve, select 2.

99 Return to previous menu

Type the number for your selection, then press Enter: 1
```

NOTAS:

12

Processos

Visão Geral

- **Programas ou comandos executando no computador**
- **Hierarquia pai-filho**
- **pid - multiprocessamento - scheduler**
- **foreground - background**
- **daemons**
- **zombie**

NOTAS:

O que é um processo? É um programa ou comando que está sendo executado pelo computador, o qual pode executar mais de um processo ao mesmo tempo (multiprocessamento).

A criação de processos obedece à hierarquia pai-filho. Assim todo processo pai pode ter mais que um processo filho, e todo processo filho possui apenas um pai. O processo pai é aquele que foi criado a partir de um programa ou comando. E o processo filho é aquele criado por um processo do tipo pai.

Por ser um sistema de multiprocessamento, o sistema associa um número a cada processo (***pid***), que é diferente de todos os números já associados a processos durante o tempo relativo da entrada da máquina no ar até o momento de sua queda. Cada processo utiliza uma determinada quantidade de tempo disponível do sistema, de acordo com uma potilítica de escalonamento feita pelo *scheduler,* e também de acordo com as prioridades dos processos.

Processos considerados *foreground*, são aqueles criados através da linha de comando e que precisam de interação com o usuário. Durante a sua execução nenhum outro comando pode ser executado na linha de comando. Processos background são aqueles que executam independentemente da criação pela linha de comando, sem a interação com o usuário.

Daemons são processos criados no momento de boot da máquina e que continuam executando até o encerramento do sistema. Estes tipos de processos executam serviços do sistema e estão disponíveis sempre para mais de um usuário ou tarefa. Eles somente podem ser criados ou parados pelo usuário root.

Zombies são processos “stopped” mas que ainda são reconhecidos pela tabela de processos. Este tipo de processo não utiliza nenhum recurso do sistema operacional. Eles são liberados somente quando o processo pai também é “stopped”, ou o sistema reinicializado.

Comandos úteis (1)

- **linha de comando**
- **&**

```
$ find . -name telnet -print &
$ find . -name x -print
^Z
$ bg/jobs/kill %njob
$ %numjob
```

- **nice, renice**

```
$nice -n 15 cc -c *.c &
$renice +1 87 -u daemon root -p
32
```

NOTAS:

Para criar um processo (*foreground*) basta executar um comando através da linha de comando, como por exemplo executar um programa.E para criar um processo *background,* basta executar um comando seguido do símbolo &. Os processos deste tipo geralmente são aqueles que executam um trabalho muito pesado no sistema, podendo comprometer a sua performance. Por isto, estes processos devem executar com prioridades mais baixas.

O comando `nice` executa um comando com prioridade diferente da normal. Ele é usado no momento do "start" de um programa. Um processo só pode ter sua prioridade aumentada pelo usuário root.

Sintaxe: `nice [- Number | -n Number]` *comando*

Parâmetros :

- *Number*	Especifica valores de prioridades que o sistema pode dar a um determinado processo. Na versão System V do Unix, estes valores ficam entre 0 a 39, e na versão BSD os valores são de -20 a 20. O número 20 não tem valor correspondente no range dos valores da versão BSD. A prioridade mais alta é a de valor 0 (System V) ou -20 (BSD). Se este parâmetro não for especificado, a prioridade do processo será diminuida.
-n *Number*	Especifica os mesmos valores que o -*Number*.
comando	Defini-se aqui o comando a ser executado, com todos os seus parâmetros.

O comando ***renice*** também é utilizado para alterar prioridades de processos, mas que já estão executando no sistema. Os processos só podem ter sua prioridade alterada para uma prioridade maior pelo usuário root.
Sintaxe : **renice** *Priority* ***[-g GroupID ...] -p PID ...] [-u*** *UserName****...]***

Comandos Úteis (2)

- **ps**

