

CBPF - CENTRO BRASILEIRO DE PESQUISAS FÍSICAS

Rio de Janeiro

Notas Técnicas

CBPF-NT-004/96

December 1996

Introdução à Administração do Sistema Solaris 2

Mário Vera & Renato Moutinho



ÍNDICE

1-Introdução	1
1.1 - O papel do Administrador do Sistema	1
1.2 - Visão Geral de Administração de Sistemas UNIX	1
2 - Inicializando o Sistema	3
2.1 - <i>Boot & Shutdown</i>	3
2.2 - <i>OpenBoot & POST</i>	3
2.3 - <i>The Monitor & NVRAM</i>	4
3 - Instalando o Solaris	5
3.1 - Apresentação	5
3.2 - Tipos de Sistemas	5
3.3 - <i>Solaris Software Group</i>	5
3.4 - <i>Install Servers</i>	6
3.5 - Gerenciamento de Partições	7
3.5.1 - UFS	7
3.5.2 - NFS	9
3.5.3 - TMPFS	9
3.5.4 - PROCFS	10
3.6- Os Níveis de Inicialização do Sistema	10
4 - Configuração do Sistema	11
4.1 - Implementando o NFS	11
4.1.1 - O comando <i>mount</i>	11
4.2 - NIS	12
4.2.1 - Definição	12
4.2.2 - NIS+	12
4.2.3 - Vantagens do NIS+ sobre o NIS	13
4.2.4 - Terminologia do NIS+	13
4.2.5 - Configuração do <i>Root Domain</i>	15
4.2.6 - Criando uma Servidora Replica	16
4.2.7 - Adicionando um cliente a um domínio já existente	17
4.3 - DNS	17
4.3.1- Tipos de RR	19
4.3.1.1 - SOA	19
4.3.1.2 - A	20
4.3.1.3 - NS	21
4.3.1.4 - CNAME	21
4.3.1.5 - PTR	21

4.3.1.6 - MX	22
4.3.1.7 - HINFO	22
4.3.2- Atualização de Informações	23
5 -Manutenção do Sistema	24
5.1 - Gerenciando a Estrutura do Sistema	24
5.2 - Gerência de Processos	24
5.3 - Gerência de Dispositivos	25
5.3.1 - <i>Physical Devices</i>	25
5.3.2 - <i>Logical Devices</i>	25
5.3.3 - Nomenclatura	26
5.3.3.1 - Discos :	26
5.3.3.2 - Fitas Magnéticas :	26
5.3.4 - Gerenciador de Dispositivos (<i>Volume Management</i>)	26
5.3.5 - <i>Tape Devices</i>	26
5.4 - Preparando o <i>Backup</i>	27
5.4.1 - Tipos de <i>backup</i>	27
5.4.2 - Comandos	28
5.4.3 - Escolha do <i>FileSystem</i>	28
5.4.4 - Escolha do Dispositivo	29
5.4.5 - Dia de <i>backup</i> ..	29
5.5 - <i>Shell Scripts</i>	30
5.5.1 - A Escolha do <i>Shell</i>	30
6 - Bibliografia	34

1 - Introdução

1.1 - O papel do Administrador do Sistema

A Administração de Sistemas de Computadores exerce papel fundamental em qualquer ambiente de trabalho que mantenha informações armazenadas logicamente. O Sistema Operacional disponibiliza dados e serviços que serão oferecidos a seus usuários. A ferramenta do Administrador é o Sistema Operacional, ele deve conhecer o Sistema profundamente para que possa organizar, de forma simples e eficiente, o ambiente de trabalho dos usuários.

A configuração do ambiente de trabalho, deve ser feita de acordo com as características específicas das atividades de seus usuários. Um Sistema que atende a Pesquisadores de uma Instituição Científica será organizado de forma diferente que o Sistema de uma empresa da área de Consultoria Imobiliária por exemplo.

Um Sistema bem administrado pode ser encarado como fator determinante da qualidade da produção, já que proporciona um ambiente de trabalho onde seus usuários podem acessar dados e serviços de forma simples e objetiva, aumentando assim sua produção.

Fica claro, desta forma, a importância e valor de um bom Administrador de Sistemas.

1.2 - Visão Geral de Administração de Sistemas UNIX

O Sistema Operacional *Solaris*, abordado neste curso, foi desenvolvido pela empresa americana Sun Microsystems, e tem como base o UNIX.

De uns anos pra cá, várias empresas fornecedoras de Sistemas baseados em UNIX vêm se mostrando interessadas em acabar com a impopularidade deste grande Sistema. Em Sistemas antigos, como o SunOs4, Administradores experts em interagir com UNIX recebiam um título, que ficou bastante difundido no universo UNIX, de “*wizards*” (magos). Isso porque conseguiam dominar, de forma eficiente, dezenas de comandos com sintaxes extremamente complicadas. Esta, e outras características, fizeram do UNIX um Sistema odiado por usuários, principalmente da área comercial (onde não há tempo para aprender tantos comandos), por não ter sido desenvolvido com intuito de fornecer um ambiente amigável para usuários inexperientes.

Com o advento de Sistemas como o *Solaris*, a vida de Administradores e usuários ficou muito mais simples. Ferramentas para administração de contas, serviços e periféricos, uma Interface Gráfica bastante amigável, e principalmente a padronização do "UNIX System V Release 4 (SVR4)" por quase todas as empresas fabricantes de Sistemas, vêm trazendo uma nova perspectiva de mercado para o UNIX, que deixa de ser um Sistema difundido unicamente no meio Acadêmico para se tornar um grande concorrente no mercado de Sistemas Operacionais.

2- Inicializando o Sistema

2.1 - *Boot & Shutdown*

Estes dois procedimentos ficarão gravados na memória de qualquer Administrador. O processo de “derrubar” o Sistema (“*shutdown*”) será descrito mais a frente e o de inicialização (“*boot*”) é abordado abaixo.

2.2 - *OpenBoot & POST*

A Sun desenvolveu um *firmware* de inicialização independente de *hardware* denominado OpenBoot. Este *firmware* permite ao Administrador as seguintes facilidades :

- *Plug-in Device Drivers* - podemos instalar uma Controladora de Dispositivos através de uma placa conectada. A principal vantagem disso e poder utilizar Controladoras sem modificar a PROM.
- *FCode Interpreter* - possibilita a utilização de dispositivos escritos em *Fcode* (Firmware Code).
- *Device Tree* - estrutura que mostra a árvore contendo todos os dispositivos instalados e conectados ao Sistema.
- *Programamable User Interface* - oferece uma linguagem (*Forth*) para escrever programas que permitam depurar *hardware* e *software*.

O ambiente *OpenBoot* é acessado através do *prompt* da PROM¹. Isto é possível de 3 formas :

- “Derrubando” o Sistema Operacional (*halt*).

¹ Este *prompt* se resume ao símbolo de *Ok*.

- Digitando "*stop+a*".
- Ligando o Sistema (Power On), e digitando "*stop+a*" após a mensagem de *banner*.

2.3 - *The Monitor & NVRAM*

Cada Sistema possui um chip de PROM (*programmable read-only memory*) instalado. No ambiente *OpenBoot*, a execução independe do Sistema Operacional, desta forma ainda não estamos interagindo com o *Solaris*, mas sim com o *monitor*, que reside na PROM e pode ser de dois tipos :

- *Restricted monitor* (antigas Sun4/nnn).
- *Forth monitor*.

