

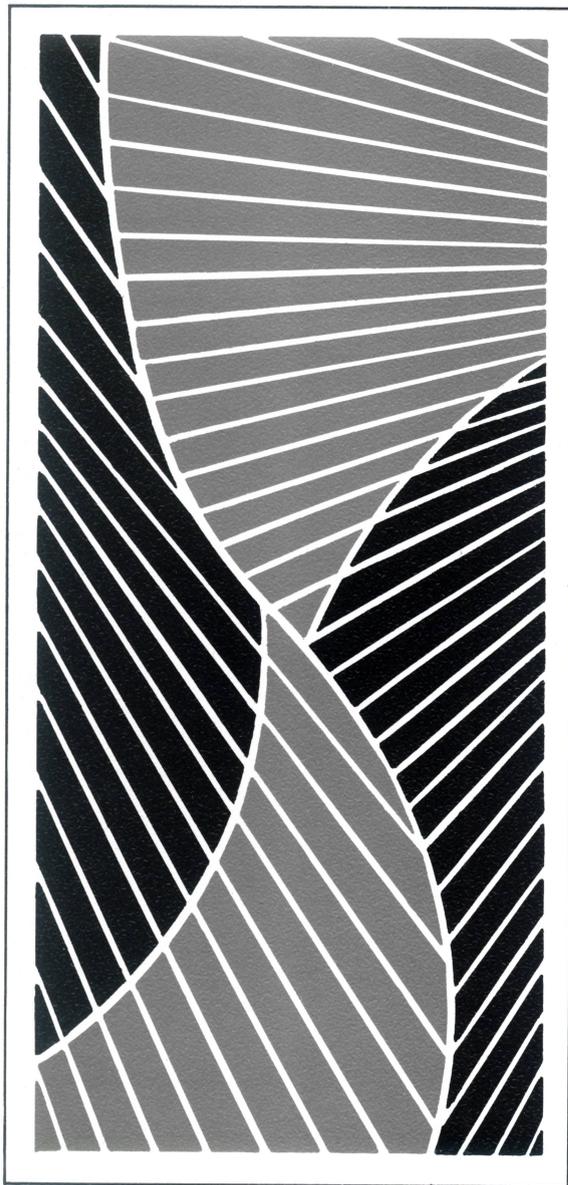
SOX

Manual do Administrador
do Sistema SOX



cobra

SOX



cobra



Computadores e Sistemas Brasileiros S/A

MANUAL DO ADMINISTRADOR DO SISTEMA SOX

SX0389-03.0

**MANUAL
ADMINISTRADOR
DO SISTEMA SOX**

3ª Edição

abril/89

SX0389-03.0

Este manual é uma publicação da Divisão de Testes e Integração de Sistemas / Setor de Documentação, aplicável a produtos comercializados pela Cobra.

O Fabricante reserva-se o direito de alterar a qualquer tempo, sem notificação prévia, as características e especificações técnicas do produto retratado nesta documentação, bem como promover, sempre que necessário, alterações e correções na forma e conteúdo desta publicação.

Cobra - Computadores e Sistemas Brasileiros S.A.

Todos os direitos reservados. Esta obra não pode ser comercializada, nem ter qualquer uso diferente do autorizado, nem pode também ser reproduzida, total ou parcialmente, por qualquer meio ou forma, sem prévia autorização escrita concedida pela Cobra.

Av. Comandante Guarany, 447 - Jacarepaguá
22785 Rio de Janeiro RJ

BRASIL

REGISTRO DE ATUALIZAÇÃO

Manual:	Emissão:
Administrador do Sistema SOX	abril/89

Esta página é uma orientação ao usuário quanto à identificação e aplicação específica deste manual, sua estrutura e o histórico das atualizações aplicáveis ao mesmo. Mantenha esta folha anexa ao início do manual, para controle de sua atualização.

APLICAÇÃO DO MANUAL:

Equipamento: LINHA X

Sistema: SOX

Produto: SOX

CONTROLE DE PÁGINAS DO MANUAL:

Cap. 1 - 2	Cap. 5 - 28	Anexo C - 7
Cap. 2 - 10	Cap. 6 - 14	Glossário - 1
Cap. 3 - 20	Anexo A - 3	
Cap. 4 - 12	Anexo B - 57	

HISTÓRICO DAS ATUALIZAÇÕES:

- 1 - Esta documentação encontra-se em sua terceira edição, motivada pela inclusão do Capítulo 6 referente ao Spool de Saída.

SISTEMA DE ATUALIZAÇÃO DE DOCUMENTAÇÃO:

Com o objetivo de manter os usuários sempre a par das constantes evoluções e melhorias de seus produtos, a COBRA dispõe de um Sistema de Atualização de Documentação (**SAD**), que consiste em um conjunto de arquivos de texto reunidos em um ou mais discos flexíveis denominados **altera-doc**.

Estes discos, que são fornecidos com o sistema operacional SOX, contêm as informações necessárias para a atualização dos manuais disponíveis para a linha de computadores X/SOX, enquanto a documentação correspondente ainda não foi atualizada ou reeditada.

Os arquivos do SAD pertencem ao diretório **/man/altera** e são de dois tipos distintos: os arquivos de atualização, que contêm as alterações a serem aplicadas a cada manual, e um arquivo **índice**, que identifica cada arquivo de atualização existente no diretório.

Os arquivos que descrevem as atualizações de cada manual são formados por blocos de informações, cada um dos quais contêm as alterações liberadas numa mesma data. É dividido em duas partes: a primeira, com informações gerais e um sumário das alterações aplicáveis ao manual, e a segunda, com as alterações propriamente ditas, precedidas de informações que possibilitam sua localização no manual.

Já o arquivo **índice**, que deve ser o primeiro a ser consultado, contêm as seguintes informações:

- No cabeçalho: Número sequencial e data de emissão das atualizações. Indica também qual o sistema e linha de produtos a que o SAD se aplica.

- Títulos dos manuais disponíveis para essa linha de produto. Observar, entretanto, que alguns dos manuais listados referem-se a produtos de software opcionais, não fornecidos com o pacote básico do SOX.
- Códigos dos manuais correntemente em vigor, e para os quais se aplicam as atualizações emitidas. Através desse código, o usuário pode conferir se as edições de seus manuais são as mais atualizadas. Qualquer problema nesse sentido poderá ser esclarecido junto ao representante ou filial COBRA mais próximo.
- Nomes dos arquivos, tal como constam no diretório /man/altera, que contém as atualizações aplicáveis a cada manual. Estes nomes são os que deverão ser informados ao sistema, para acesso a cada um dos mesmos.
- Número seqüencial e data de emissão de cada atualização já liberada para os manuais indicados. Observar que estas atualizações são sempre complementares para uma mesma edição do manual, ou seja, uma atualização mais recente não invalida ou inclui uma atualização anterior.

O SAD é um mecanismo de atualização ágil e dinâmico, mas de caráter transitório. A medida que as atualizações do SAD vão sendo incorporadas às novas revisões e edições dos respectivos manuais, ou novas alterações se façam necessárias a uma documentação já liberada, os arquivos de atualização dos discos altera-doc vão sendo alterados correspondentemente, dando origem a novas versões do SAD.

A implantação e uso deste sistema se faz através de um comando especial do SOX, denominado am, que é descrito no "Manual de Referência Comandos SOX" (SX01).

Página intencionalmente em branco.

Este manual destina-se basicamente a dois grupos de leitores:

- . Ao usuário do SOX que tem como atividade administrar os recursos disponíveis na sua instalação.
- . Ao usuário que tenha interesse em obter conhecimentos específicos sobre configuração, ativação, bem como sobre o sistema de arquivos do SOX.

Supõe-se que os leitores deste manual estejam familiarizados com a utilização dos comandos do SOX, e também com o interpretador de comandos (Shell).

Este manual é constituído de cinco capítulos e três anexos.

O Capítulo 1 introduz a figura do Administrador do Sistema, apresentando as tarefas que devem ser por ele executadas, e os cuidados que devem ser por ele tomados.

O Capítulo 2 apresenta o Sistema de Arquivos do SOX. É mostrada a sua estrutura, os tipos de arquivo, a relação entre arquivos especiais e dispositivos físicos e os problemas que podem ocorrer no Sistema de Arquivos.

O Capítulo 3 apresenta a Ativação do Sistema. Nele se encontram breves descrições da carga do sistema e da ativação de terminais, bem como descrições detalhadas dos programas envolvidos na ativação de terminais.

O Capítulo 4 apresenta procedimentos para a realização das tarefas do Administrador do Sistema.

O Capítulo 5 apresenta o Configurador do SOX. Nele encontram-se as informações necessárias para que o Administrador possa adaptar o SOX às necessidades de sua instalação.

O Capítulo 6 apresenta o Spool de Saída. Nele estão as informações necessárias ao Administrador, para o gerenciamento dos recursos de impressão da instalação.

O Anexo A descreve os componentes básicos do Sistema de Arquivos do SOX, apresentando seus principais diretórios.

O Anexo B apresenta os arquivos manipulados e utilizados pelo Administrador do Sistema.

O Anexo C contém a descrição dos discos lógicos e partições que devem ser conhecidos pelo administrador do SOX.