```
$ ps -ef
$ ps -flu root,server
$ ps -t
$ smit process
```

- **kill**

```
$ps -ef | grep ze
$ kill  -9 1975
$ smit kill
```

- **killall**

```
$ killall -
$ killall -2
```

NOTAS(cont):

Um usuário normal consegue alterar somente a prioridadade de processos dos quais seja dono e entre os valores 0 a 20 (ou 20 a 39 System V).

Parâmetros:

Priority	Especifica a prioridade a ser dada a determinado processo.
-g *GroupID*	Especifica o número do grupo do processo a ser alterado.
-p *PID*	Especifica o número do processo (***PID***) a ser alterado.
-u *Username*	Especifica o username que é o proprietário do processo.

O comando **ps** é utilizado para listar informações sobre os processos que estão executando no sistema.
Sintaxe:

ps [-A] [-a] [-d] [-e] [-f] [-k] [-l] [-F *Format*] [-G *Glist]* -[g *Glist*] [-p *Plist]*] [-t *Tlist]* [-U *Ulis*t] [-u *Ulist*]
ps [a] [c] [e] [ew] [eww] [g] [l] [n] [s] [t Tty] [U] [u] [v] [w] [x] [*ProcessNumber*]

Parâmetros:

Os parâmetros **-f,-F,l,-l,s,u,v** determinam o tipo de informação a ser exibida sobre um processo: eles não determinam que processos são listados. Estes parâmetros são mutuamente exclusivos.

Com o flag **-f**, o comando **ps** determina onde o nome e os parâmetros do comando estavam quando o processo foi criado examinando a memória e a área de paginação. Se o comando ps não achar esta informação, o nome do comando armazenado no kernel é mostrado em colchetes.

s Lista os processos mostrando a informação sobre o estado em que eles se encontram. Os estados podem ser dos seguintes tipos:

- 0 - Nonexistent
- S - Sleeping
- R - Running
- I - Intermediate
- Z - Canceled
- T - Stopped
- K - Available Kernel process
- X - Growing

c Lista o nome do comando, conforme armazenado internamente no sistema para propósitos de account, ao invés dos parâmetros, os quais são armazenados no espaço de endereçamento do processo.

-e Lista informações sobre todos os processos do sistema exceto os do kernel.

-f Gera uma lista completa de informações, com os seguintes campos : USER, PPID, C, STIME, TIME, COMMAND

g Lista todos os processos do sistema.

-k Lista somente processos do kernel.

-l Gera uma lista com os campos PPID, C, PRI, NI, ADDR, SZ e WCHAN.

-u,U *Ulist* Gera uma lista de informações sobre os processos pertencentes aos usernames definidos em ***Ulist.***

-p *Plist* Gera uma lista de informações sobre os processos cujos PID's foram especificados em ***Plist.***

Para parar a execução de um processo executando no modo ***foreground*** utilize <CTRL>C.Para parar a execução de determinado processo em ***background*** usa-se o comando **kill**.

Este comando envia para o processo especificado um sinal do tipo SIGTERM, e todos os processos que ignoram ou que não receberam este sinal são encerrados (stopped).

Um usuário normal pode encerrar somente os processos que foram iniciados por ele mesmo. O usuário **root** pode encerrar qualquer processo.

Sintaxe:

`kill [-s {`*SignalName |SignalNumber}] PID ...*

`kill [ -`*SignalName | -SignalNumber] PID ...*

Para listar os nomes dos sinais :

`kill -l [`*ExitStatus]*

Parâmetros:

-s{Name|Number} Especifica o sinal como um número ou nome, como por exemplo SIGKILL -9 ou SIGTERM -15.

-SignalName Especifica o nome do sinal

ProcessID Especifica um número inteiro decimal representando o processo ou grupo do processo a ser assinalado. Se o o valor do PID é positivo, o comando kill evia o sinal para os processos cujo PID é igual ao PID especificado. Se o valor do PID é 0, o sinal é enviado para todos os processos que possuem o GID igual ao do processo que executou o comando. Neste caso os processos com PID 0 ou -1 não são avisados.

Se o Valor do PID é -1 o comando kill envia o sinal para todos os processos pertencentes ao processo que executou o comando.Neste caso o sinal não é enviado para os processos de números 0 e -1. Se o valor do PID é negativo e diferente de -1, o sinal é enviado para todos os processos cujo GID é igual ao valor absoluto do PID especificado.

-l Lista todos os nomes de sinais conhecidos pelo sistema.

-lExitStatus Lista os nomes de sinais separados do prefixo SIG. Se o ExitStatus é um valor inteiro decimal, o nome correspondente ao sinal é mostrado. Se o ExitStatus é o valor de exit status correspondendo a um processo que foi terminado por um sinal, o nome do sinal correspondente é o do sinal que terminou o processo.

Existe outro comando para parar processos conhecido como killall. Este comando cancela todos os processos que o usuário iniciou, menos o correspondente ao killall.

Sintaxe: **killall [-] [- *Signal*]**

Parâmetros:

- Envia um sinal SIGTERM e depois de 30 segundos envia o SIGKILL para os processos que sobreviveram ao sinal anterior. Este procedimento dá uma chance aos processos de se “terminarem” de um modo não radical. Se ambos(- e -SignalName) forem especificados, o comando **killall** manda o sinal especificado primeiro aos processos e depois envia o SIGKILL, a fim de que os processos tenham uma oportunidade de tratar o sinal anterior antes de serem cancelados.

- SIGNAL Envia o número ou o nome do sinal

Outros comandos

- **at**