O *monitor* controla a operação do Sistema antes do *Kernel* estar disponível. Quando ligamos o Sistema, o *monitor* executa rapidamente um procedimento de auto-teste que verifica *hardware* e memória do Sistema (POST - *Power On Self Test*). Se não for encontrado nenhum erro, o Sistema começa o processo automático de *boot*.

Os parâmetros de inicialização do Sistema estão guardados na memória não volátil de acesso aleatório (NVRAM). Estes parâmetros podem ser alterados de acordo com a preferência do administrador. Para verificar os valores desses parâmetros, pelo *prompt* da PROM :

Ok printenv

Um exemplo seria o parâmetro *selftest-#megs* que indica o número de *bytes* a serem testados (em Mb). Por *default* este parâmetro tem valor 1, ou seja serão testados 1Mb do total de espaço de memória do Sistema (RAM). Para alterar este valor, pelo *prompt* da PROM :

Ok setenv selftest-#megs n ²

Os recursos oferecidos pelo *OpenBoot* são muito poderosos, e no momento oportuno serão de grande utilidade, porém os comandos mais usados são :

- banner
- go
- boot

² n é o novo valor obviamente .

- probe-scsi
- test
- reset

3 - Instalando o *Solaris*

3.1 - Apresentação

O *Solaris* vem em CD-ROM, juntamente com outros pacotes de *patches* e *updates*, e o CD_ROM contendo o manual em *HyperText* batisado de *AnswerBook*. Caso o Sistema não possua uma unidade de leitura de discos CD-ROM, a instalação deverá ser remota (num outro Sistema.que contenha a unidade de leitura).

Para interagir com o ambiente de Instalação do *Solaris* digite :

Ok boot cdrom

Após alguns testes, você deverá responder algumas perguntas referentes a localização, data, hora etc... para que o Sistema possa ser autenticado.

3.2 - Tipos de Sistemas

Existem 3 tipos distintos de Sistemas :

- *Server*
- *Standalone*
- *Diskless*
- *Dataless*

Durante o processo de instalação, o Administrador deverá selecionar o tipo de Sistema a ser instalado.

3.3 - *Solaris Software Group*

Após a escolha do tipo de Sistema, o Administrador deverá optar pelo grupo de Sistema a ser instalado. Sempre que possível deve-se optar pela opção mais abrangente (se o espaço em disco for o suficiente).

A tabela a seguir lista os possíveis grupos :

<i>Opção do Grupo³</i>	<i>O que será instalado</i>	<i>Espaço em disco</i>
<i>Core System Support</i>	O mínimo para carregar o Solaris	70 Mb
<i>End User System Support</i>	<i>Core group</i> mais o recomendado para o usuário final	180Mb
<i>Developer System Support</i>	<i>End User Group</i> mais software para desenvolvimento, incluindo bibliotecas, manuais, ferramentas para programação, Compiladores e Depuradores não estão incluídos	270 Mb
<i>Entire Distribution</i>	O pacote completo (todo o CD-ROM), Compiladores e Depuradores não estão incluídos	360Mb

3.4 - Install Servers

Existem 3 formas diferentes de se instalar o **Solaris** :

- *Interactive*
- *Jumpstart*
- *Custom Jumpstart*

A escolha do método vai depender do número de Sistemas a serem instalados, e também do grau de especificação que eles deverão sofrer. Por exemplo, se formos instalar apenas um pacote, o melhor método é sem dúvida o interativo. Porém, se tivermos que instalar o **Solaris** em 10 Sistemas, sendo que metade destes devem ter uma configuração de *Dataless*, sendo 2 *Servers* e os outros *Diskless*, então o tempo perdido

³ Existe um outro grupo disponível em alguns pacotes, é o *Entire Distribution plus OEM*, que instala pacotes para produtos de outros fabricantes.

em instalar um por um será muito dispendioso. Nesta hipótese deveríamos utilizar a instalação customizada (*Custom Jumpstart*).

Quando temos que instalar o Sistema em várias máquinas, já sabemos que devemos usar as instalações de *Jumpstart*, porém estas instalações requerem um Servidor de Instalação configurado para atender as especificações do Cliente. Para disparar a instalação em *Jumpstart* :

Ok boot net - install

Neste momento, o Cliente envia uma mensagem para todos os que fazem parte da Rede Local (*broadcast*), perguntando quem é seu *Install Server*. Através de características específicas do Cliente, o Servidor responde ao pedido e começa a instalação do *Solaris*.

Os parâmetros mais simples para caracterizar um cliente são, por exemplo o nome do Cliente e seu endereço de IP. Para o Cliente ser identificado por estas informações deve-se incluir, inicialmente, o Cliente no NIS do Servidor de Instalação (se o NIS já estiver rodando), utilizando a nova ferramenta para a Administração de Sistema que vem com o *Solaris*, chamada *Admintool*. Esta ferramenta é muito simples de se utilizar.

3.5 - Gerenciamento de Partições

3.5.1 - UFS

São os *FileSystems* do UNIX. Compõe a árvore de diretório do Sistema (*/var /usr /etc...*), e estão fisicamente conectados ao Sistema.

Um disco de Sistema usado para o armazenamento de dados é dividido em partições ou *Slices*. O Sistema de Arquivos do UNIX é nada mais do que a árvore de diretórios que compõem o Sistema. Estes diretórios que compõem a árvore são denominados *FileSystems*⁴. Cada *FileSystem* pode estar ocupando uma partição diferente do disco, sendo que normalmente o número máximo de partições é 8. O *Solaris* tem como *default*, dividir as partições de acordo com uma tabela pré-determinada, porém na maioria dos Sistemas temos de fazer uma reconfiguração desta tabela.

Podemos modificar a tabela de partição do disco, antes e depois de instalar o Sistema Operacional. Logicamente é recomendado que a configuração seja feita antes da instalação do *Solaris*. Para isso podemos :

⁴Lembre-se : “ para o UNIX tudo é arquivo ! ”

1- Customizar o particionamento preenchendo as tabelas do ambiente de instalação. (Instalação Interativa)

2- Customizar o particionamento juntamente com outras informações sobre a instalação do Sistema no arquivo de Configuração *Profile*. (Instalação Customizada)

Se for necessário reparticionar um disco após a instalação do Sistema Operacional devemos :

- Usar o comando *format*⁵. Após o reconhecimento dos discos instalados no Sistema o *format* lhe dá um *prompt*, digite a opção *partition* :

```
format> partition
```

PARTITION MENU :

0	- change '0' partition
1	- change '1' partition
2	- change '2' partition
3	- change '3' partition
4	- change '4' partition
5	- change '5' partition
6	- change '6' partition
7	- change '7' partition

select - select a predefined table

modify - modify a predefined partition table

name - name the current table

print - display the current table

label - write partition map and label to the disk

quit

⁵ Este comando só pode ser utilizado pelo Supervisor.

Para verificarmos como o *FileSystem* está organizado :

#df -k

Ao organizar a tabela de particionamento do disco (ou o arquivo) devemos ter em mente uma regra geral :

- Nunca dividir o disco contendo os diretórios de Sistema (/usr /var...) na mesma partição.