CAPÍTULO 1 - O ADMINISTRADOR DO SISTEMA

1.1	Funções do Administrador . . .	1-1
1.2	Privilégios do Administrador . .	1-2

CAPÍTULO 2 - O SISTEMA DE ARQUIVOS

2.1	Estrutura Hierárquica e Tipos de Arquivos	2-1
2.2	Volumes	2-2
2.3	Dispositivos Físicos e Arquivos Especiais Bloco e Caractere . .	2-5
2.4	Erros no Sistema de Arquivos .	2-8

CAPÍTULO 3 - CARGA DO SISTEMA

3.1	Seqüência de carga do SOX . . .	3-1
3.2	Seqüência de Ativação de Terminais	3-2
3.3	Programas INIT, GETTY e LOGIN .	3-2
3.3.1	INIT	3-2
3.3.1.1	Introdução	3-2
3.3.1.2	O Arquivo /etc/inittab	3-4
3.3.1.3	Ativação do INIT	3-7
3.3.1.4	O Comando INIT	3-9
3.3.1.5	O Arquivo /etc/rc	3-10
3.3.2	GETTY	3-11
3.3.2.1	Introdução	3-11
3.3.2.2	O Comando GETTY	3-11
3.3.2.3	O Arquivo /etc/gettydefs . . .	3-15
3.3.2.4	O Arquivo /etc/issue	3-19
3.3.3	LOGIN	3-19
3.4	Desativação do Sistema	3-20

CAPÍTULO 4 - TAREFAS ADMINISTRATIVAS

4.1	Cadastramento de Novos Usuários	4-1
4.1.1	Introdução	4-1
4.1.2	O Arquivo /etc/passwd	4-2
4.2	Comunicação com Usuários . . .	4-4
4.3	Verificação e Recuperação de Sistemas de Arquivos	4-5
4.4	Salvamento e Recuperação de Sistemas de Arquivos	4-8
4.5	Comunicação do Sistema com o Operador	4-9
4.5.1	Intervenções	4-9
4.5.2	Erros	4-11
4.5.2.1	Erros no Sistema de Arquivo . .	4-11
4.5.3	O Processo-Console	4-12

CAPÍTULO 5 - CONFIGURADOR DO SOX - CONF

5.1	Introdução	5-1
5.2	O Arquivo conf.sox	5-2
5.2.1	Descrição	5-2
5.2.2	Parâmetros de Configuração do Núcleo	5-5
5.2.2.1	Descrição dos Parâmetros . . .	5-5
5.2.3	Parâmetros de configuração do Servidor	5-6
5.2.3.1	Descrição dos Parâmetros . . .	5-6
5.2.4	Parâmetros de Configuração da Máquina Real	5-8
5.2.4.1	Descrição dos Parâmetros . . .	5-8
5.3	Funcionamento do Programa CONF	5-12
5.3.1	Carga e Ativação	5-12
5.3.2	Leitura do Arquivo conf.sox . .	5-14

	Pág.	
5.3.3	Análise de Consistência do Arquivo conf.sox	5-14
5.3.4	Carga dos Módulos	5-15
5.3.4.1	Carga de Software em Controladores	5-16
5.3.4.2	Carga do Núcleo	5-18
5.3.4.3	Carga dos PMRs	5-18
5.3.4.4	Carga do Servidor	5-20
5.3.5	Final de Operação	5-20
5.4	Valores-padrão	5-22
5.4.1	Parâmetros do Núcleo	5-22
5.4.2	Parâmetros do Servidor	5-23
5.5	Mensagens de Erro/Advertência	5-24
5.5.1	Mensagens de Erro	5-24
5.5.2	Mensagens de Advertência	5-27

CAPITULO 6 - SPOOL DE SAÍDA

6.1	Introdução	6-1
6.2	Arquivos Auxiliares do Spool	6-2
6.2	Os Comandos	6-3
6.4	Iniciando/Finalizando o Spool	6-6
6.5	O Programa Impressor	6-7
6.6	Ambiente de Instalação	6-13

Anexo A - Diretório Básico do SOX

Anexo B - Arquivos de Administração

Anexo C - Discos Lógicos e Partições

Glossário

O ADMINISTRADOR DO SISTEMA

1.1 FUNÇÕES DO ADMINISTRADOR

O administrador do Sistema é responsável por:

- Adaptar o SOX às necessidades de sua instalação, configurando-o de forma adequada;
- Salvar e restaurar sistemas de arquivos;
- Verificar e recuperar sistemas de arquivos;
- Cadastrar novos usuários;
- Comunicar aos usuários do sistema eventos de seu interesse;
- Manter atualizados os arquivos utilizados na ad ministração do sistema;
- Controlar e manter o uso do sistema.

1.2 PRIVILÉGIOS DO ADMINISTRADOR

Devido à necessidade de acessar e manipular todos os recursos do sistema, o Administrador do Sistema tem que ter privilégios que não são dados aos demais usuários, como executar comandos especiais, ler e gravar arquivos, sem que permissões de acesso sejam testadas.

No SOX existe uma classe de usuários, denominados Superusuários, que possuem estes privilégios. É considerado Superusuário qualquer um que tenha identificação de usuário igual a zero. Normalmente, a sessão de nome root está associada a um Superusuário, devendo ser usada pelo Administrador, quando necessário.

Como os privilégios dados ao Administrador podem causar danos a outros usuários caso sejam mal utilizados, recomenda-se ao Administrador que se torne Superusuário apenas quando o uso de seus privilégios for estritamente necessário.

Existem três maneiras do Administrador se tornar Superusuário:

- . Quando o sistema é ativado em modo monousuário (ver Capítulo 3 deste manual) é assumido que o usuário é o Superusuário;
- . Abrindo uma sessão associada a um Superusuário, fornecendo a senha correta;
- . Quando, em uma sessão que não seja do Superusuário, for executado o comando su, e for fornecida a senha correta.

CAPÍTULO 2

O SISTEMA DE ARQUIVOS

2.1 ESTRUTURA HIERÁRQUICA E TIPOS DE ARQUIVOS

O sistema de arquivos no SOX é organizado na forma de uma árvore hierárquica, onde os diversos nós da árvore são os arquivos do sistema. Os dados contidos nos arquivos podem ser de naturezas diferentes, dependendo do tipo do arquivo.

Os tipos de arquivo existentes no SOX são:

- Regular - Arquivos que contêm dados e programas do usuário;
- Diretório - Arquivos compostos de entradas, onde cada entrada contém o nome e um ponteiro para um arquivo imediatamente abaixo na hierarquia;
- Especial bloco e caractere - Arquivos que correspondem a dispositivos de E/S;
- Especial FIFO - Arquivos que implementam um canal de comunicação entre processos.

O primeiro arquivo da hierarquia (de nível mais alto) é sempre do tipo diretório e é chamado de raiz do sistema de arquivos. Seu nome é / (barra).

Os nós não-folha da árvore hierárquica são do tipo diretório e os nós-folha são de qualquer tipo.

2.2 VOLUMES

São as entidades lógicas onde residem os sistemas de arquivo. Cada volume contém um sistema de arquivos.

. ESTRUTURA DE UM VOLUME

Um volume é dividido em blocos. O bloco do volume é a menor unidade de alocação e transferência. Na versão atual do SOX, o tamanho do bloco é fixo, tendo o valor de 1024 octetos.

As informações contidas em um volume podem ser divididas em duas categorias:

. Dados

São os blocos que contêm os dados dos arquivos.

. Controle

Informações usadas pelo sistema para controlar o volume. É formado por três estruturas de controle.

. Seção de informação

Contém informações gerais sobre o volume, como tamanho do bloco do volume, número de blocos livres, ponteiros para as outras estruturas de controle, etc.

. Área de BCAs

Um BCA (Bloco de Controle de Arquivo) é a estrutura que implementa a entidade arquivo. Contém informações sobre o arquivo, como tamanho do arquivo, ponteiros para os blocos do volume que pertencem ao arquivo, etc. A área de BCAs é uma seção do volume onde estão os BCAs dos arquivos. Um ponteiro para um arquivo, conhecido como ibca, é o índice nessa área do BCA correspondente ao arquivo.

. Mapa de blocos

É uma seção do volume que indica, para cada bloco do volume, se ele está livre ou ocupado (um bloco está ocupado se já estiver alocadado para algum arquivo ou estrutura de controle).

. OPERAÇÕES SOBRE VOLUMES

As principais operações sobre volumes são:

- . Criação;
- . Montagem;
- . Desmontagem.

A criação de um volume é feita através dos comandos iv ou mkfs (ver "Manual de Referência Comandos SOX") e consiste em criar as estruturas de controle e, pelo menos, o arquivo / do volume.

A montagem de um volume é feita através do comando **mount**. Montar um volume significa tornar o sistema de arquivos contido no volume disponível ao SOX. Este sistema de arquivos se torna disponível como subvolume do diretório especificado no comando **mount**. Este comando tem dois parâmetros: um arquivo do tipo especial bloco, que corresponde ao dispositivo físico onde será colocado o volume, e outro arquivo, do tipo diretório, que funcionará como "ponte" para o volume montado. Depois de montado o volume, qualquer referência ao arquivo "ponte" será considerada como uma referência ao diretório-raiz do sistema de arquivos contidos no volume.

Para se montar qualquer volume no SOX (isso vale também para o volume-raiz, já que este é montado automaticamente na carga do sistema), deve ter sido configurado um **buffer pool** que tenha tamanho de buffer igual ao tamanho do bloco do volume.

A desmontagem de um volume é feita através do comando **umount**. Desmontar um volume significa que o Sistema de Arquivos nele residente não estará mais disponível ao SOX, retornando o Sistema de Arquivos à configuração anterior à emissão do comando **mount**.

. VOLUME-RAIZ

É um volume que é montado automaticamente na carga do sistema, e permanece montado durante toda a operação do sistema.

2.3

DISPOSITIVOS FÍSICOS E ARQUIVOS ESPECIAIS BLOCO E CARACTERE

No SOX, um dispositivo físico está intimamente ligado ao conceito de arquivo especial bloco ou caractere. Para que um dispositivo físico possa ser acessado pelo SOX, é necessário que exista um arquivo especial bloco ou caractere, que esteja associado a ele. Cada dispositivo físico é manipulado pelo seu programa de controle que no SOX é também chamado de PMR. Em outras palavras, uma operação sobre um arquivo especial bloco ou caractere resulta em um pedido de serviço ao programa de controle de dispositivo físico associado. A um programa de controle de um dispositivo físico podem estar associados um ou mais arquivos especiais.

Em operações de E/S realizadas em arquivos especiais bloco, a transferência dos dados é feita através do **buffer pool** do sistema, o que não ocorre nas operações de E/S realizadas em arquivos especiais caractere.