```
$ at now + 2 days backup
$ at now + 2 days /u/ze/backup
$ echo backup | at 3:00 pm Jan 24
$ echo backup | at now tomorrow
$ at -l
$ at -r root.635677200.a
```

- **batch**

```
$ batch <<!
job 2>&1 > outline | mail root
```

- **cron**

- **crontab**

```
0,30 * * * * /bin/date > /dev/console
0    0 * * * calendar -
15 0 * * * /usr/etc/sa -s > /dev/null
0,30 * * * * /etc/dmesg - >> /u/msg
```

NOTAS:

O usuário algumas vezes precisa executar suas tarefas em momentos em que o sistema não esteja tão carregado, ou porque o próprio processo a ser executado carregará muito o sistema.

Pode acontecer também do usuário estar interessado em executar tarefas em horários específicos para não prejudicar a execução de outros procedimentos. Em todos estes casos citados utilizamos comandos do sistema que nos permitem alcançar estes objetivos. São eles:

- **at**

Executa comandos em horários pré-determinados. Qualquer job é executado apenas uma vez. Depois da execução da tarefa uma mensagem é enviada ao usuário que executou o comando, avisando-lhe sobre os erros e mostrando a saída do comando. A mensagem contém também o número do processo associado à tarefa executada.

Um usuário pode executar comandos via o comando **at** somente se seu username aparece no arquivo `/var/adm/cron/at.allow`. Se este arquivo não existe, o comando `at` verifica o arquivo `/var/adm/cron/at.deny` para checar se o usuário não está proibido de executar o comando `at`. Se nenhum destes arquivos existem, somente o usuário *root* pode executar o comando `at`.

Sintaxe:

`at [-c | -k | -s | -q`*`Queue] [-m] {-t Date | Time [Day] Increment]}`*
`at -l [Job ... | -qQueue]`
`at -r Job ...`

Parâmetros:

-c	Requer o uso da csh shell para executar os comandos do job.
-k	Requer o uso da ks shell para executar os comandos do job.

-l Lista seus jobs na fila de execução. Somente o usuário root pode ver os jobs dos outros usuários na fila.

-m Envia um mail para o usuário com o resultado da execução do job.

-q Queue Especifica a fila na qual o job executará. A fila default é *a*. As filas *b,c* e *d* são reservadas para jobs do comando `batch`, jobs do comando `cron` e do comando **sync**, respectivamente.A fila *e* executa jobs que usam a shell ksh, equivalente a opção -k. E a fila *f* executa jobs com a shell csh, equivalente à opção -c.

-r Job ... Remove jobs que estão na fila para execução, onde Job é o número associado pelos comandos `at` ou `job`. Se o usuário não tiver privilégio de *root,* ele pode remover somente os seus próprios jobs. O comando `atrm` é disponível somente para o *root* e é utilizado para remover jobs de outros usuários.

-b Requer a shell `sh` para execução dos comandos do job.

-t Date Submete o job para ser executado no horário especificado pela variável Date.
Date é do seguinte formato :
[*[CC]YY]MMDDhhmm[.SS],* onde
CC -dois primeiros digitos do ano;
YY - dois útimos digitos do ano;
MM - mês; mm- minuto;
DD - dia; SS - segundo.
hh- hora;

-t Time Especificado no seguinte formato : *hh:mm*, seguindo dos seguintes sufixos opcionais: *am* ou *A, pm* ou *P, zulu*(horário Greewich Mean Time). O comando **at** também reconhece os nomes *noon* ou *N, midnight* ou *M* e *now*.

O valor opcional de Day pode ser especificado como o nome do mês seguindo do número do dia, ou o dia da semana. Também são reconhecidos os valores *today* e *tomorrow* . O valor para o parâmetro incremento pode ser um sinal *+* seguindo de um dos seguintes nomes: minute(s), *hour(s), day(s)*, etc. Ou pode ser a palavra *next* seguida de um dos valores já citados.

Existe também o comando `batch`, utilizado para executar jobs quando o sistema achar conveniente.

Sintaxe: `batch`

O comando `batch` lê do "input" o comando a ser executado e envia ao usuário um mail com a saída da execução do comando, assim como os erros, a menos que a saída seja redirecionada.