Essa restrição deve-se ao fato de que estes *FileSystems* são os que podem apresentar problemas e conseqüentemente devem estar passíveis de correção. Colocando uma partição única para todos os *FileSystems* torna o trabalho muito mais difícil (talvez impossível). Enquanto que dividindo bem os *FileSystems* podemos mecher separadamente em cada um deles.

3.5.2 - NFS

O NFS (*Network File System*) é um Serviço baseado em RPC (*Remote Procedure Call*) que permite a distribuição de *FileSystems* (DFS) através de uma Rede de Computadores .⁶

No que diz respeito a etapa de instalação do Sistema, pode-se listar os *File Systems* que desejamos acessar durante a instalação interativa , assim como na etapa de Configuração do Sistema.

3.5.3 - TMPFS

A área de *Swap* consiste numa área reservada em disco especialmente formatada para ser usada exclusivamente pelo Sistema Operacional, quando este quiser adiar a execução de um determinado processo em memória, guardando as informações na área de *Swap* (*Swap-out*); e vice-versa (*Swap-in*).

Esta área deve ser suficientemente grande para que o Sistema não gaste muito tempo com *Swap* (*Trashing*). Uma regra prática é fazer a área de *Swap* ter o dobro da memória do Sistema.⁷

Por default o *Solaris* monta a área de *Swap* na mesma partição que o diretório /tmp pois ambos serão deletados após o processo de *boot*.

⁶ NFS é explicado mais adiante.

⁷ Este valor pode ser alterado de acordo com a necessidade do Sistema.

Para gerenciar a área de *Swap* usamos o comando *swap*. Para listarmos qual(ais) a(s) área(s) que está(ão) sendo utilizada(s) :

```
#swap -l
```

Para adicionarmos um arquivo especial⁸ à área de *swap* :

```
#swap -a [swapfile name]
```

3.5.4 - PROCFS

Existe um *FileSystem* dedicado a conter os processos listados pelo número (PID), denominado */proc*, este diretório não ocupa espaço em disco.

3.6- Os Níveis de Inicialização do Sistema

O SunOS possui 8 estados de inicialização. O estado *default* de funcionamento, especificado no arquivo */etc/inittab*, é o nível 3 de execução.

A tabela a seguir lista todos os possíveis estados :

<i>Init State/Run Level</i>	Função
0	“Derrubada de Sistema” (<i>Power Down State</i>)
1	Estado de Administração do Sistema (<i>Single User</i>)
2	Estado de operação em nível Multi-Usuário
3	Estado de operação em nível Multi-Usuário (recursos disponíveis)
4	Estado alternativo (não é usado atualmente)
5	<i>Software reboot State</i>
6	Reinicialização
S,s	Estado de Administração do Sistema (<i>Single User</i>)

O programa */sbin/init* mantém o Sistema rodando corretamente. Além disso, */sbin/init* é o comando que deve ser usado para se trocar o estado de execução do Sistema.

⁸ O arquivo deve ser construído utilizando procedimentos específicos para área de *Swap*.

Existem na prática 4 tipos de *init states* :

- *Power-Down* (0)
- *Single-User* (1, S ou s)
- *Multi-User* (2 e 3)
- *Reboot* (5 e 6)

No SunOS5, os *scripts* responsáveis pela disponibilidade dos serviços correspondentes a um *init state* específico, estão localizados no diretório `/etc/rc[init state]` de acordo com o nível de execução.

Nestes diretórios os *scripts* que têm o prefixo 'S' (*Start*) correspondem aos ativadores de serviços, e os de prefixo 'K' (*Kill*) aos terminadores de serviços.

4 - Configuração do Sistema

4.1 - Implementando o NFS

Um *FileSystem* pode estar logicamente distribuído formando o que é conhecido na literatura como um *Distributed File System (DFS)*⁹. A Sun implementou esta técnica desenvolvendo dois tipos de Sistemas : NFS e RFS. O RFS não existe mais no *Solaris*, restando apenas o NFS.

Para verificar os parâmetros de configuração de todos os *FileSystems* :

```
#more /etc/vfstab
```

4.1.1 - O comando *mount*

Para ter acesso a um *FileSystem* o Administrador deve se certificar de dois pré-requisitos :

- O Servidor do *FileSystem* deve conceder permissão de acesso ao Cliente.
- O Cliente deve “montar” o *FileSystem*.

Para verificar se o Servidor está disposto a dividir o *FileSystem* com seus Clientes, no Servidor use :

```
#share
```

Para verificar se o Cliente está montando o *FileSystem*, no Cliente digite :

```
#showmount
```

Por exemplo, para o Cliente *SunClient* montar o *FileSystem* contendo a área de usuários do Servidor *SunServer*, o procedimento seria :

No Servidor ..

```
SunServer#share /export/home
```

⁹ Por esse motivo os arquivos exportados por um Servidor estão listados no diretório */etc/dfs (SVR4)*.

No Cliente :

SunClient#mount /mnt/home

Parâmetros de especificação de como deve ser “exportado” os *FileSystems* ficam guardados no arquivo */etc/dfs/dfstab* no Servidor, e os parâmetros de como deve ser montado o *FileSystem* ficam no arquivo */etc/vfstab* no Cliente.

4.2 - NIS

4.2.1 - Definição

Dentro de qualquer Sistema, existem informações, de caráter administrativo, que serão consultadas por serviços, pertencentes ou não ao Sistema, e que conseqüentemente devem estar imediatamente disponíveis. Tais informações ficam armazenadas em tabelas na memória do Sistema. Exemplos destas tabelas seriam :

- *Hosts* : guarda os nomes dos Sistemas que fazem parte da Rede Local.
- *Password* : guarda as informações relacionadas a cada usuário do Sistema (senha, números de identificação, área de trabalho, *shell* etc...).

Agora, imagine se tivermos diversos Sistemas fazendo parte de uma Rede de Computadores, e que dividem serviços oferecidos por diversos membros desta Rede. Uma atualização de informação deveria ser registrada, ou seja distribuída por todos os membros da Rede. Numa Rede com 5 Sistemas isso é bastante simples, porém no caso de termos uma Rede de 50 Computadores, distribuídos fisicamente inclusive, inviabilizaria a tarefa do Administrador pois ele teria de atualizar a informação em cada Sistema.

Para eliminar este problema, criou-se um serviço que gerencia as informações dentro de uma Rede. Este serviço foi batizado de NIS (*Network Information Service*). Ele distribui a informação contida nas tabelas que estão sendo gerenciadas pelo NIS (*NIS maps*), de forma que qualquer alteração feita é automaticamente registrada em todos os Sistemas que fazem parte do domínio do NIS.

4.2.2 - NIS+

O NIS acompanhou os Sistemas Operacionais da Sun e de outros fabricantes, durante muito tempo. Porém, seu mecanismo de Segurança é muito fraco, pois, com o aumento do poder computacional dos Sistemas atuais, tornou-se fácil reverter o processo de criptografia de suas informações.

A Sun desenvolveu um novo serviço de gerência de informações baseado no NIS, porém com um nível de segurança e organização muito superiores. Este novo serviço acompanha o *Solaris* e foi batizado de NIS+.