A associação entre um programa de controle de dispositivo físico e um arquivo especial se dá através de um número chamado dispositivo, fornecido na criação do arquivo especial. Este número é composto de duas partes: tipo e subtipo. O tipo identifica o programa de controle de dispositivo associado e é um número na faixa de 0 a 239, enquanto que o subtipo é um número interpretado exclusivamente pelo programa de controle de dispositivo e está na faixa de 0 a 255. Cada tipo só pode estar associado a um programa de controle de dispositivo, embora o mesmo possa responder por um ou mais tipos.

Para se criar um novo dispositivo no SOX, os seguintes passos devem ser observados:

- 1 - Verificar se o programa de controle de dispositivo desejado está sendo carregado na inicialização do sistema (ver arquivo /conf.sox - Capítulo 5 deste manual).

Dois casos podem ocorrer:

- O programa de controle de dispositivo não está sendo carregado.

Neste caso, deve-se providenciar para que ele seja configurado (ver Capítulo 5 deste manual).

- O programa de controle de dispositivo já está carregado.
- 2 - Criar um arquivo especial caractere e/ou bloco, através do comando **mknod**. O parâmetro tipo fornecido ao comando deve ser igual ao tipo atribuído ao programa de controle de dispositivo na configuração, e o parâmetro subtipo deve ser definido de acordo com a especificação do programa de controle de dispositivo.

Por convenção, todos os arquivos especiais estão agrupados no diretório /dev, sendo seus nomes codificados com o formato:

(r)ddduu

tendo o seguinte significado:

r - O caractere "r" é usado para os dispositivos que podem ser acessados a bloco ou caractere. Se presente, indica caractere. Se ausente, bloco.

ddd - Contém uma informação sobre a natureza do dispositivo. Por exemplo, tty para terminal, dw para disco Winchester, etc.

uu - Informação adicional.

A tabela a seguir apresenta exemplos de nomes de arquivos especiais bloco e caractere, com seus respectivos tipos, e uma indicação se bloco ou caractere.

dispositivo	tipo	subtipo	arquivo (/dev)	bloco/caractere
terminal	22	0	/tty0	caractere
	22	1	/tty1	caractere
	22	2	/tty2	caractere
	22	3	/tty3	caractere
disco Winchester	01	0	/dw0	bloco
	17	0	/rdw0	caractere
disco flexível	00	0	/df0	bloco
	16	0	/rdf0	caractere

2.4 ERROS NO SISTEMA DE ARQUIVOS

As informações contidas nos volumes do SOX podem, em certos casos, sofrer algum tipo de deterioração, tornando, por consequência, o volume inconsistente.

Caso ocorra algum erro, podem ocorrer dois casos:

- . As informações defeituosas pertencem a áreas de controle

Se o erro for detectado pelo sistema, aparecerá, no terminal-console, mensagem indicando que o volume está defeituoso. Ver o formato da mensagem no item 4.5.2.1 deste manual.

Se o erro ocorrido não for detectado, as consequências são imprevisíveis, pois as informações erradas poderão se propagar tornando o volume completamente destruído.

- . As informações defeituosas pertencem à área de dados do volume

Neste caso, somente o arquivo ao qual a área pertence será afetado.

As causas mais freqüentes de ocorrência de erro são as seguintes:

- . Falhas de equipamento ou do meio de entrada/saída

Neste caso estão incluídos os defeitos de hardware nas unidades de disco e também as falhas nos meios de armazenamento. Os discos flexíveis, principalmente, estão sujeitos a erros desse tipo.

. Erros de operação

Qualquer processo que execute como superusuário pode ser considerado um "operador", na medida em que ele tem acesso a recursos restritos do sistema. Certos comandos de operação (por exemplo, `clri`, `fsdb`) podem causar grandes danos aos sistemas de arquivos, se forem mal utilizados. Programas com erro, feitos por usuários, se forem executados por superusuários, podem também causar danos (por exemplo, um programa que acesse um disco fisicamente pode destruir completamente as áreas de controle).

- . Quedas de força e desligamento da máquina sem realização do procedimento devido.

Se ocorrer uma queda de força, o sistema de arquivos pode não ser atualizado, devido à existência de um **buffer pool** no SOX. As últimas operações de escrita realizadas em um sistema de arquivo podem estar registradas apenas no **buffer pool**, não tendo sido gravadas ainda no momento da queda.

O comportamento do sistema em relação ao sistema de arquivos, no caso do operador desligar a máquina sem realizar o procedimento devido, será equivalente à uma queda de força.

Nestes casos serão duas as conseqüências:

As últimas operações de saída realizadas nos arquivos podem não ter sido efetivadas. O conteúdo do arquivo estará válido e consistente, embora não-atualizado. O sistema de arquivos estará também válido e consistente.

Cada vez que ocorre uma queda de força (ou se o volume for retirado da unidade sem ser desmontado), alguns blocos livres podem ser perdidos. Cada vez que isso ocorre, o volume perde espaço, até que fique completamente cheio.

Casos de queda de força não tornam o sistema de arquivos inconsistente.

Em qualquer das situações mencionadas acima, ou sempre que se suspeitar de que houve algum erro, o sistema de arquivos deve ser verificado e, se possível, recuperado. Os procedimentos de verificação e recuperação se encontram no Capítulo 4 deste manual.

CAPÍTULO 3

CARGA DO SISTEMA

Este capítulo apresenta:

- . A seqüência de ações executadas durante a carga do SOX;
- . A seqüência de ações necessárias à ativação de terminais;
- . Descrição detalhada dos programas **init**, **getty** e **login**;
- . Desativação do sistema.

3.1 SEQUÊNCIA DE CARGA DO SOX

- 1 - O programa **boot** é carregado em memória por algum programa de ativação.
- 2 - O programa **boot** carrega o programa Configurador do SOX a partir do arquivo /conf, executando-o em seguida.
- 3 - O programa Configurador do SOX carrega os módulos que compõem o sistema, a partir do disco de carga, bem como cria a infra-estrutura necessária à operação do SOX, a partir de parâmetros obtidos no arquivo /conf.sox. Faz parte desta fase a inicialização do Servidor SOX.
- 4 - Após a inicialização do Servidor SOX, é executado o programa **init**.

3.2 SEQUENCIA DE ATIVAÇÃO DE TERMINAIS

- 1 - Normalmente, o programa **init** dispara um processo **getty** para cada a terminal ser ativado, a partir de informações obtidas de entradas do arquivo **/etc/inittab**.
- 2 - Cada processo **getty** abre o arquivo especial **cor** respondente ao terminal a ser ativado, **exibe** **men** **sagens** identificando o sistema e pede ao usuário que **entre** com a sua identificação. A partir da leitura desta identificação e das entradas do **ar** **quivo** **/etc/gettydefs**, **getty** estabelece a **configu** **ração** adequada do terminal. Em seguida, **getty** **exe** **cuta** o programa **login**.
- 3 - O programa **login** verifica a validade da **identi** **cação** do usuário, consultando o **arquivo** **/etc/passwd**. Caso a identificação seja válida, é **disparado** o programa especificado pelo último **cam** **po** da entrada correspondente ao usuário no **arqui** **vo** **/etc/passwd** (em geral **/bin/sh**).

3.3 PROGRAMAS INIT, GETTY E LOGIN

3.3.1 **INIT**

3.3.1.1 Introdução

A cada instante, durante a operação, o SOX se encontra em um determinado "nível de operação". Um nível de operação pode ser considerado como uma **configura** **ção** dinâmica de software, em que apenas um **determina** **do** grupo de processos está ativo. **init** é responsável, após a carga do sistema e após uma transição de nível de operação, pela administração da configuração de software correspondente ao nível atual.

O nome **init** se referencia a duas entidades diferentes:

- Um processo especial do SOX, que é criado ao fim do carregamento do sistema, e permanece ativo durante todo o tempo em que o sistema estiver em operação.
- Um programa que corresponde ao arquivo /etc/init. Este programa é executado pelo processo **init**, e também pode ser executado por um Superusuário na forma de comando, e permite trocar dinamicamente o nível de operação.

O processo **init** é criado como último passo do procedimento de carga do SOX e, imediatamente após ser criado, executa o programa **init** e fica a ele associado até o fim de operação do sistema. O nome **init** será usado (a menos que explicitamente mencionado) para se referenciar tanto ao programa como ao processo.

A execução do **init** é controlada pelo nível de operação e pelo arquivo /etc/inittab. Nas transições de nível de operação, **init** percorre este arquivo e, para cada entrada que tenha um nível de operação correspondente ao nível de operação-alvo, cria um processo de acordo com os parâmetros obtidos da entrada. **init** destrói também os processos que foram por ele criados no nível de operação anterior, e que não estão especificados nas entradas correspondentes ao nível-alvo.

3.3.1.2 O Arquivo /etc/inittab

Este arquivo é constituído de entradas, sendo cada uma associada a um ou mais níveis de operação.

O procedimento-padrão é a existência de uma entrada na inittab para cada linha (terminal) que deve ser ativada numa configuração, associada à seqüência de programas **getty-login**.

O arquivo inittab é um arquivo ASCII, composto de entradas separadas pelo caractere **newline**. As entradas são compostas de campos posicionais, separados por : (dois pontos). Entradas que começam por : são ignoradas.

Uma entrada tem o seguinte formato:

ident:nivel:tipo:comando

Onde:

ident - Usado para identificar a entrada. Contém até quatro caracteres. Não pode ser vazio, linhas começadas por : (dois pontos) são ignoradas.

nivel - Contém o nível de operação da entrada. Pode ser composto dos caracteres [0,1,2,3,4,5,6,S,s]. Se este campo for vazio, será interpretado como "0123456".

. tipo - Contêm informações adicionais para o controle da execução da entrada. Seus valores podem ser:

. off

Indica entrada desativada.

. sysinit

Indica uma entrada que deve ser executada sempre antes de uma mudança para modo monousuário.

. boot

Indica que a entrada deve ser executada apenas uma vez, quando o **init** estiver entrando no nível de operação correspondente à entrada, se o **init** tiver sido ativado pela carga do sistema.