O comando `cron` é um daemon usado para executar comandos em horários pré-determinados. Os comandos a serem executados são colocados em arquivos crontabs, que são armazenados no diretório `/var/spool/cron/crontabs.`

O `cron` examina os arquivos crontabs e os arquivos at somente na hora do boot e quando os arquivos são modificados. Os logs são armazenados no arquivo `/var/adm/cron/log/file`. Um mail é enviado ao usuário com a saída e os erros do job executado.

O comando `crontab` é usado para submeter, editar, listar ou remover comandos para serem executados pelo daemon *cron.* Para um usuário poder utilizar o comando `crontab` o username deve estar registrado no arquivo `/var/adm/cron/cron.allow`. Se este arquivo não existir, o arquivo `/var/adm/cron/cron.deny` é verificado para se determinar se o usuário está proibido de executar o comando `crontab.` Se nenhum destes arquivos existirem, somente o usuário com privilégio de *root* pode usar o comando `crontab.`

Sintaxe : `crontab [-e | -l | -r | -v |` *`File]`*

Parâmetros:

-e Edita o arquivo crontab do usuário

-l Lista o arquivo crontab do usuário

-v Lista o status do arquivo crontab do usuário.

-r Remove o arquivo crontab do usuário do diretório /var/adm/cron/crontabs.

File Copia o arquivo especificado para o /var/adm/cron/crontabs.

NOTAS(cont.):

Cada

entrada do arquivo **crontab** consiste de uma linha com 6 campos, separados por espacos ou “tabs”, definidos como :

Campo	Valores Permitidos
minuto	0-59
hora	0-23
dia do mês	1-31
mês	1-12 (Janeiro - Dezembro)
dia da semana	1-7 (Segunda - Domingo)
comando	

13

Subsistemas

SYSTEM RESOURCE CONTROLLER

- **Mecanismo de controle de subsistemas**
- **Características principais:**
  - **Registro de encerramento anormal de subsistemas;**
  - **Trace de subsistemas;**
  - **Controle de operações de sistemas remotos;**
  - **Refresh de subsistemas;**
  - **Facilidades de notificação quando da ocorrência de anormalidades**
- **Projetado de forma a reduzir a intervenção de operadores**

NOTAS:

O *System Resource Controller (SRC)* fornece um grupo de comandos e subrotinas para tornar a criação e controle de subsistemas mais fácil para o administrador. Um subsistema é qualquer programa, processo ou grupos de programas ou processos que é normalmente capaz de operar independentemente do sistema operacional. Um subsistema é projetado como um todo de forma a atender determinada função.

O **SRC** foi projetado de forma a minimizar a necessidade da intervenção de operadores. Ele provê um mecanismo para controlar processos de subsistemas usando a linha de comandos e a interface C. Este mecanismo inclui:

- Interface consistente para a inicialização, encerramento e verificação de status;
- Registro de término anormal de subsistemas;
- Programa de notificação invocado quando do encerramento anormal de subsistemas e processos correlatos;
- Trace de subsistemas;
- Suporte para controle de operações em sistemas remotos;
- Refresh de subsistemas

Subsistemas (1)

Um subsistema pode ter uma ou mais das seguintes características:

- **É conhecido pelo sistema por seu nome;**
- **Requer um ambiente de execução mais complexo;**
- **Inclui programas aplicativos e bibliotecas além do código do subsistema**
- **Controla recursos que podem ser ativados ou desativados pelo nome;**

NOTAS:

Subsistemas: (2)

- **Requer a notificação se um processo relacionado não consegue se recuperar ou se reorganizar (cleanup);**
- **Requer mais controle operacional que processos comuns**
- **Precisa ser controlado por um operador remoto**
- **Implementa subservidores para gerenciar recursos específicos**
- **Não pode ser colocado em background**

NOTAS:

Alguns exemplos de subsistemas são: *ypserv*, *ntsd*, *qdaemon*, *inetd*, *syslogd* e *sendmail.*

Para listar os subsistemas ativos e inativos em seu sistema emita o comando:

```
lssrc -a
```

Subsistemas (3)

- **Grupo de subsistemas**
- **Subservidores**

NOTAS:

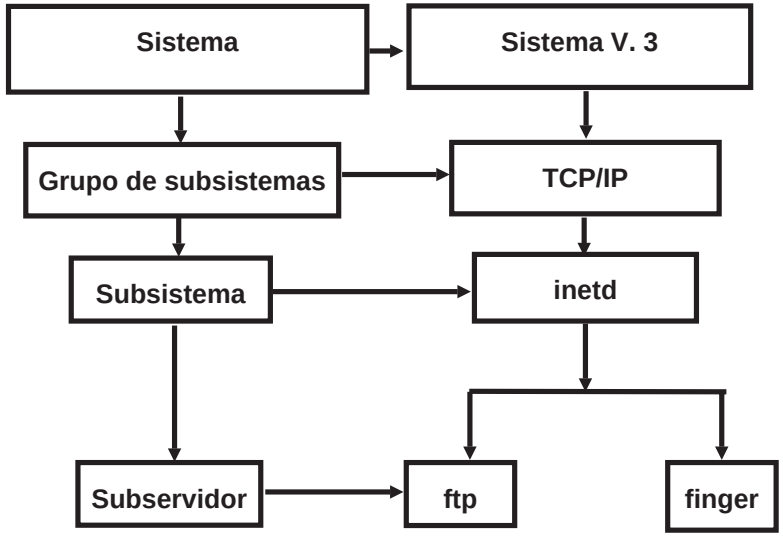
Grupo de subsistemas:

Um grupo de subsistemas é um agrupamento de qualquer grupo específico de subsistemas. O agrupamento de subsistemas permite o controle de vários subsistemas a um mesmo tempo. Exemplos de grupos de subsistemas são TCP/IP, serviços SNA, NIS e NFS.

Subservidores:
Um subservidor é um programa ou processo que pertence a um subsistema. Um subsistema pode ter vários subservidores e ser responsável pela inicialização, encerramento e exibição de status dos subservidores.

Os subservidores são inicializados quando seus subsistemas pai são ativados. Se se tenta ativar um subservidor sem que o subsistema pai esteja ativo, o comando `startsrc` ativa o subsistema juntamente com o subservidor.

SRC
Hierarquia



NOTAS:

O *SRC (System Resource Controller)* possui uma estrutura hierárquica. A hierarquia começa no sistema operacional (versão 3) seguida por um grupo de subsistemas (TCP/IP, por exemplo), os quais contêm um subsistema (*inetd*, por exemplo), o qual por sua vez pode possuir vários subservidores (tais como o daemon *ftpd* e o comando `finger`).

Comandos Importantes

- **srcmstr**
- **startsrc**
- **refresh**
- **traceson**
- **tracesoff**
- **lssrc**

NOTAS:

srcmstr	ativa o System Resource Controller.
startsrc	ativa um subsistema, grupo de subsistemas, ou subservidor.
stopsrc	encerra um subsistema, grupo de subsistemas, ou subservidor.
refresh	efetua a reinicialização (refresh) de um subsistema.
traceson	ativa o trace de um subsistema, grupo de subsistemas, ou subservidores.
tracesoff	desativa o trace de um subsistema, grupo de subsistemas ou de um subservidor.
lssrc	lista informações de status de um subsistema.

SRC
Ativação

- **Criar uma entrada para o daemon srcmstr no arquivo /etc/inittab:**

```
mkitab -i fbcheck \
srcmstr:2:respawn:/etc/srcmstr
```

- **Instruir o comando init a reprocessar o arquivo /etc/inittab:**

```
telinit q
```

NOTAS:

O System Resource Controller (*SRC*) é ativado durante a inicialização do sistema através de um registro contido no arquivo `/etc/inittab`. O arquivo `/etc/inittab` já vem configurado com a entrada para o *srcmstr*. O *SRC* pode ser ativado também através da linha de comandos, um profile, um shell script, mas existem várias razões para ativá-lo durante a inicialização do sistema:

- A ativação através de uma entrada no arquivo */etc/inittab* pemite que o comando `init` o reative se por alguma razão ele se encontrar desativado;
- O *SRC* foi projetado de forma a simplificar e reduzir a necessidade de intervenção de operadores para controlar os subsistemas. A ativação do *SRC* de qualquer outro modo que não seja através do arquivo `/etc/inittab` seria contraprodutiva tendo em vista este objetivo;
- O arquivo `/etc/inittab` default contém um registro para ativar o subsistema de impressão (`qdaemon`) com o comando `startsrc`. Instalações típicas possuem outros subsistemas que são ativados pelo comando `startsrc`**.** Como um pré-requisito para que o comando `startsrc` funcione é que o SRC esteja funcionando, a remoção do daemon `srcmstr` do arquivo `/etc/inittab` fará com que todos os comandos `startsrc` não funcionem.

  Para ativar o *srcmstr:*

- Crie uma entrada para o daemon srcmstr no arquivo /etc/inittab usando o comando mkitab. Por exemplo, para criar uma entrada idêntica à que vem no arquivo /etc/inittab default, emitir o seguinte comando:

  ```
  mkitab -i fbcheck srcmstr:2:respawn:/etc/srcmstr
  ```

  A opção `-i fbcheck` assegura que o registro criado será inserido antes dos registros dos demais subsistemas.

- Informe ao comando `init` para reprocessar o arquivo `/etc/inittab`:

  ```
  telinit q
  ```

  Quando o comando `init` reler o arquivo `/etc/inittab` ele irá processar apenas o registro recém-criado e inicializará o *SRC*.

Ativação de subsistemas, grupo de subsistemas ou subservidores

- **/etc/inittab**
- **linha de comandos**
- **SMIT**

NOTAS:

Para ativar um recurso controlado pelo *SRC* (subsistemas, grupo de subsistemas ou um subservidor) use o comando `startsrc`**.** O comando `startsrc` pode ser usado de três maneiras:

- a partir de uma entrada no arquivo `/etc/inittab;`
- a partir da linha de comandos;
- através do *SMIT.*

Ao se ativar um grupo de subsistemas, todos os subsistemas a ele pertencentes são também ativados. Ao se ativar um subservidor, o subsistema ao qual o mesmo pertence é também ativado (caso esteja desativado).

Para que se possa usar o comando **startsrc** é preciso:

- que o **SRC** esteja ativo (**srcmstr** daemon).
- que a pessoa ativando um recurso gerenciado pelo **SRC** possua autoridade do usuário **root**.

Para se ativar um subsistema a partir da linha de comando:
/bin/startsrc -s *nome_do_subsistema*

Para se ativar um subsistema usando **SMIT**:
smit startsys

14

Contabilização de Uso de Recursos

Contabilização do Sistema

São contabilizados os seguintes recursos:

- **Tempo que cada usuário permanece logado no sistema;**
- **Uso do tempo de CPU, memória e recursos de I/O;**
- **Espaço em disco;**
- **Uso de impressoras e plotters;**
- **Número de vezes um determinado comando é usado.**

NOTAS:

O sistema de contabilidade do sistema coleta dados e gera relatórios individuais ou em grupo sobre vários recursos do sistema.

Estas informações podem ser usadas para cobrar dos usuários pelos recursos utilizados e também para monitorar determinados aspectos da operação do sistema.

O sistema de contabilização também fornece dados para avaliar a adequação dos recursos disponíveis às tarefas desempenhadas, definir limites ou cotas para determinados recursos, prever necessidades futuras e encomendar suprimentos para impressoras e outros equipamentos.

O sistema grava um registro para cada sessão e para cada processo. Estes registros são totalizados (`tacct`) organizados de acordo com os usuários e combinados em um registro diário.

Embora a maior parte dos dados de contabilização sejam coletados e processados automaticamente, qualquer membro do grupo `adm` pode emitir certos comandos para obter informações específicas.

Tempo de Conexão

Informações coletadas pelo daemon init e pelo comando login;

/etc/utmp

- **nome do usuário**
- **date e hora do login**
- **port onde foi feito o login**

NOTAS:

O tempo de conexão é coletado pelo comando `init` e pelo comando `login`. Quando o usuário loga o programa `login` escreve um registro no arquivo `/etc/utmp`. Este registro contém o nome do usuário, a data e a hora do login e a porta onde foi feito o login. O comando `who`, entre outros, usa este arquivo para determinar quais usuários estão logados e em quais portas. Se o arquivo `/var/adm/wtmp` existe o comando login acrescenta uma cópia deste registro a ele.

Quando se encerra a sessão, o comando `init` registra o final da sessão escrevendo um outro registro no arquivo `/var/adm/wtmp`. Registros de logout possuem o formato descrito no arquivo `utmp.h`.

O comando `acctwtmp` também grava no arquivo `/var/adm/wtmp` entradas especiais relativas a encerramentos e inicializações do sistema.

Processos

- **Números de grupo e de usuários sob os quais o processo roda;**
- **Primeiros oito caracteres do nome do comando;**
- **Tempo de conexão e tempo de CPU;**
- **Uso de memória;**
- **Número de caracteres transferidos;**
- **Número de blocos de disco lidos ou gravados.**

NOTAS:

O sistema coleta informações sobre os diversos processos à medida em que executam.