O NIS+ possui um mecanismo de distribuição de credenciais baseado num protocolo desenvolvido no M.I.T. denominado *Kerberos*. Este protocolo se resume em criar, dinamicamente, os parâmetros que servirão de base para o processo de criptografia das informações acessadas por cada serviço dentro do Sistema.

Neste protocolo, usuários e serviços possuem credenciais que serão verificadas a cada tentativa de acesso a serviços do Sistema.

4.2.3 - Vantagens do NIS+ sobre o NIS

As duas principais vantagens do NIS+ sobre o NIS são :

- Possibilidade de estabelecer vários domínios formando uma hierarquia onde cada componente possui sua própria configuração.

- As informações são criptografadas com chaves geradas em tempo de acesso ao serviço.

O NIS+ possibilita além das vantagens listadas acima uma maior flexibilidade na organização das tabelas do que o NIS (que não oferece nenhuma facilidade). Permitindo ao Administrador configurar permissões de acesso e uso de serviços de acordo com o nível de segurança desejado.

4.2.4 - Terminologia do NIS+

A terminologia do NIS+ segue o padrão de nomes do DNS (*Domain Name Service*), através deste padrão podemos ter uma hierarquia de domínios.

Os principais termos são :

- *NameSpace* : espaço global que informações servidas pelo NIS+ são levadas.

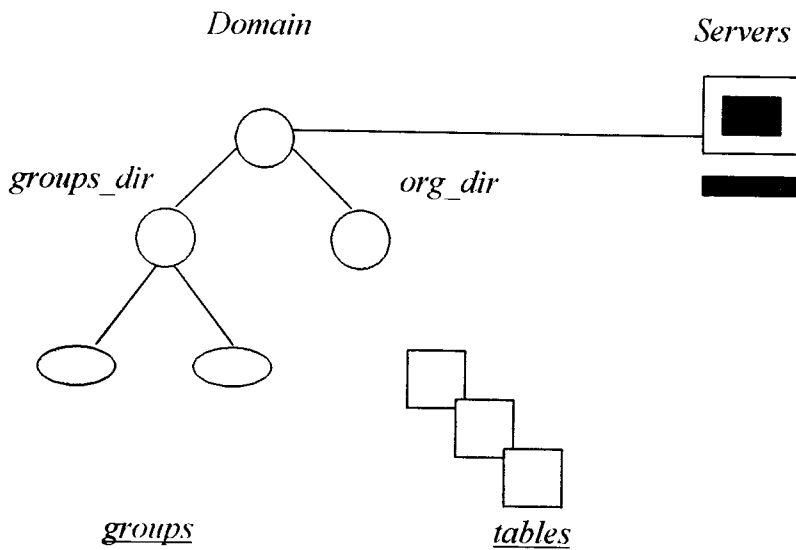
- Domínio : é a estrutura que engloba servidores e objetos. O principal domínio é aquele que servirá de "pai" (*Root domain*) para toda a hierarquia de domínios¹⁰.

- Objetos : são as tabelas e os diretórios de grupos.

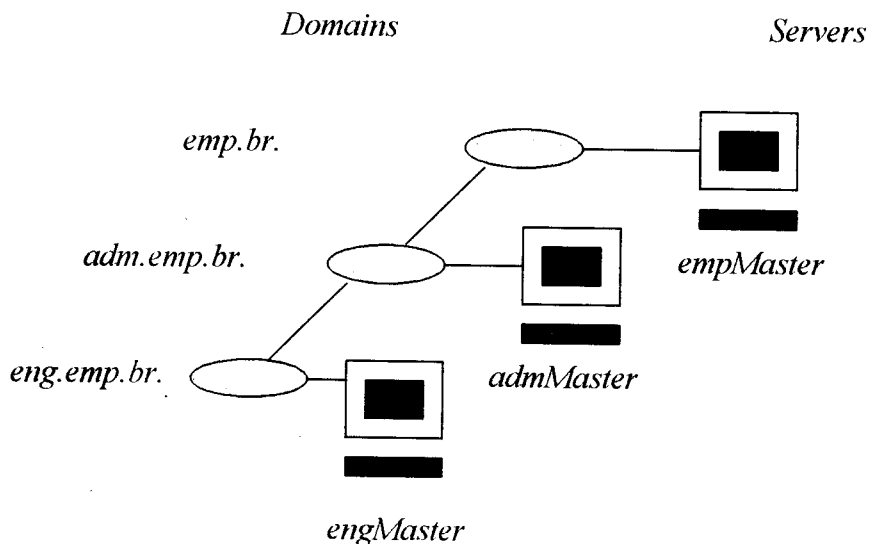
¹⁰ Cada Servidor pode servir a vários domínios, porém um domínio só pode ser servido por um unico Servidor.

- *org_dir* : objeto que guarda as tabelas.
- *groups_dir* : objeto que guarda os grupos reconhecidos dentro do domínio.
- Servidores : Sistemas que dão suporte a um determinado domínio, podem ser *Master*, ou *Replicas*.

Exemplo de um domínio :



Exemplo de uma hierarquia de domínios :



4.2.5 - Configuração do *Root Domain*

Configurar o domínio raiz do *NameSpace* envolve os seguintes passos¹¹ :

- 1 - Estabelecer um nome para o domínio a ser criado na *Master Server*.
- 2 - Checar o arquivo */etc/nsswitch.conf*.
- 3 - Criar o diretório raiz e inicializar a *Master Server*.
- 4 - Ativar o *NIS+ daemon*.
- 5 - Criar os subdiretórios e suas tabelas.
- 6 - Criar as credenciais para a *Master Server*.
- 7 - Criar o grupo *admin* para o diretório raiz.
- 8 - Acrescentar o *Master Server* ao grupo *admin*.
- 9 - Atualizar as chaves publicas do diretório raiz.
- 10 - Ativar o *NIS+ cache manager*.
- 11 - Reinicializar o *NIS+ daemon* com o nível de segurança 2.
- 12 - Adicione as credenciais locais (LOCAL) do Administrador ao domínio raiz.
- 13 - Adicione as credenciais DES do Administrador ao domínio raiz.
- 14 - Adicione credenciais para outros Administrador.
- 15 - Adicione os Administradores ao grupo *admin* do domínio.

A tabela abaixo apresenta um resumo dos procedimentos de configuração do domínio raiz na sequência que devem ser executados¹² :

<i>RootMaster</i> % su	"Log on" do Supervisor. (Entre com a senha do usuário <i>root</i> .)
Password:	
#domainname	Verifica-se o nome atual do domínio.
#more <i>/etc/nsswitch.conf</i>	Verifica-se se o Serviço de Nomes (<i>Name Service Switch</i>) está corretamente configurado.
#nisinit -r	Inicializa o <i>Root Master</i>
#rpc.nisd -S 0	Ativa o <i>daemon</i> controlador de pedidos com o nível de segurança 0.
# <i>/usr/lib/nis/nissetup</i>	Cria os objetos do domínio.
#nisaddcred des	Cria a credencial para o <i>Root Master</i> . (Entre com a senha de login do usuário <i>root</i> .)
Enter login password:	
#nisgrpadm -c admin.dom.br.	Cria o grupo <i>admin</i> .
#nischmod g+rncd dom.br.	Estabelece direitos totais ao grupo <i>admin</i> sobre o diretório raiz.