. bootwait

Tem o significado de boot, mais o fato de que o **init** deve esperar pelo término de sua execução.

. once

Indica que a entrada deve ser executada somente uma vez, se o **init** estiver entrando em um novo nível e se este corresponder ao nível da entrada.

. wait

Tem o significado de once, mais o fato de que o **init** deve esperar pelo término da execução.

. respawn

Indica que a entrada deve ser executada sempre que o sistema estiver no nível de operação correspondente à entrada, e se o processo criado para executar a entrada terminar, deve ser reexecutado.

. initdefault

Este tipo de entrada tem um significado diferente dos demais. Serve apenas para indicar ao **init** o nível de operação inicial a ser feita a carga do sistema.

comando - Indica o comando que o **init** irá fornecer para o interpretador de comandos sempre que a entrada for ativada. A cadeia contida neste campo será acrescida do prefixo **exec**, para que não seja criado um novo processo. O **init** irá criar um processo e executar

```
sh -c "exec comando"
```

Sendo assim, pode-se utilizar de todas as facilidades do interpretador de comandos.

Para se alterar a **inittab**, deve-se usar o editor. Pode-se alterar a **inittab** dinamicamente e, se for necessário que a alteração entre em vigor imediatamente, executa-se o comando **init** com parâmetro **Q**. Caso contrário, a modificação só entrará em vigor quando o **init** for ativado por alguma razão.

Um exemplo de uma inittab é dado a seguir:

```
: Exemplo de um formato da inittab.  
: Esta configuracao tem 4 terminais, mas so` dois  
: devem ser ativados.  
: O arquivo /etc/rc contem comandos que devem ser  
: executados antes do sistema entrar em operacao  
: normal, como backup's.  
: O ativacao do programa /etc/xxx esta` desativada.  
: O nivel de operacao sera` 2.  
nivi:2:initdefault:  
in:2:bootwait:/etc/rc>/dev/console</dev/console2>&!  
tty0:2:respawn:/etc/getty tty0 TI200-9.6 TI-200  
tty1:2:respawn:/etc/getty tty1 TI200-9.6 TI-200  
:tty2:2:respawn:/etc/getty tty2 TI200-9.6 TI-200  
:tty3:2:respawn:/etc/getty tty3 TI200-9.6 TI-200  
xxxx:2:off:/etc/xxx parametros
```

3.3.1.3 Ativação de INIT

O **init** é ativado quando ocorrem os seguintes eventos:

- . Carga do sistema;
- . Término de um filho, ou
- . Recebimento de um sinal.

Quando o **init** é ativado pela carga do sistema, ele procura na inittab uma entrada do tipo **initdefault**, para determinar o nível de operação inicial. Três situações podem ocorrer:

- 1 - Existe uma entrada do tipo `initdefault` no arquivo `/etc/inittab`.

Neste caso, o nível especificado nesta entrada é assumido como o nível inicial.

- 2 - Existe o arquivo `/etc/inittab`, mas não existe entrada do tipo `initdefault`.

Neste caso, **init** pede, através do terminal, ao sociado ao arquivo `/dev/console`, que o usuário forneça o nível inicial. Se for fornecido o nível `S`, **init** se comportará como descrito em 3.

- 3 - Não existe o arquivo `/etc/inittab`, ocorreu erro de abertura/leitura no arquivo `/etc/inittab` ou ainda, o arquivo existe mas está vazio.

Neste caso, é assumido o nível de operação `monousuário(S)`, o terminal-`console(/dev/console)` é aberto para entrada e saída, e o comando `/bin/su` é executado imediatamente. Para sair deste nível de operação, pode-se terminar o interpretador de comandos, quando **init** vai pedir um novo nível de operação, ou então fornecer um novo nível de operação através do comando **init**.

Com **init** ativado pela carga do sistema ou ao se passar do nível `monousuário(S)` para outro nível qualquer, **init** vai procurar na `inittab` e executar as entradas do tipo `boot` e `bootwait`, que tenham o nível de operação correspondente. Depois disto, a `inittab` é percorrida novamente, sendo então ativadas as demais entradas correspondentes ao nível, de acordo com o tipo.

Se algum processo-filho do **init** terminar, e se este processo for um filho imediato (e não um processo her^udado), o **init** irá obter da **inittab** a entrada cor^uresponde^unte, e se esta for do tipo **respawn**, o processo será reativado.

init provê uma facilidade que é o controle de disparo de entradas inconsistentes em **/etc/inittab** (erro de sintaxe, inexistência de comando, etc.). Se uma entrada requisitar muitos disparos em um pequeno intervalo de tempo, **init** a inibe, informando através de uma mens^usagem de erro qual entrada não será ativada, para que o usuário possa corrigi-la.

O recebimento de um sinal pelo **init** pode significar que ele deve mudar de nível ou então que ele deve per^ucorrer a **inittab** novamente, mantendo o nível anterior (opção **Q** do comando **init**). De qualquer modo, o **init** irá percorrer a **inittab** ativando as entradas devidas (que tenham nível igual ao nível atual, e de acordo com a ação especificada no tipo), e cancelando os pro^ucessos correspondentes às entradas indevidas (que es^utejam ativas, mas o nível não corresponde ao nível atual).

O cancelamento é feito de uma forma em que o processo tenha tempo de se preparar para terminar. Primeiro o **init** envia o sinal **SIGTERM**, para então, após terem sido decorridos alguns segundos, enviar o sinal **SIGKILL**.

3.3.1.4 O Comando INIT

Através do comando **init** pode-se alterar dinamicamente o nível de operação em curso. O comando **init** é usado da forma a seguir:

`init [0123456sSqQ]`

Deve-se fornecer apenas um parâmetro, tendo o seguinte significado:

0-6 - Faz com que o comando **init** envie um sinal ao processo **init**, informando-o que deve alterar o nível de operação para o valor fornecido.

s,S - Igual ao anterior, mas o novo nível será o monousuário.

q,Q - Informa ao processo **init** que ele deve percorrer novamente a `inittab`, sem alterar o nível de operação. Isto pode ser usado para que uma alteração feita na `inittab` (via editor) entre em vigor imediatamente.

3.3.1.5 O Arquivo /etc/rc

A existência deste arquivo não é obrigatória para o funcionamento de **init**, mas seu uso é bastante frequente. Seu conteúdo é definido pelo Administrador do Sistema, e é um programa para o interpretador de comandos-padrão (Shell). É usualmente disparado após transições de nível, e executa funções básicas tais como verificar e montar sistemas de arquivos, fazer cópias de segurança, etc. Convém notar que este arquivo só será executado se existir entrada em `/etc/inittab` que o utilize como comando, ou seja, **init** não é programado para executá-lo automaticamente. As entradas de `/etc/inittab` associadas a este arquivo são usualmente do tipo `wait`.

3.3.2 GETTY

3.3.2.1 Introdução

Este programa vem após o **init**, na seqüência usual de ativação do terminal. **getty** abre o arquivo especial associado ao terminal, criando, deste modo, os arquivos padrão `stdin`, `stdout` e `stderr`, configura as características de suporte ao terminal (velocidade de transmissão/recepção, caracteres associados a sinais e funções de edição, etc.), escreve na tela uma mensagem de identificação do sistema, uma tela de apresentação do sistema, uma mensagem de abertura de sessão, lê o nome da sessão e executa o programa **login**. Os campos da mensagem de identificação do sistema são obtidos através da chamada ao sistema `uname`. A tela de identificação do sistema é o conteúdo do arquivo `/etc/issue`, se existir. As características de suporte do terminal e a mensagem de abertura de sessão são obtidas dentro do arquivo `/etc/gettydefs`.

3.3.2.2 O Comando GETTY

Normalmente o arquivo `/etc/inittab` contém, para cada linha assíncrona ou terminal ligado ao sistema, uma entrada contendo um comando **getty**, conforme a seguinte sintaxe:

```
getty [-h] [-l] [-t tempo] [-e limerro] term [identr]  
[tipo] [linedisc]
```

Onde:

-h

Este parâmetro informa ao **getty** que o sinal DTR (Data Terminal Ready) da interface assíncrona não deve ser desativado no momento em que a linha é reconfigurada.

-l

Este parâmetro informa ao **getty** que o programa **login** não deve ser executado. Neste caso, **getty** termina após a reconfiguração da linha.

-t tempo

Este parâmetro faz com que **getty** termine tempo segundos após a escrita da mensagem de abertura de sessão, se nada for digitado no teclado do terminal.

-e limerro

Quando ativado com este parâmetro, **getty** termina sem executar **login**, se um nome válido de sessão não tiver sido digitado após limerro tentativas.

term

Especifica o nome do arquivo especial, dentro do diretório /dev, associado ao terminal.

identr

Identifica a entrada do arquivo /etc/gettydefs, onde **getty** deve buscar as características de suporte ao terminal e a mensagem de abertura de sessão.

tipo

Este parâmetro é uma cadeia, que será passada como parâmetro ao programa **login**, e copiada por este na variável TERM do ambiente de execução da Shell. Se tipo="NONE" ou "none", **getty** assume uma cadeia nula. Se este parâmetro não está presente, **getty** lê a cadeia do teclado do terminal após a leitura do nome da sessão.

linedisc

Este parâmetro é uma cadeia que informa a disciplina de suporte ao terminal. Na presente versão do sistema está disponível apenas a disciplina de fault igual a zero, que é a única padronizada.

O programa **getty** possui um mecanismo para ajuste da velocidade da interface assíncrona da UCP à velocidade do terminal. Uma mensagem ininteligível na tela é a indicação típica de que o terminal e a UCP estão trabalhando com velocidades diferentes. Se o operador digitar o caractere nulo, **getty** irá procurar uma configuração alternativa dentro do arquivo /etc/gettydefs, e escrever novamente a mensagem de identificação do sistema, o conteúdo de /etc/issue e a mensagem de abertura de sessão. Quando estas mensagens aparecerem corretamente na tela, as velocidades estarão compatibilizadas.