O comando `accton` registra estes dados no arquivo `/var/adm/pacct`.

Disco

- **Os registros são coletados no arquivo** */var/adm/acct/nite/dacct*

- **Os registros são criados pelo comando** *dodisk***, o qual é invocado periodicamente pelo comando** *cron*

Impressoras

- **A coleta de dados de impressoras é resultado de uma interação entre o comando** *enq* **e o subsistema** *qdaemon*

NOTAS:

Muitas das informações são coletadas à medida que os recursos são consumidos. O comando `dodisk`, executado a partir da definição na `crontab`, grava periodicamente registros de uso de disco no arquivo `/var/adm/acct/nite/dacct`. Para tal o comando `dodisk` invoca outros comandos. Dependendo do nível de detalhe, podem ser usados os comandos `diskusg` ou `acctdusg` para coletar os dados. O comando `acctdisk` é usado para gravar um registro de totalização. O registro de totalização por sua vez é usado pelo comando `acctmerg` para preparar o relatório de contabiilização diário.

O comando `dodisk` cobra dos usuários pelos links de arquivos encontrados no seu diretório home e divide a conta igualmente entre aqueles que possuem links para o mesmo arquivo. Desta forma o custo de compartilhamento de arquivos é distribuído igualmente entre todos os usuários.

A coleta de dados relativa ao uso de impressoras é resultado de uma interação entre o comando `enq` e o subsistema `qdaemon`. O comando `enq` enfileira o nome do usuário, número do job e o nome do arquivo a ser impresso. Após o arquivo ser impresso, o comando `qdaemon` escreve um registro ASCII no arquivo `/var/adm/qacct`, contendo o nome, número do usuário e o número de páginas impressas. Estes registros podem ser organizados e convertidos em registros de contabilização total.

Relatórios

- Tempo de conexão
- runacct
  - acctcon1
  - acctcon2
- lastlogin
- Processos
  - acctprc1
  - acctprc2
  - acctcms
  - acctcom

NOTAS:

Tempo de conexão

O comando `runacct` invoca dois outros comandos, `acctcon1` e `acctcon2`, para processar os registros de login, logout e de encerramento do sistema que são gravados no arquivo `/var/adm/wtmp`. O comando `acctcon1` converte estes registros em registros de sessão e os escreve no arquivo `/var/adm/act/nite/lineuse`. O comando `acctcon2` por sua vez converte os registros de sessão em um registro de totalização que é gravado no arquivo `/var/adm/logacct`, que o comando `acctmerg` adiciona aos registros diários.

Se o comando `acctcon1` é invocado a partir da linha de comandos, deve-se incluir a opção `-l` para gerar o arquivo `/var/adm/acct/nite/lineuse`. Para produzir um relatório geral para determinado período (`/var/adm/acct/nite/reboots`), invocar o comando `acctcon1` com a opção `-o`.

Processos

Dois comandos processam os dados coletados no arquivo */var/adm/pacct*. O comando *acctprc1* traduz o ID do usuário no nome do usuário e grava registros ASCII contendo os tempos de CPU gastos (prime e noprime), tamanho de memória usada, número de I/O.

O comando *acctprc2* transforma estes registros em registros de totalização que são adicionados aos relatórios diários.

O comando *acctms* sumariza recursos utilizados por nome de comando.

O comando *acctcom* provê informações mais detallhadas para comandos específicos (quem usou o comando, etc.).

Uso de disco e impressora

Os registros de utilização de disco coletados no arquivo */var/adm/acct/nite/dacct* são incluídos nos relatórios diários pelo comando *acctmerg*.

O mesmo ocorre com os registros de utilização de impressora que estão no arquivo */var/adm/qacct*.

Principais Arquivos de Account

- **Arquivos de Dados**
  - */var/adm/pacct*
  - */var/adm/wtmp*
  - */var/adm/Spacct.mmdd*
- **Arquivos de sumarização e relatórios estão nos diretórios**
  - */var/adm/acct/nite*
  - */var/adm/acct/sum*
  - */var/adm/acct/fiscal*

Arquivos de formato

NOTAS:

Arquivos de dados

O arquivo */var/adm/pacct* é o principal arquivo de coleta de dados de account, onde é gravada a maioria dos registros de utilização de recursos.