¹¹ Todos os passos devem ser feitos utilizando-se a conta de Supervisor.

¹² *RootMaster* é a *Master Server* do domínio raiz.

#nisgrpadm -a admin.dom.br. rootmaster.dom.br.	Acrescenta o <i>Root Master</i> ao grupo <i>admin</i> .
#/usr/lib/nis/nisupdkeys dom.br.	Atualiza as chaves ¹³ para o diretório raiz. Atualiza as chaves para as tabelas.
#/usr/lib/nis/nisupdkeys org_dir.dom.br.	Atualiza as chaves para os grupos.
#/usr/lib/nis/nisupdkeys groups_dir.dom.br.	
#nis_cachemgr	Ativa a gerência de memória do NIS+.
#ps -ef grep rpc.nisd #kill -9 process-id #rpc.nisd	Reativa o controlador de pedidos com o nível de segurança 2.
#nisaddcred -p UID -P topadmin.dom.br. local	Cria credenciais locais (LOCAL) para o Administrador.
#nisaddcred -p unix.UID@dom.br -P topadmin.dom.br. des	Cria credenciais DES para o Administrador. (Entre com a senha de <i>login</i>)
Enter login password:	
#nisaddcred membros...	Cria credenciais para outros membros do grupo <i>admin</i> .
#nisgrpadm -a admin.dom.br. membros...	Adiciona outros membros ao grupo <i>admin</i> .

OBS. O arquivo /etc/passwd deve conter entradas para todos os membros do grupo *admin*.

4.2.6 - Criando uma Servidora Replica

Sempre que quisermos aliviar o trabalho da *Master Server* devemos criar uma (ou mais) *Replica Server*. A presença de uma *Replica Server* no domínio agiliza a distribuição das informações do NIS+.

Para se criar uma *Replica Server* para o domínio *dom.br.* deve-se¹⁴ :

```
RootMaster#nismkdir -s rootreplica dom.br.
RootMaster#nismkdir -s rootreplica org_dir.dom.br.
RootMaster#nismkdir -s rootreplica groups_dir.dom.br.
```

¹³ As chaves servirão para a verificação das credenciais no processo de criptografia.

¹⁴ rootreplica é o nome da *Replica Server*.

4.2.7 - Adicionando um cliente a um domínio já existente

Caso o Sistema Operacional tenha sido instalado e deseja-se incluir o Cliente no domínio de NIS+ deve-se :

```
NisClient#nisclient -i dom.br.
```

4.3 - DNS

Para acessarmos serviços de conexão com outros Sistemas (*telnet,ftp,rlogin...*), devemos fornecer algum parâmetro que especifique o endereço do Sistema com o qual queremos estabelecer a comunicação. Este endereço pode ser obtido de duas formas¹⁵ :

- *IP Address.*
- *Hostname.*

O DNS (*Domain Name Service*) possibilita o Sistema “resolver” como encontrar o endereço de um Sistema qualquer, sem precisar colocá-lo nem no mapa local do Sistema (*/etc/hosts*), nem no NIS (o que seria inviável).

A configuração de um DNS consiste, basicamente de quatro arquivos, no mínimo, à saber:

- */etc/named.boot* - Arquivo que contém a configuração básica do DNS.
- */var/named/named.cache* - Possui os endereços IPs dos “root servers”.
- */var/named/named.hosts* - Contém a configuração das máquinas locais
- */var/named/named.rev* - Configuração de IPs das máquinas locais em notação reversa.

Descreveremos cada um dos arquivos à seguir:

/etc/named.boot

Consiste de uma linha contendo o diretório a partir do qual se encontram os arquivos com os dados do servidor e mais três linhas (normalmente) contendo três campos cada.

O primeiro campo, indica o tipo de informação naquela linha, podendo ser: *directory, cache, primary*, ou *secondary*, onde¹⁶ :

¹⁵ Assumindo o protocolo TCP/IP.

¹⁶ Na verdade existem também os tipos *forwarders* e *slave*, que não tem muita importância para nós aqui.

- directory* — Diretório a partir do qual se encontram os arquivos de configuração do servidor.
- cache* — A linha indicará os IPs que permanecerão no “cache” do servidor, normalmente consistindo dos “root servers”.
- primary* — Esta linha declara o servidor de DNS como “responsável” principal pelo domínio especificado no segundo campo.
- secondary* — Este campo indica ao servidor que ele deve procurar um “servidor secundário” para obter endereços neste domínio.

O segundo campo, exceto no caso da linha *directory* onde conterà o diretório dos arquivos do servidor, contém o domínio sobre o qual o arquivo conterà informações.

Este campo pode ter três formatos distintos, como em exemplos à seguir:

- . — Representa um domínio raiz.
- cat.cbpf.br* — Domínio em formato de nome.
- 0.0.127.in-addr.arpa — Domínio para indicar o endereço *loopback*.
- 253.84.152.in-addr.arpa — Domínio em notação IP reversa.

O terceiro campo conterà o nome do arquivo que possui a informação acerca do domínio especificado (*named.hosts*, *named.rev*, etc..).

Exemplo de um */etc/named.boot*:

```

; Arquivo de boot para servidor DNS
;
directory /var/named
;
; Tipo      Domínio      Arquivo associado
cache      .             named.cache
primary    0.0.127.in-addr.arpa  named.local
primary    253.84.152.in-addr.arpa  named.rev
primary    cat.cbpf.br.           namd.hosts

```

/var/named/named.cache

Possui endereços IPs dos servidores raiz (servidores que contêm informação acerca de domínios não locais, tais como **nic.cerf.net** ou **www.puc-rio.br**. Dessa maneira, qualquer endereço terminado em algo diferente de **cbpf.br**, por exemplo, necessitará de uma consulta à esses servidores.

Esse arquivo, consiste de conjuntos de duas linhas, da seguinte forma:

```

.           9999999   IN   NS   NS.NIC.DON.MIL
NS.NIC.DON.MIL 9999999   IN   A   26.3.0.103

```

O formato das linhas se “encaixa” no formato dos RR’s (*Resource Records*) indicados à seguir. Sendo que cada servidor de nomes (NS) tem que ter um IP associado (através de um *record* tipo A).

`/var/named/named.hosts` e `/var/named/named.rev`

Os dados contidos nesse arquivo são separados em campos de informação (RR). Eles possuem informações do tipo, que máquina corresponde a cada IP, quem tratará o *mail* para determinada máquina e nomes alternativos para as máquinas (*aliases*).

Os RR’s contêm um formato comum:

```
[domínio] [ttl] [classe] tipo rdata
```

Que significam:

[domínio]	—	Nome do domínio sobre o qual o RR se aplica
[ttl]	—	“Time to Live”, é o tempo durante o qual a informação permanece válida após a consulta ao servidor.
[classe]	—	Classe de endereço. (IN para endereços do tipo Internet) Se nenhuma classe for especificada a classe do RR anterior é assumida como válida
tipo	—	Este campo descreve o tipo de RR. Os tipos comuns são A, SOA, PTR e NS, que serão descritos à seguir
rdata	—	Contêm o dado associado ao RR. O formato deste campo depende do tipo de RR

4.3.1- Tipos de RR

Os tipos de RR podem ser:

4.3.1.1 - SOA

Este tipo descreve uma “zona de autoridade” (SOA significa “Start of authority” ou Início de autoridade), indicando que o rdata à seguir contém informação confiável (também chamada de autoritária) sobre o domínio. O rdata, por sua vez é da seguinte forma, por exemplo:

Onde **cbpfsu1.cat.cbpf.br.** indica o endereço do servidor de nomes, **marita.cbpfsu1.cat.cbpf.br** é o endereço do responsável pelo domínio (com um ponto substituindo o @), os outros campos são :

— **serial** é o número de série da informação contida no arquivo. Este valor deve ser acrescentado sempre que forem feitas mudanças no arquivo, de maneira que os servidores secundários possam saber se houve alterações apenas consultando este RR.