Ao digitar o nome da sessão, se o usuário utilizar apenas letras maiúsculas, **getty** irá supor que o terminal em questão não possui minúsculas e ajustará o suporte ao terminal para conversão sistemática de maiúsculas para minúsculas na entrada, e de minúsculas para maiúsculas na saída.

Quando o parâmetro `identr` é omitido, ou quando não existe, em `/etc/gettydefs`, uma entrada cujo identificador é `identr`, ou quando não existe o arquivo `/etc/gettydefs`, **getty** assume elementos de configuração adequados a terminais COBRA.

O comando **getty** pode ser usado para testar um arquivo de configurações. Neste caso, a sintaxe é a seguinte:

```
getty -c [arqconf]
```

-c

Opção de teste. Lista o arquivo cujo nome é `arqconf`, verificando a sintaxe e assinalando erros eventuais. Se o parâmetro `arqconf` não está presente, o teste é aplicado sobre o arquivo `/etc/gettydefs`.

3.3.2.3 O Arquivo /etc/gettydefs

Este arquivo constitui um catálogo de configurações, cobrindo a variedade de terminais existente na instalação (Ver estrutura TERMIO no Anexo B).

/etc/gettydefs é composto de entradas independentes separadas pelo caractere ~~newline~~. Entradas iniciadas pelo caractere # e entradas vazias são ignoradas. Entradas válidas têm o seguinte formato:

```
ident#configuração inicial#configuração final  
#msg#proxident
```

Onde:

ident

Identificador da entrada. Serve para pesquisa de uma entrada específica dentro do arquivo por cada um dos diversos programas **getty** gerados pelo programa **init**.

configuração inicial

A configuração inicial é um conjunto de cadeias separadas por brancos. Cada cadeia pode representar um **flag** da estrutura IOCTL para terminais (ISIG, ICANON, etc.) que deve ser ligado, ou o valor de um dos caracteres configuráveis. Neste último caso, a cadeia tem a forma L=xx, onde L é uma letra do conjunto {I, Q, E, K, F, L, M, T}, representando, respectivamente, os caracteres associados aos sinais interrupt (I) ou quit (Q), ou os caracteres associados às funções ERASE (E), KILL (K), END-OF-FILE (F) ou END-OF-LINE (L), ou os valores para os parâmetros MIN e TIME. xx representa um código hexadecimal para os caracteres, ou um valor inteiro para MIN ou TIME. A configuração inicial é uma configuração mínima, elementar, para escrita das mensagens iniciais na tela e leitura do nome da sessão.

configuração final

Este campo possui o mesmo formato que a configuração inicial. A configuração final é a configuração de regime, sob a qual os programas **login** e **shell** vão ser executados. Ela é setada imediatamente antes da execução do programa **login**.

msg

Mensagem de abertura de sessão, que aparece na tela, para convidar o usuário a escrever o nome da sessão. Dentro da cadeia que compõe a mensagem podem figurar todos os símbolos que são aceitos dentro de uma cadeia, pelo compilador C, para representar caracteres de controle.

Exemplo:

\t, \n, etc...

proxident

Identificador da entrada alternativa, que deve ser experimentada por **getty** quando o usuário digitar um caractere nulo para ajuste de velocidades. Normalmente as entradas /etc/gettydefs são encadeadas circularmente, cobrindo todas as velocidades suportadas pelo programa de controle de terminais.

A seguir é apresentado um exemplo de arquivo /etc/gettydefs:

```
TI200-9.6 # B9600 OPOST ONLCR ECHO ICRNL CREAD TAB3
CS8 CLOCAL ISIG ICANON I=7F # B9600 HUPCL ECHOE ECHOK
ECHO CLOCAL CREAD TAB3 CS8 OPOST ONLCR ISIG ICANON
ICR NL IXON IGNBRK I=1B Q=0C E=08 F=04 #\a\nAbra ses
sao : # TI200-19.2
```

```
TI200-19.2 # B19200 OPOST ONLCR ECHO ICRNL CREAD TAB3
CS8 CLOCAL ISIG ICANON I=7 F # B19200 HUPCL ECHOE
ECHOK ECHO CLOCAL CREAD TAB3 CS8 OPOST ONLCR ISIG ICA
NON ICRNL IXON IGNBRK I=1B Q=0C E=08 F=04 #\a\nAbra
sessao : # TI200-19.2
```

```
RENPAC-0.3 # B300 CSTOPB OPOST ONLCR ISIG ICRNL CREAD
TAB3 ISIG ICANON BRKINT CS 8 I=7F # B300 CSTOPB OPOST
ONLCR TAB3 HUPCL CREAD CS8 ISIG ICANON ICRNL IXON BRK
INT I=7F Q=1C E=08 K=40 F=04 #\a\nAbra sessao : # REN
PAC-0.3
```

```
ANSI-9.6 # B9600 OPOST ONLCR ECHO ICRNL CREAD TAB3
CS8 CLOCAL ISIG ICANON I=7F # B9600 HUPCL ECHOE ECHOK
ECHO CLOCAL CREAD TAB3 CS8 OPOST ONLCR ISIG ICANON
ICRN L IXON IGNBRK I=7F Q=0C E=08 K=40 F=04 #\a\nAbra
sessao : # ANSI-19.2
```

```
ANSI-19.2 # B19200 OPOST ONLCR ECHO ICRNL CREAD TAB3
CS8 CLOCAL ISIG ICANON I=7F # B19200 HUPCL ECHOE
ECHOK ECHO CLOCAL CREAD TAB3 CS8 OPOST ONLCR ISIG ICA
NON I CRNL IXON IGNBRK I=7F Q=0C E=08 K=40 F=04
#\a\nAbra sessao : # ANSI-19.2
```

3.3.2.4 O Arquivo /etc/issue

Este arquivo não é obrigatório para o funcionamento de **getty**. Se existir, seu conteúdo será copiado para o terminal que está sendo configurado. É freqüentemente usado para a exibição de tela de apresentação do sistema.

3.3.3 **LOGIN**

Este programa é o último na seqüência usual de ativação de um terminal. Ao ser executado, ele verifica a validade do nome da sessão e da senha, consultando o arquivo /etc/passwd, atualiza os arquivos /etc/utmp e /etc/wtmp e executa o programa especificado no último campo da entrada do arquivo /etc/passwd, correspondente ao usuário que está abrindo a sessão. Este programa é, em geral, um interpretador de comandos.

Qualquer cadeia de caracteres adicional, fornecida em resposta ao pedido de nome de sessão, será inserida no ambiente do interpretador de comandos executado pelo **login**.

As cadeias adicionais fornecidas podem ter a forma aaa ou bbb=ccc. Cadeias sem o sinal de igual serão colocadas no ambiente na forma de:

Ln=aaa

onde n é um número que começa de 0 e é incrementado a cada argumento que não tenha o sinal de igual. Cadeias com o sinal de igual serão colocadas inalteradas no ambiente. Se alguma cadeia fornecida já se encontrar no ambiente, ela será modificada para o valor fornecido. Existem apenas três exceções: as variáveis **COBRA**, **PATH** e **SHELL** que não podem ser alteradas.

3.4 DESATIVAÇÃO DO SISTEMA

Para que o SOX seja desativado de uma forma controlada, deve ser executado o comando sfo (sistema fora de operação). Recomenda-se, entretanto, que, antes da execução deste comando, o administrador tome certas medidas, como: avisar aos usuários que o sistema vai sair de operação algum tempo antes de executar o comando, colocar mensagens para os usuários que estejam executando processo em lote avisando que o sistema foi desativado (estes usuários podem não estar presentes e por consequência não vão saber o que houve com seus processos).

Geralmente, para desativar o sistema, o administrador deve se utilizar do arquivo /etc/shutdown, que é um arquivo de comandos para o interpretador. Este arquivo contém os comandos que implementam a desativação ordenada do sistema, de acordo com as necessidades do administrador.

CAPÍTULO 4

TAREFAS ADMINISTRATIVAS

4.1 CADASTRAMENTO DE NOVOS USUÁRIOS

4.1.1 INTRODUÇÃO

Para que um usuário possa abrir uma sessão no SOX, e utilizar seus recursos, é necessário que:

- . Exista uma entrada no arquivo /etc/passwd associada ao usuário;
- . O usuário tenha um diretório inicial de trabalho.

O Administrador do Sistema deve observar os seguintes passos para cadastrar um novo usuário:

1. Criar uma entrada no arquivo /etc/passwd, fornecendo nome de sessão, diretório de trabalho, identificadores de usuário e de grupo, através de um editor de textos.
2. Criar, caso não exista, o arquivo-diretório atribuído ao usuário em 1, através do comando **mkdir**.
3. Estabelecer as permissões de acesso desejadas no diretório criado em 2, através do comando **chmod**.

4. Alterar os identificadores de dono e grupo do diretório criado em 2, para os identificadores de dono e grupo estabelecidos em 1, através dos comandos **chown** e **chgrp**.
5. Acrescentar o nome do usuário à lista de usuários correspondente à entrada do arquivo `/etc/group` relativa à identificação de grupo estabelecida em 1.

A atribuição de senha de acesso ao usuário recém-cadastrado é tarefa do próprio usuário. Ele deve abrir sua sessão e executar o comando **passwd**, fornecendo então a sua senha.

4.1.2 O ARQUIVO `/etc/passwd`

O arquivo `/etc/passwd` é um arquivo ASCII, composto de entradas separadas pelo caractere **newline**. Cada entrada é composta de campos separados por `:` (dois pontos), tendo-o seguinte formato:

```
nome:senha:idu:idg:comentário:diretório:intcom
```

Onde:

nome - Nome de sessão do usuário;

senha - Senha criptografada. Se este campo for nulo, significa que o usuário não tem senha;

idu - Identificador de usuário;

idg - Identificador de grupo;

comentário - Comentário qualquer. Não é considerado pelo sistema. Pode ser nulo;

diretório - Diretório corrente inicial;

intcom - Interpretador de comandos que será usado pela sessão do usuário. Caso se já nulo, será usado o interpretador-padrão do SOX (/bin/sh).