O arquivo */var/adm/wtmp* contém os principais registros relativos aos logins dos usuários.

Os arquivos */var/adm/Spacct.mmdd* são arquivos temporários utilizados no instante de execução do comando *runacct*. Podem ser deletados se o filesystem ficar cheio.

Arquivos de sumarização/relatórios

O arquivo */var/adm/acct/nite/daytacct* é o principal arquivo de dados de account. Possui a maioria das informações coletadas diariamente e é gerado pelo comando *runacct* (que é ativado pelo daemon *cron*). É um arquivo binário que condensa informações de diversos outros arquivos de account.

Os arquivos */var/adm/acct/sum/tacct.mmdd* nada mais são do que os arquivos de account relativos ao uso de CPU, disco, impressora, I/O para o dia mmdd.

Os arquivos */var/adm/acct/sum/rprtmmdd* são os relatórios diários de sumarização de uso de comandos. São utilizados pelo comando *monacct* na geração do relatório fiscal */var/adm/acct/fiscal/fiscrptnn*, juntamente com os arquivos */var/adm/acct/sum/tacct.mmdd*.

Arquivos de Formato

Aos principais arquivos de account estão associados arquivos de formato que descrevem os campos sumarizados. A maioria dos arquivos de formato se encontra no diretório */usr/include/sys*.

Principais comandos de Account

- *runacct*
- *ckpacct*
- *monacct*

NOTAS:

Os comandos de account funcionam de muitas maneiras. Alguns comandos:

-Coletam dados ou produzem relatórios para um tipo específico de contabilização (uso de disco, tempo de conexão)

- Chamam outros comandos de account, que podem tanto gerar registros de account como gerar/atualizar relatórios de account.

O comando *runacct* manipula o principal procedimento de account diário. Normalmente inicializado pelo daemon *cron,* ele chama diversos outros comandos para processar arquivos com dados de account e para produzir/atualizar sumários. Além disso gera relatórios diários de utilização de recursos.

O comando *ckpacct* manipula o tamanho do arquivo */var/adm/pacct*, que é o arquivo onde é gerada a maioria dos registros de account. Quando o arquivo */var/adm/pacct* ultrapassa 500 blocos, o comando *ckpacct* chama outro comando (o comando *turnacct switch*) que temporariamente desabilita o account e cria um arquivo */var/adm/pacctx*, liberando o arquivo */var/adm/pacct* para novos registros.

O comando *monacct* produz um sumário periódico a partir dos arquivos diários. A saída deste comando é o arquivo */var/adm/acct/fiscal/fiscrptnn*, que totaliza tanto os recursos de CPU, I/O, disco e impressoras quanto a utilização de cada comando para o período fiscal determinado (geralmente definido como um mês).

Configuração do Account

- **Prover acesso correto aos arquivos de account**
- **Configurar feriados e horários Prime e Noprime**
- **Habilitar account na máquina**
- **Identificar os filesystems que serão contabilizados**
- **Especificar o arquivo de dados de impressora**
- **Criar diretórios de account**
- **Incluir chamadas dos comandos de account na crontab (dodisk, ckpacct, runacct, monacct)**

NOTAS:

Prover o acesso correto aos arquivos de account através do comando */usr/sbin/acct/nulladm wtmp pacct*.

Configurar arquivo */etc/acct/holidays*, definindo nesse arquivo os feriados do ano e os horários Prime e Noprime.

Habilitar o account incluindo no */etc/rc* o comando:

/usr/bin/su -adm -c /usr/bin/acct/startup

Editar o arquivo */etc/filesystems* identificando quais os filesystems serão contabilizados por uso de disco:

account=true

Editar o arquivo */etc/qconfig*, especificando o arquivo de dados a ser usado na contabilização do uso da impressora, se for o caso:

acctfile= /var/adm/qacct

Como usuário adm, criar os diretórios nite, fiscal e sum:

su -adm
cd /var/adm/acct
mkdir nite fiscal sum
exit

Incluir chamadas dos comando de account na crontab:

*01*** /usr/sbin/acct/dodisk*
*02***/usr/sbin/acct/ckpacct*
*03***/usr/sbin/acct/runacct2>/var/adm/acct/nite/accterr*
*041**/usr/sbin/acct/monacct*