— **refresh** é o intervalo de tempo que este servidor (caso seja servidor secundário) deve esperar para consultar o registro SOA dos servidores primários para saber se houve mudanças.

— **retry** indica quanto tempo este servidor deve esperar para tentar novamente um “refresh” caso este falhe (em virtude de problemas de rede, por exemplo).

— **expire** especifica o tempo que o servidor deve esperar para descartar a informação em seu *cache* caso não tenha conseguido contactar um servidor primário

— **minimum** é o “ttl” default para qualquer RR que não possua esta informação (ver acima)

Exemplo de RR tipo SOA:

@	IN	SOA	cbpfsu1.cat.cbpf.br. (marita.cbpfsu1.cat.cbpf.br. 16 ; serial 86400 ; refresh:uma vez por dia (86400 seg.) 3600 ; retry: uma hora 3600000; expire: 42 dias 604800 ; minimum: 1 semana)
---	----	-----	---

No exemplo acima¹⁷, o campo [domínio] é substituído por um @, que equivale ao domínio especificado para este arquivo no */etc/named.boot*.

4.3.1.2 - A

Este tipo de RR associa um endereço IP à um “host”. O campo rdata conterá o endereço IP em notação numérica.

Exemplo:

atomo	IN	A	152.84.253.16
lepton.cat.cbpf.br.	IN	A	152.84.253.17

¹⁷ Os parênteses indicam que a informação do campo continua na linha seguinte.

Note que na primeira linha especificamos apenas o nome da máquina, pois um endereço sem um ponto no final, indica um endereço relativo ao domínio especificado no */etc/named.boot*. Já na segunda linha colocamos o endereço completo, com um ponto no final, significando um endereço absoluto.

4.3.1.3 - NS

Aponta para um servidor de nomes do domínio especificado.

Exemplo:

	IN	NS	cbpfsu1.cat.cbpf.br
--	----	----	---------------------

4.3.1.4 - CNAME

Associa um alias à um host, especificado por sua vez através de um RR tipo A.

Exemplo:

cat1	IN	A	152.84.253.10
www	IN	CNAME	cat1

Note que www é um alias para cat1.

4.3.1.5 - PTR

Esse tipo de RR é usado para associar endereços IP à hosts em arquivos usando notação reversa. Se, por exemplo, indicarmos no arquivo */etc/named.boot* uma linha da forma:

primary 84.152.in-addr.arpa named.rev

Teremos no arquivo */var/named/named.rev*, por exemplo:

; Hosts no domínio 152.84.252.XXX			
;			
130.252	IN	PTR	www-cfc.cat.cbpf.br.
131.252	IN	PTR	sirius.cat.cbpf.br.
; Hosts no domínio 152.84.253.XXX			
;			
10.253	IN	PTR	cat1.cat.cbpf.br.
11.253	IN	PTR	spin.cat.cbpf.br.

Note que o exemplo supõe que o servidor possui autoridade sobre o domínio 152.84.XX.XX. Caso a autoridade do servidor se estendesse apenas à dois subdomínios (152.84.252.XX e 152.84.253.XX), poderíamos ter, ao invés o seguinte no */etc/named.boot*:

primary 252.84.152.in-addr.arpa named.rev.252

primary 253.84.152.in-addr.arpa named.rev.253

E os arquivos:

/var/named/named.rev.252

; Hosts para o domínio 152.84.252.XX			
;			
130	IN	PTR	www-cfc.cat.cbpf.br.
131	IN	PTR	sirius.cat.cbpf.br.

e

/var/named/named.rev.253

; Hosts para o domínio 152.84.253.XX			
;			
10	IN	PTR	cat1.cat.cbpf.br.
11	IN	PTR	spin.cat.cbpf.br.

4.3.1.6 - MX

Indica quem “tratará” os mails para este “host”.

Exemplo:

quark.cat.cbpf.br	IN	A	152.84.253.18
	IN	MX	0 cbpfsu1.cat.cbpf.br.

Aqui temos um campo à mais (contendo o valor zero) que indica a ordem de preferência para o servidor de mail, podendo-se dessa forma especificar mais de um *mail server* para um determinado host.

Note-se também que na linha do registro MX, não especificamos o domínio, de maneira que o servidor assumirá que se trata do último host especificado (no caso quark.cat.cbpf.br). Isto ocorre também para os tipos CNAME, e HINFO que veremos à seguir.

4.3.1.7 - HINFO

Este tipo indica que forneceremos informações sobre a máquina ou domínio especificado.

Exemplo:

beta	IN	A	152.84.253.19
		HINFO	Sun/Sparc4 Solaris2.5.1

Após o tipo HINFO temos dois campos no qual normalmente colocamos o *hardware* e o *software* usados neste host.

Note que aqui além de omitirmos o domínio na linha, omitimos também a classe do RR. Novamente o servidor assume o último RR válido para obter o domínio e a classe.

No sistema Solaris 2, após configurarmos o servidor de nomes, devemos editar o arquivo `/etc/nsswitch.conf` nos clientes e alterar a linha que contém “hosts: nis [NOTFOUND=return] files” para adicionarmos após o nis, o parâmetro dns.

Após isso, devemos criar o arquivo `/etc/resolv.conf` que conterà, no caso do servidor em si (por exemplo):

```
domain cat.cbpf.br.  
nameserver 127.0.0.1
```

E no caso dos clientes:

```
domain cat.cbpf.br.  
nameserver xx.yy.zz.rr
```

Onde `xx.yy.zz.rr` é o número IP do servidor.

4.3.2 - Atualização de Informações

Após concluir qualquer alteração nos arquivos de configuração do DNS, deve-se reiniciar o *daemon* **in.named** para validar a nova informação.

5 - Manutenção do Sistema

5.1 - Gerenciando a Estrutura do Sistema

O *Solaris* traz uma novidade que é uma verdadeira “mão na roda” para os Administradores de Sistemas. Esta novidade é a ferramenta *Admintool*, com ela é possível incluir, deletar e modificar contas de usuários de forma muito interativa. Todas as tabelas do NIS+ estão disponíveis para serem configuradas de acordo com a necessidade do Administrador.