A seguir são apresentados alguns critérios para o preenchimento de uma linha do arquivo /etc/passwd.

Coloca-se o nome da sessão atribuído ao novo usuário; o campo senha deve ser deixado nulo. Os valores do identificador de usuário e de grupo devem ser escolhidos, de acordo com os critérios da instalação. Normalmente agrupa-se os usuários que tenham algum relacionamento, por exemplo, mesmo projeto, mesmo departamento, etc. Deve ser lembrado que o esquema de proteção de acesso a arquivos no SOX é baseado nesses identificadores e, também, que um usuário que tenha identificador 0 será considerado Superusuário, tendo privilégios especiais, o que deve sempre ser evitado. Por convenção, os valores 0, 2 e 3 são os identificadores dos usuários root, bin e sys, respectivamente, e o valor 1 é reservado.

A escolha do diretório inicial de trabalho deve ser feita levando em consideração a distribuição de uso dos volumes, caso a instalação permita a existência de vários volumes montados ao mesmo tempo, e também o relacionamento entre os usuários, podendo ser atribuído o mesmo diretório para um determinado conjunto de usuários.

O interpretador de comandos a ser utilizado pelo usuário normalmente é o arquivo /bin/sh, que é o interpretador-padrão do SOX. Mas pode-se fornecer qualquer programa, o que pode significar tornar o usuário restrito a um sistema fechado, implementado pelo programa fornecido.

A seguir é apresentado um exemplo de arquivo /etc/passwd.

```
root:cWHVZT:0:0:sessao raiz geral:/:/bin/sh
instal::0:0:sessao de instalacao:/bin:/bin/shin
eduardo::100:100:./usr/eduardo:/bin/sh
paulo::101:100:./usr/paulo:
frank::200:150:./usr/frank:/usr/bin/sistema1
```

4.2 COMUNICAÇÃO COM USUÁRIOS

Existem no SOX alguns comandos e procedimentos para que o administrador se comunique com os usuários. A escolha de qual deles será usado vai depender da situação.

Normalmente, para que os usuários recebam uma mensagem sempre que abrirem uma sessão, deve ser usado o arquivo /etc/motd e o comando **news**. O arquivo /etc/motd deve conter uma mensagem que será sempre impressa no terminal do usuário ao abrir sessão. O comando **news**, que se utiliza do arquivo /etc/news, é usado para que os usuários recebam mensagens apenas uma vez. Mensagens já recebidas não serão enviadas novamente.

Para que essa forma de comunicação seja implementada, é necessário que seja incluído no arquivo `/etc/profile` um comando que gere a impressão do arquivo `/etc/motd` e a execução do comando `news`.

Recomenda-se que se atualize o arquivo `/etc/motd` todo dia e o arquivo `/etc/news` toda semana.

Para que se envie uma mensagem a todos os usuários que estejam com suas sessões abertas, deve-se utilizar do comando `wall`.

Existe ainda o comando `write` que envia uma mensagem para um determinado usuário.

A descrição do uso dos comandos aqui mencionados pode ser encontrada no "Manual de Referência Comandos SOX", enquanto que a descrição dos arquivos pode ser encontrada no Anexo B deste manual.

4.3 VERIFICAÇÃO E RECUPERAÇÃO DE SISTEMAS DE ARQUIVOS

Conforme mencionado no Capítulo 2 deste manual, podem ocorrer vários tipos de erros no Sistema de Arquivos. Com o objetivo de constatar e reparar esses erros, de vem ser adotados procedimentos de verificação e recuperação de Sistemas de Arquivos.

O comando `fsck` é o responsável pela verificação e recuperação dos sistemas de arquivos do SOX. Sua descrição completa deve ser vista no "Manual de Referência Comandos SOX". Aqui será dado apenas um resumo de sua operação.

O **fsck** é um comando interativo, necessitando da intervenção do operador. À medida que ele for encontrando inconsistências no Sistema de Arquivos, será apresentado o tipo de inconsistência e perguntado ao operador se ele concorda com a solução para sua recuperação. O comando opera em seis fases:

Fase 1: Verifica caminhos dos arquivos

Percorre a árvore hierárquica, verificando se os BCAs referenciados pelas diversas entradas dos diretórios são válidos, e marcando-os como ocupados.

Fase 2: Verifica lista de BCAs

Percorre a lista de BCAs do volume à procura daqueles que estejam marcados ocupados, mas não foram referenciados em diretório algum (BCAs perdidos). Para cada BCA perdido encontrado, será perguntado ao operador se ele deseja reconectá-lo. Para que essa operação seja possível, é necessário que exista um diretório de nome /lost+found, com entradas disponíveis.

Fase 3: Verifica conectividade

Para cada BCA correspondente a um arquivo tipo diretório encontrado é verificado se a primeira entrada é . (ponto) e se sua segunda entrada é .. (ponto-ponto). Além disto, é verificado se o BCA correspondente à entrada .. (ponto-ponto) (BCA do pai) possui uma entrada que se referencia ao BCA corrente. Qualquer inconsistência nesta fase é comunicada.

Fase 4: Verifica contadores de referência

Verifica para cada BCA se a soma das referências feitas a ele é igual ao seu contador de associações. Verifica, também, se o número de BCAs livres, contido na seção de informação, está correto. Qualquer inconsistência encontrada nessa fase é comunicada e é perguntado se o operador deseja corrigi-la.

Fase 5: Verifica mapa de blocos

Nesta fase são verificados os blocos ocupados por cada arquivo do volume. É feita a consistência se existir algum bloco de um arquivo que esteja marcado como livre no mapa de blocos, se algum bloco pertence a mais de um arquivo ou se o bloco é inválido (fora do volume). Caso ocorra alguma inconsistência, o comando irá perguntar ao operador se ele deseja cancelar a referência ao bloco. Se a resposta for não, **fsck** termina.

Fase 6: Grava novo mapa de blocos

É feita a contagem dos blocos perdidos e informado ao operador o seu número, caso exista algum. É perguntado ao operador se ele deseja gravar o novo mapa de blocos obtido.

4.4 SALVAMENTO E RECUPERAÇÃO DE SISTEMAS DE ARQUIVOS

A forma mais segura e garantida de evitar a perda de dados, em casos de erros ocorridos no sistema de arquivos, é a adoção de uma política de salvamento regular e freqüente dos sistemas de arquivos da instalação. Qual política a ser adotada deve ser decidida pelo administrador, pois cada instalação tem suas próprias necessidades. Os fatores que devem ser levados em consideração para a elaboração dessa política, são, entre outros, a disponibilidade de tempo para ser dispendido com o salvamento, a freqüência com que ocorrem erros no sistema, a natureza dos dados armazenados no sistema de arquivos, etc.

O meio no qual devem ser feitos os salvamentos também vai depender da instalação. A configuração da máquina pode conter unidades de fita magnética ou unidades de fita-cartucho ou, ainda, apenas unidades de disco flexível.

Para maiores informações sobre este assunto, devem ser consultados no "Manual de Referência Comandos SOX", os seguintes comandos:

- . **find** em conjunto com **cpio**;
- . **tar**;
- . **bkpcp**.

Em máquinas da linha X-500, o utilitário COPVOL do Sistema SOD (versões posteriores à W.00 do utilitário) também é capaz de salvar os sistemas de arquivos do SOX.

4.5 COMUNICAÇÃO DO SISTEMA COM O OPERADOR

Em certas situações, o SOX pode precisar de uma resposta do operador para alguma intervenção ou então precisa informar algum erro ocorrido. Para isto, o SOX se utiliza de mensagens de intervenção e de erro.

4.5.1 INTERVENÇÕES

Sempre que o SOX se deparar com algum evento anormal, em que for necessária a intervenção do operador, será impressa no terminal-console uma mensagem de intervenção, com o seguinte formato:

ip-nn: mensagem

onde ip significa "intervenção pendente", nn é um número que identifica a intervenção e "mensagem" é a descrição do evento ocorrido. A mensagem impressa vai depender do evento, por exemplo, "Impressora não pronta.", etc.

Ao receber a mensagem de intervenção, o operador deve realizar a operação requerida, por exemplo, colocar papel na impressora, ligar um dispositivo, colocar um disco flexível na unidade e responder à intervenção através do comando ip. Este comando tem o seguinte funcionamento:

ip [nn] [ação]

Os parâmetros são posicionais, e têm o seguinte significado:

nn - Número de identificação da intervenção que se deseja responder.

ação - Pode ter os valores:

a : Aborta a operação que gerou a intervenção;

r : Repete a operação;

i : Ignora o evento que gerou a intervenção.

Se não for fornecido o parâmetro ação, será assumido por default o valor r.

Caso seja executado o comando sem parâmetros, será gerada uma lista das intervenções não respondidas.

Este comando se comunica com o sistema através do arquivo /dev/erro. Recomenda-se que este arquivo seja colocado no sistema com permissões tais que, apenas os Superusuários possam acessá-lo.

4.5.2 ERROS

Caso o sistema detecte algum erro, será impressa no terminal console uma mensagem de erro com o seguinte formato:

erro: mensagem

onde mensagem relata o tipo de erro ocorrido.

Mensagens de erro servem apenas para informar ao operador que houve o erro, não requerendo resposta.

4.5.2.1 Erros no Sistema de Arquivo

Caso o sistema detecte algum erro no sistema de arquivos, será impressa uma mensagem de erro com o seguinte formato:

erro: dddd - Volume defeituoso (txt).

Onde:

dddd - Indica o dispositivo onde está montado o volume. Corresponde ao último nome do caminho do arquivo especial associado. Por exemplo: df0, se o volume estiver montado em /dev/df0.

txt - Indica a área de controle onde ocorreu o erro. Pode ser "BCAs", indicando a área de BCAs; "secinf", indicando a seção de informação; ou "mb", indicando o mapa de blocos.