Para acessar a ferramenta deve-se :

```
RootMaster#admintool
```

5.2 - Gerência de Processos

A definição de **processo** é “qualquer unidade em execução dentro de um Sistema”. Para obter-se uma listagem dos processos que estão rodando no Sistema :

```
#ps -ef (Bourne Shell)
```

```
#ps -aux (C Shell)
```

O Administrador deve periodicamente usar este recurso para saber quais os processos que podem estar “pendurados” no Sistema (processos que permaneceram indevidamente no Sistema). Muitas vezes processos executados de forma inadvertida tornam as coisas ruins para todos os usuários do Sistema. É papel do Administrador retirar estes processos do Sistema. Para “matar” um processo, deve-se :

```
#kill -9 [PID]
```

Uma boa maneira de educar os usuários é fazer uso dos comandos :

- *nice*
- *background / foreground (bg/fg)*
- *batch*
- *at*

5.3 - Gerência de Dispositivos

Durante a vida de um Sistema, em vários momentos será interessante (e necessário), adicionar dispositivos para torná-lo mais adequado as necessidades de seus usuários e Administradores.

Inicialmente é necessário fazer a distinção entre o Dispositivo Lógico e Físico.

5.3.1 - *Physical Devices*

São os *raw devices*, estão localizados no diretório `/devices` e são construídos na inicialização do Sistema (*Open Boot*), através de informações sobre :

- Tipo de dispositivo.
- Em que porta (paralela / serial) que ele está conectado. etc...

Exemplo :

```
/devices/iommu@0,10000000/sbus@0,10001000/espdma@5,8400000/esp@5,8800000/sd@3,0:a
```

Refere-se à partição 0 (zero) do disco de *boot* do Sistema. É a localização física do dispositivo dentro do Sistema.

5.3.2 - *Logical Devices*

São *links* para acessar de forma correta os *raw devices*, contêm informações específicas e mais mneomônicas sobre o acesso a estes dispositivos.

Exemplo :

```
/dev/dsk/c0t3d0s0
```

É um *link* para o *physical device* mencionado anteriormente.

5.3.3 - Nomenclatura

5.3.3.1 - Discos :

```
/dev/dsk/cWtXdYsZ (Z = slice; Y = driver number; X = physical bus target;  
W = logical controller).
```

5.3.3.2 - Fitas Magnéticas :

/dev/rmt/X A n (*n = nonrewinding; A = Density=lmh,comp,ultra;*
X = tape drivenumber)

5.3.4 - Gerenciador de Dispositivos (*Volume Managment*)

O *Solaris* vem com um gerenciador de dispositivos de *diskettes* e *CD-ROM* exclusivamente, chamado *Volume Manager*. Este aplicativo possui inclusive uma Interface Gráfica que é automaticamente acionada quando se insere um volume no dispositivo correspondente. O usuário através do *OpenWindows* interage diretamente com o conteúdo do volume.

5.3.5 - *Tape Devices*

Os dispositivos de Fita Magnética têm inúmeras utilidades :

- *Backups*
- Armazenamento de Dados muito volumosos. etc...

Para instalar um dispositivo de Fita Magnética, deve-se seguir alguns passos importantes :

- *touch /reconfigure*
- *shutdown*
- *conectar o dispositivo*
- *boot*

A tabela a seguir mostra o número que o dispositivo leva, assumindo estarmos utilizando barramento SCSI :

Dispositivo	<i>SCSI target</i>
Primeiro Disco	3
Segundo Disco	1
Terceiro Disco	2
Quarto Disco	0
Primeira Unidade de Fita Magnética	4
Segunda Unidade de Fita Magnética	5
CD-ROM	6

5.4 - Preparando o *Backup*

O procedimento de *backup* é, sem dúvida, uma das mais importantes tarefas de um Administrador. Apesar de simples, pode ser um “salva-vida” em certas situações.

Existem três principais razões para se fazer o *backup* de um Sistema :

- Assegurar a Integridade do *FileSystem* contra um possível *crash* do Sistema.
- Garantir a recuperação de arquivos de usuários e de Sistema, em caso de uma deleção acidental.
- Assegurar a possibilidade de reaver todo o conjunto de *FileSystems*, em caso de um *Upgrade*, ou uma reinstalação mal feita.

Para tornar o procedimento de *backup* totalmente seguro é preciso tomar algumas medidas :

- Definir o Tipo de *backup* a ser feito.
- Certificar-se da compatibilidade dos comandos de *backup*¹⁸
- Definir quais os *FileSystems* a serem arquivados.
- Escolher o dispositivo a ser utilizado. (Fita Magnética, DAT etc...)
- Estabelecer com que frequência será feito o *backup*.

5.4.1 - Tipos de *backup*

Existem basicamente dois tipos de *backup* :

- *Full*
- *Incremental*

O tipo do *backup* é especificado, no comando *ufsdump*, de acordo com o parâmetro *level*. Este parâmetro pode assumir valores de 0-9. O valor ‘0’ indica o *Full backup*, ou seja, especifica que todo o *FileSystem* deve ser salvo. Os valores de 1 a 9 estabelecem a política de *Incremental backup* que copia apenas os arquivos que sofreram alteração depois do último *backup* de valor inferior.

As datas referentes à *backups* já realizados ficam guardadas no arquivo */etc/dumpdates*, isso quando especificada a opção “u” para o comando *ufsdump*.

¹⁸ Certos comandos podem exigir certas condições que tornam a recuperação inviável.

5.4.2 - Comandos

No SunOS5 o comando para copiar um *FileSystem* para uma unidade de Fita Magnética é o “*ufsdump*”.

Para reaver um *Filesystem* da unidade de Fita Magnética para o disco deve-se utilizar o “*ufsrestore*”. este comando exige que o diretório corrente em sua chamada seja o “pai” do *FileSystem* selecionado. Em outras palavras, para reaver o *FileSystem* /var deve-se executar o comando estando no diretório raiz do Sistema ‘/’. (*Root Directory*)

Exemplo :

```
#pwd
/
#/usr/sbin/ufsdump Of /dev/rmt/0n /var
```

O comando acima copia todo o *FileSystem* /var para o *Tape Device* apontado por /dev/rmt/0n.

Para reaver um arquivo localizado neste *FileSystem* devemos seguir os seguintes passos :

- Inserir a Fita utilizada para o *backup* do *FileSystem* /var na Unidade de Fita Magnética.
- Posicionar a cabeça de leitura da unidade no início do *FileSystem*, utilizando o comando “mt”.
- Interagir com o ambiente gerado pelo comando “ufsrestore” com a opção ‘i’ (*Interaction*), selecionando o arquivo desejado.

5.4.3 - Escolha do *FileSystem*

Existem algumas áreas dentro do Sistema que são de prioridade máxima em qualquer estratégia de *backup* :

- Área de usuários.
- O disco contendo os *FileSystems* que são alterados periodicamente (/ /var /etc /usr).

Já alguns *FileSystems* não requerem procedimento de *backup*. Por exemplo, uma área distribuída pelo Sistema na qual usuários de um grupo de trabalho guardam arquivos temporários. Ou seja, a escolha do *FileSystems* a serem salvos depende, em parte, da configuração do Sistema.

5.4.4 - Escolha do Dispositivo

Tradicionalmente o Dispositivo utilizado no procedimento de *backup* em quase todos os Sistemas é a Unidade de Fita Magnética. Isso porque ainda é o meio mais seguro e confiável de salvar dados.

Para acessar um dispositivo o Sistema Operacional recorre à Controladora do Dispositivo. A tabela a seguir lista as *Tape Drivers* que o *Solaris 2.x* dispõe :

<i>Tape Format</i>	<i>Model</i>
1/4"	<i>Archive Viper QIC - 150 streaming tape drive</i>
1/2"	<i>Emulex MT-02 tape controller</i>
4mm	<i>HP -88780 tape driver</i>
8mm	<i>Archive Python DAT tape subsystem</i>
	<i>Exabyte ESB-8200/8500 cartridge tape</i>

OBS. A *Sun* não garante compatibilidade com controladoras não fornecidas por ela.