4.5.3 O PROCESSO-CONSOLE

Este processo é o responsável pela impressão das mensagens de intervenção e erro no terminal-console do sistema. Corresponde ao arquivo /etc/console. Sua execução pelo SOX não é automática. Recomenda-se que se ja inserida no arquivo /etc/inittab uma entrada que gere a criação e execução desse processo (ver Capítulo 3 desse manual).

CONFIGURADOR DO SOX-CONF

5.1 INTRODUÇÃO

O Configurador do SOX (conf) é o programa que carrega os módulos do SOX na memória, de acordo com informações obtidas no arquivo de configuração denominado conf.sox.

O SOX é constituído seguindo uma arquitetura que o divide em Núcleo, Servidor e Processos de Máquina Real (PMRs). O Núcleo cria os ambientes onde o Servidor e os PMRs executam. O Servidor é o módulo que implementa as funções do sistema. Os PMRs são os módulos correspondentes aos programas de controle dos dispositivos periféricos.

Este capítulo contém uma descrição completa do Configurador, incluindo a configuração dos PMRs, de tabelas internas ao Núcleo e Servidor. O configurador é capaz de atribuir valores default para a maioria dos parâmetros em questão. Os únicos parâmetros de configuração que são obrigatórios são os relativos aos PMRs e o parâmetro relativo ao Servidor, que indica o dispositivo-raiz do SOX.

Normalmente o administrador precisa se preocupar apenas com os itens 5.2.3.1 (parâmetro S_draiz) e 5.2.4.1 (relativos aos PMRs). Os demais parâmetros só devem ser alterados em situações especiais.

5.2 O ARQUIVO conf.sox

5.2.1 DESCRIÇÃO

Este arquivo contém informações de configuração do Núcleo, do Servidor e dos PMRs que deverão ser carregados durante a carga do sistema. É um arquivo ASCII, contendo comandos separados por ; (ponto-e-vírgula), podendo existir comentários entre estes. Um comentário é, como na Linguagem C, iniciado por /* e termina do por */. **conf** ignora todo o texto entre o início e o fim do comentário, de modo que comentários e comandos podem ser livremente misturados. Brancos, tabulações (\t) e fim-de-linha (\n) encontrados no texto são ignorados.

Um comando atribui um valor a um parâmetro de configuração de um módulo, da seguinte forma:

```
<modulo>_<parm>=<valor>;
```

Onde:

<modulo>

Um caractere designando o módulo ao qual se aplica o parâmetro de configuração, sendo suportados N (Núcleo), S (Servidor SOX) e R (Processo da Máquina Real).

<parm>

Nome do parâmetro. Para cada módulo aplica-se um certo conjunto de parâmetros.

<valor>

Valor do parâmetro, podendo ser um nome de arquivo, um número ou uma cadeia de caracteres. Os parâmetros numéricos serão interpretados da seguinte forma:

base hexadecimal - Cadeia de dígitos, iniciada com 0x ou 0X podendo conter os caracteres de a a f (maiúsculos ou minúsculos) (ex. 0x1a).

base decimal - Cadeia de dígitos (de 0 a 9) (ex. 14).

Os comandos não necessitam estar grupados por módulo, podendo livremente ser misturados. A Figura 5-1 fornece um exemplo de um arquivo de configuração.

```

/*
 * Parametros de configuracao do Nucleo do SOX
 */

N_nucleo = nucleo ; /* nome do arq. modulo */
                    /* de carga do nucleo*/
N_nds = 10 ;        /* no. maximo de descritores */
                    /* de sinal */
N_ndpc = 24 ;      /* no. de portas de comunic. *
N_ndpi = 16 ;      /* no. de portas de interrup */
N_ndev = 40 ;      /* no. de portadores de mens.*
N_ndati = 16 ;     /* no. de ATI's */

```

```

/*
 * Parametros de configuracao do Servidor SOX
 */

```

```

S_serv = servidor ; /* nome do arq. mod. de */
                      /* carga do Servidor */
S_nodo = MAQUINA ; /* nome da maquina na rede */
S_draiz = 0x0000 ; /* disposit. raiz do sist. */
S_npmv = 50 ;      /* no. de processos da */
                      /* da maquina virtual */
S_ntvol = 5 ;      /* no. volumes montados */
S_ndalc = 30 ;     /* no. desc. de arq aberto */
S_naca = 20 ;      /* no. areas de contr de arq */
S_ndsmc = 8 ;      /* no. desc. de memoria */
                      /* compartilhada */
S_ndsem = 4 ;      /* no. desc. de semaforo */
S_nsem = 8 ;       /* no. semaforos */
S_ndrsem = 16 ;    /* no. desc. de recuperacao */
                      /* de semaforos */
S_nlocf = 20 ;     /* no. "locks" ativos */
S_nbuf1k = 20 ;    /* no. buffers de 1 Kbytes */
S_nbuf2k = 0 ;     /* no. buffers de 2 Kbytes */
S_nbuf4k = 0 ;     /* no. buffers de 4 Kbytes */
S_ndblc = 8 ;      /* no. disposit. tipo bloco */

/*
 * Param. de Config. do Ambiente de Maquina Real
 */

R_i=0x14:pmrdt:daddt; /* tty's plc base */
R_e=0x16:pmrc1a::
    carcla:nuccla::; /* cla */
R_i=0x00:pmrdf;;     /* df's e rdf's */
R_e=0x13:pmrmem;;    /* mem e kmem */
R_e=0x15:pmrip;;     /* lp */
R_e=0x19:pmrerr;;    /* erro */

```

Figura 5-1
 EXEMPLO DE UM ARQUIVO DE CONFIGURAÇÃO

5.2.2 PARÂMETROS DE CONFIGURAÇÃO DO NÚCLEO

Todos os parâmetros de configuração do Núcleo são opcionais. Para cada parâmetro não-declarado, **conf** assume um valor-padrão, dependente do número de PMRs configurados, conforme especificado no item 5.4.1.

Se este valor não for satisfatório, o usuário deve alterar o arquivo **conf.sox** e declarar explicitamente o parâmetro desejado, atribuindo a ele o valor de sua conveniência.

5.2.2.1 Descrição dos Parâmetros

- N_nucleo** - Cadeia de caracteres que especifica o nome do arquivo contendo o executável formato FAOS do Núcleo do SOX.
- N_nds** - Número máximo de sinais de software que podem ser enviados simultaneamente.
- N_ndpc** - Número máximo de Portas de Comunicação.
- N_ndpi** - Número máximo de Portas de Interrupção.
- N_ndev** - Número máximo de Portadores de Mensagem.
- N_ndati** - Número máximo de Atendedores de Interrupção.

5.2.3 PARÂMETROS DE CONFIGURAÇÃO DO SERVIDOR

Dentre os parâmetros de configuração do Servidor, apenas o que define o dispositivo raiz do sistema (S_draiz) é obrigatório. Se ele não estiver declarado, a inicialização do sistema é interrompida. Os demais parâmetros são opcionais e, para cada um deles que não estiver declarado, **conf** assume um valor-padrão, conforme descrito no item 5.4.2.

5.2.3.1 Descrição dos Parâmetros

- S_serv - Cadeia de caracteres, que especifica o nome do arquivo contendo o executável formato FAOS do Servidor SOX.
- S_nodo - Cadeia de caracteres alfanuméricos, que define o nome da máquina na rede.
- S_draiz - Este parâmetro define para o Servidor SOX qual o dispositivo a ser referenciado como raiz do sistema de arquivos. É um valor do tipo int, em que os 8 bits mais significativos representam o tipo e os 8 bits menos significativos indicam o subtipo do dispositivo. Este parâmetro é obrigatório.
- S_npmv - Número máximo de Processos da Máquina Virtual (PMVs).
Dimensiona a Tabela de Processos do sistema.
- S_ntvol - Número máximo de volumes simultaneamente montados.
Dimensiona a Tabela de Volumes do sistema.

- S_ndalc** - Número máximo de arquivos abertos.
Dimensiona a Tabela de Descritores de Ar
quivos do sistema.
- S_ndsmc** - Número máximo de segmentos de memória com
partilhada.
Dimensiona a Lista de Descritores de Seg
mentos de MC do sistema.
- S_ndsem** - Número máximo de descritores de semáforo.
Dimensiona a Lista de Descritores de Semá
foros do sistema.
- S_nsem** - Número máximo de semáforos.
Dimensiona a Lista de Semáforos do siste
ma.
- S_ndrsem** - Número máximo de descritores de recupera
ção de semáforo.
Dimensiona a Lista de Descritores de Recu
peração de Semáforos do sistema.
- S_naca** - Número máximo de Áreas de Controle de Ar
quivo (ACAs).
Dimensiona a Lista de ACAs do sistema.
- S_nlocf** - Número máximo de reservas simultâneas em
arquivos.
Dimensiona a Lista de Nós de Reserva do
sistema.
- S_nbuf1k** - Número de buffers de 1 kocteto no **buf**
fer pool central do sistema.
- S_nbuf2k** - Número de buffers de 2 koctetos no **buf**
fer pool central do sistema.

S_nbuf4k - Número de buffers de 4 koctetos no **buf**
fer pool central do sistema.

S_ndblc - Número de dispositivos tipo bloco.

5.2.4 PARÂMETROS DE CONFIGURAÇÃO DA MÁQUINA REAL

Este item descreve os PMRs a serem carregados pelo programa **conf**.

5.2.4.1 Descrição dos Parâmetros

Os parâmetros de configuração, cujo nome de módulo é R, referem-se à configuração de PMRs e, opcionalmente, à carga de software adicional em controladores de periféricos.

O programa **conf** preocupa-se apenas com os parâmetros R_e e R_i, que se referem à configuração estática da Máquina Real. O parâmetro R_i significa também que o PMR é imprescindível ao funcionamento do SOX e, se houver algum erro durante a sua carga, o procedimento de inicialização será abortado. Opcionalmente podem ser inseridos parâmetros "R_" referindo-se à configuração dinâmica de PMRs no sistema (parâmetros R_d).