5.4.5 - Dia de *backup*...

A frequência com que será feito o procedimento de *backup* depende diretamente do tipo de *FileSystem*. Por exemplo, a área de usuários do Sistema certamente sofre modificações mais frequentes do que o *FileSystem* /etc.

Existem várias Tabelas de Padronização da frequência do procedimento de *backup*¹⁹. Uma sugestão seria :

	Seg	Ter	Qua	Qui	Sex
Primeiro dia de mês, <i>Full backup.</i>					
Primeira semana	9	9	9	9	5
Segunda semana	9	9	9	9	5
Terceira semana	9	9	9	9	5
Quarta semana	9	9	9	9	5

Seguindo a tabela acima teríamos todas as sextas-feiras um *backup* de tudo que foi alterado desde o começo do mês, e teremos cópias diárias dos arquivos alterados.

¹⁹ O manual AnswerBook traz vários exemplos destas tabelas.

5.5 - Shell Scripts

As operações envolvidas no cotidiano de um Administrador de Sistemas podem tornar-se mais complexas de acordo com o número de máquinas em sua Rede. Uma tarefa feita em 20 Sistemas é tecnicamente invariável para um único Sistema, o que muda é a forma que a tarefa passará a ser executada. Imaginem por exemplo trocar a *password* de Supervisor em todas as Clientes de uma Rede com 30 (o que não é muito) Sistemas! Se essa tarefa fosse feita em um Sistema por vez estaríamos desperdiçando um tempo enorme.

Para contornar este tipo de situação escrevemos *scripts* que auxiliarão o Administrador a realizar suas tarefas com maior rapidez, e principalmente confiabilidade.

Scripts, como o nome já diz, são arquivos do tipo texto contendo comandos, organizados geralmente de forma sequencial, que determinam a ordem de execução das tarefas.

5.5.1 - A Escolha do Shell

O *Solaris* dispõe de três *UNIX Shells* :

- *csh* (*C-Shell*)
- *sh* (*Bourne-Shell*)
- *ksh* (*Kourne-Shell*)

A escolha do shell pode ser feita através da linha de comando ou no próprio *script*.

Exemplo :

Pela linha de comando :

```
#csh program1
```

No arquivo²⁰ "program1" :

```
#!/bin/csh
```

O *Shell* tradicionalmente usado para se escrever *scripts* é o *Bourne-Shell*. O *ksh* é um *shell* relativamente novo e veio com a intenção de padronizar a utilização do *Bourne-*

²⁰ O Arquivo deverá ter permissão de execução.

Shell, no qual ele se baseia, trazendo novos e poderosos recursos, como por exemplo a utilização de *Arrays*.

Vamos utilizar exemplos de *Shell Scripts* para começarmos a programar em UNIX.

Exemplo 1 :

```
#!/bin/sh
#
# Isto e' um comentario
#
for host in $*
do
    rsh $host reboot
    echo $host
done
#
```

Este *script* recebe como entrada os nomes dos clientes a serem reinicializados. Utilizamos aqui o recurso de *remote-shell* (para rodar um comando remotamente nos Clientes selecionados). Também imprime na tela o nome dos clientes para assegurar a execução correta de um comando tão delicado.

Exemplo 2 :

```
#!/bin/ksh
#
# Mário Vera (mvera@lafex.cbpf.br)
#
clear

echo ""
echo "      >> Starting the job <<"
echo ""
echo ""
```

```
/usr/ucb/echo -n "Enter the Number of Input Tapes : "
```

```
read NumofTapes
```

```
TapeNumber=1
```

```
while [ $TapeNumber -le $NumofTapes ]
```

```
do
```

```
    echo "Type the LogFile Name for tape $TapeNumber :"
```

```
    read LogFile[$TapeNumber]
```

```
    TapeNumber=`expr $TapeNumber + 1`
```

```
done
```

```
echo ""
```

```
TapeNumber=1
```

```
while [ $TapeNumber -le $NumofTapes ]
```

```
do
```

```
    echo "Tape Number \"$TapeNumber\" : "
```

```
    echo ""
```

```
    csh strip.setup
```

```
        nohup lixo > logfiles/${LogFile[$TapeNumber]} 2>>
logfiles/${LogFile[$TapeNumber]} &
```

```
    echo ""
```

```
    echo "Running strip !!!"
```

```
    echo ""
```

```
    wait
```

```
    echo "Finished this tape."
```

```
    echo ""
```

```
    echo "Trying to get a new input tape."
```

```
    echo ""
```

```
    mt -f /dev/rmt/0n offline
```

```
    wait
```

```
    sleep 100
```

```
TapeNumber=`expr $TapeNumber + 1`

done

echo "Execution Finished ! No more input tapes !"

exit

#
```

Calma ! Não é tão complicado como parece. Este *Script* roda um programa que tem como entrada dados armazenados na Unidade de Fita Magnética²¹, e guarda possíveis mensagens de erros nos arquivos *logfile*s que foram especificados pelo usuário.

OBS. Reparem na linha 21 o comando :

csH strip.setup

Como este arquivo (strip.setup) contém código escrito em *csH*, tivemos que chamar o *C-Shell* para interpretá-lo.

Finalmente, um *script* que manda um *e-mail* para todos os usuários que fazem parte do NIS+ :

```
#
# O Shell neste exemplo é indiferente.
#
mail `niscat passwd.org_dir | awk -F: '{print $1}' | grep -v root | grep -v
nobody | grep -v bin | grep -v uuCP | grep -v ingres | grep -v sundiag |
grep -v sysdiag | grep -v audit | grep -v news | grep -v sys | grep -v
daemon` < $1 >& /dev/null
```

```
#
```

Com o tempo o Administrador começará a se familiarizar com os comandos do UNIX e perceberá que o *Shell* oferece uma verdadeira linguagem de Programação que facilita em muito a Manutenção do Sistema.

²¹ Um programa deste tipo é conhecido como *striper*.

6 - Bibliografia :

1. *Solaris 2.4 Answer Book Manual, Sun Microsystems, Inc.*
2. *Silberschatz, A. and Peterson, J. : Operating Systems Concepts 3rd edition.*
3. *Tanenbaum, A. : Computer Networks 2nd edition.*
4. *Linux Documentation Project : Linux Administrators' guide 96.*

NOTAS TÉCNICAS é uma pré-publicação de trabalhos técnicos-científicos do CBPF.

Pedidos de cópias desta publicação devem ser enviados aos autores ou ao:

Centro Brasileiro de Pesquisas Físicas
Área de Publicações
Rua Dr. Xavier Sigaud, 150 – 4^o andar
22290-180 – Rio de Janeiro, RJ
Brasil

NOTAS TÉCNICAS is a preprint of technical reports from CBPF.
Requests for copies of these reports should be addressed to:

Centro Brasileiro de Pesquisas Físicas
Área de Publicações
Rua Dr. Xavier Sigaud, 150 – 4^o andar
22290-180 – Rio de Janeiro, RJ
Brazil