O valor atribuído a um parâmetro R e ou R i é uma cadeia de caracteres terminada por ; (ponto-e-vírgula). O formato geral desse parâmetro é mostrado abaixo, onde colchetes [] cercam um campo opcional e chaves { } englobam um campo que pode ser repetido zero ou mais vezes, separados por , (vírgula) e () (parênteses) definem informações adicionais. A parte entre parênteses é opcional. Ela contém informações que precisam ser fornecidas quando o PMR em questão estiver associado a um controlador:

```
R_e|i={<tipo>}:<pmr>:[<dadpmr >]
(:<conf>:<cntr>:[<dadcntr>]: <repcnt r>)];
```

Onde:

<tipo>

Este valor designa a Porta de Comunicação a ser criada pelo PMR, para o recebimento de pedidos. Todos os serviços relativos a um arquivo especial caractere ou bloco, cujo "tipo" for <tipo> serão enviados, pelo Servidor, para essa Porta de Comunicação. Isto significa que o <tipo> aqui fornecido deve estar de acordo com o "tipo" dos arquivos especiais que serão tratados por este PMR (ver comando **knod** - "Manual de Referência Comandos SOX").

<pmr>

Este campo define o nome do arquivo contendo o executável formato FAOS de um PMR a ser criado e carregado na Máquina Real. Este campo é obrigatório e, se for omitido, a carga do sistema é abortada.

<dadpmr>

Este campo define o nome do arquivo contendo dados de configuração do PMR. Este campo é opcional e, se estiver declarado, **conf** criará o PMR com estes dados acoplados à sua área, conforme descrito em 5.3.4.3. A interpretação dos dados deste arquivo é de responsabilidade exclusiva do PMR.

Campos referentes à carga de software em controladores (são tem sentido para o configurador do X-10).

conf : Nome do arquivo contendo o executável FAOS do programa responsável pela configuração/instalação do software no controlador. Este programa é um programa autônomo que é carregado e ativado por **conf**.

cntr : Nome do arquivo contendo o executável FAOS do software a ser carregado no controlador. Este software, dependendo do controlador, pode ser um programa que controla o hardware, ou o Núcleo em controladores que mantém um Ambiente de Máquina Real (AMR).

dadcntr : Nome do arquivo contendo dados de configuração do software, a ser carregado no controlador.

repcntr : Nome do arquivo contendo o executável FAOS do representante, a ser carregado no controlador. Este campo só tem sentido para controladores que mantenham um AMR.

A seguir são dados alguns exemplos de como se especifica a configuração de Máquina Real, envolvendo várias situações.

Exemplo 1:

O PMR não tem necessidade de configuração, ou pelo menos pode facilmente se autoconfigurar após ativado. Processos deste tipo não possuem dados de configuração:

R_e=<tipo>:<pmr>;

Exemplo 2:

O PMR possui dados de configuração, que serão interpretados pelo <pmr> após receber o Sinal de Ativação:

R_e=<tipo>:<pmr>:<dadpmr>;

Exemplo 3:

O PMR está associado a um controlador, havendo portanto a necessidade de se carregar software neste controlador. Neste caso, devem ser fornecidos os nomes dos arquivos contendo o módulo de carga do configurador (<conf>), o módulo de carga do software a ser carregado no controlador (<cntr>) e, se for o caso, os dados de configuração (<dadcntr>):

R_e=<tipo>:<pmr>:<conf>:<cntr>:<dadcntr>;

Exemplo 4:

Este último caso se aplica a controladores que mantêm um AMR, com o Núcleo e processos. Para isto deve ser declarado o arquivo contendo o módulo de carga do Representante <repcntr> que será carregado no controlador por <conf>:

```
R_e=<tipo>:<pmr>::<conf>:<cntr>:<dadcntr>:<repcntr>;
```

A prioridade de execução entre os PMRs é definida pela ordem em que eles são declarados no arquivo de configuração. No exemplo da Figura 5-1, o "pmdt" é o mais prioritário e o "pmmem" o menos prioritário.

5.3 FUNCIONAMENTO DO PROGRAMA CONF

5.3.1 CARGA E ATIVAÇÃO

O programa **conf** é carregado pelo procedimento inicial de carga, denominado **boot**, em um endereço de memória específico, garantindo espaço suficiente para a carga do SOX, sem haver sobreposição. Após a sua carga, um mapa da memória física da máquina será semelhante ao mostrado na Figura 5-2.

Na rotina de inicialização são obtidas as seguintes informações: tamanho da memória física, tipo da máquina e identificação da máquina.

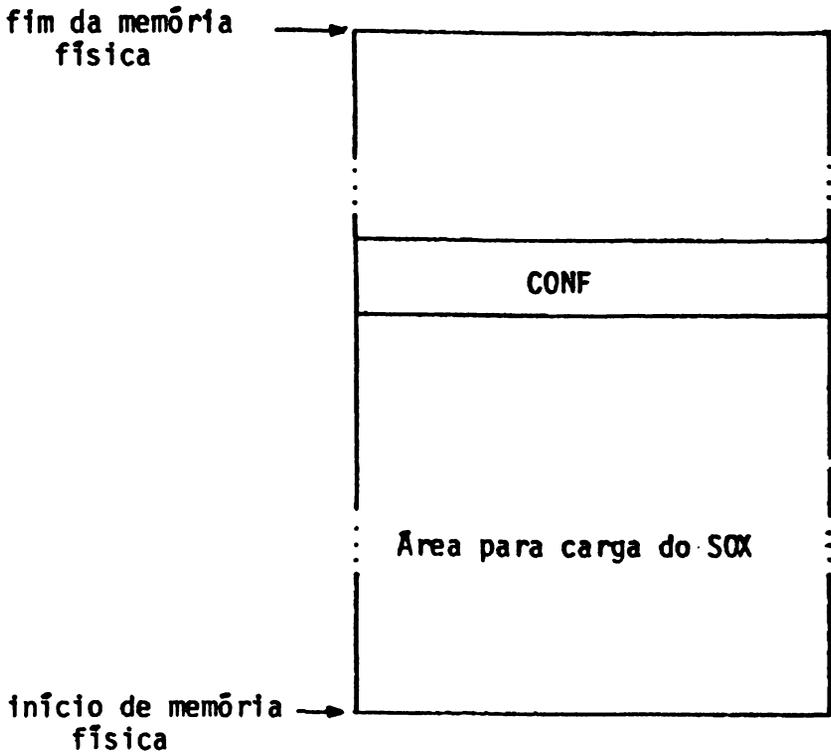


Figura 5-2
MAPA DE MEMÓRIA FÍSICA APÓS A CARGA DE CONF.

Após a rotina de inicialização, **conf** envia a mensagem: "Configurando sistema..."

5.3.2 LEITURA DO ARQUIVO **conf.sox**

Após terminadas suas tarefas de inicialização, **conf** lê o arquivo **conf.sox** para a memória. Caso haja um indicador na configuração de hardware, o conteúdo do arquivo **conf.sox** pode ser alterado em memória, permitindo assim configurações alternativas sem comprometer o arquivo de configuração.

5.3.3 ANÁLISE DE CONSISTÊNCIA DO ARQUIVO **conf.sox**

Após a leitura do arquivo para a memória, **conf** analisa seu conteúdo para obter os dados de configuração do Núcleo e do Servidor, e quais os PMRs que deverão ser carregados. Durante esta fase, algumas consistências são observadas e são aqui apresentadas:

- . Os comandos devem estar sintaticamente corretos. Se algum comando possuir erro de sintaxe, toda a linha será ignorada, ou seja, **conf** irá procurar por um caractere ; (ponto e vírgula), para proseguir a análise;
- . Nos parâmetros da Máquina Real, todos os campos não-opcionais devem estar declarados;
- . O parâmetro de configuração do Servidor **S draiz** tem que estar declarado conforme especificado, com tipo e subtipo;

- . Nos parâmetros da Máquina Real deve ter sido declarado um PMR com o valor de <tipo> idêntico ao tipo do dispositivo-raiz do sistema (parâmetro S_draiz). Se estes dados não estiverem combinados, significa dizer que não existirá um PMR para o dispositivo-raiz do sistema, o que torna impossível o funcionamento do SOX;
- . O maior tipo configurado nas declarações de parâmetros da Máquina Real não pode ultrapassar um valor máximo, estabelecido atualmente em 0xEF;
- . Todos os valores para parâmetros são testados em relação aos valores-padrão (ver item 5.4), que são considerados mínimos. Caso aqueles valores estejam abaixo do padrão, estes serão assumidos.

Por possuírem parâmetros análogos em /usr/include/limits.h, S_npmv, S_ndalc, S_naca e S_locf devem ter valores no mínimo iguais a aqueles. Caso sejam configurados valores menores, conf assume os de /usr/include/limits.h.

5.3.4 CARGA DOS MÓDULOS

Inicialmente é feita a carga do software nos controladores, se existirem, em seguida é carregado o Núcleo, os PMRs e, por último, o Servidor.

5.3.4.1 CARGA DE SOFTWARE EM CONTROLADORES

No início da carga é enviada a mensagem <cntr>..., que é o nome do arquivo contendo o módulo de carga do software a ser carregado no controlador. **conf** lê os arquivos <cntr> , <dadcntr> , <repcntr> para o início da memória física. Depois carrega o programa autônomo <conf> e o ativa passando o endereço na memória e o tamanho de cada um dos três arquivos acima.

Se **conf** receber erro na leitura/carga dos módulos acima ou se o programa <conf> retornar erro, o PMR associado a esse controlador (<pmr>) não será carregado. Se isso acontecer, e este PMR estiver declarado como imprescindível (R_i), **conf** abortará os procedimentos de inicialização do sistema.

O programa <conf> retorna o endereço do controlador, que posteriormente será passado ao PMR associado, dentro de sua Área de Parâmetros. Este processo será repetido para cada controlador configurado no sistema.

A Figura 5-3 mostra o mapa de memória física no momento que o programa <conf> é ativado.

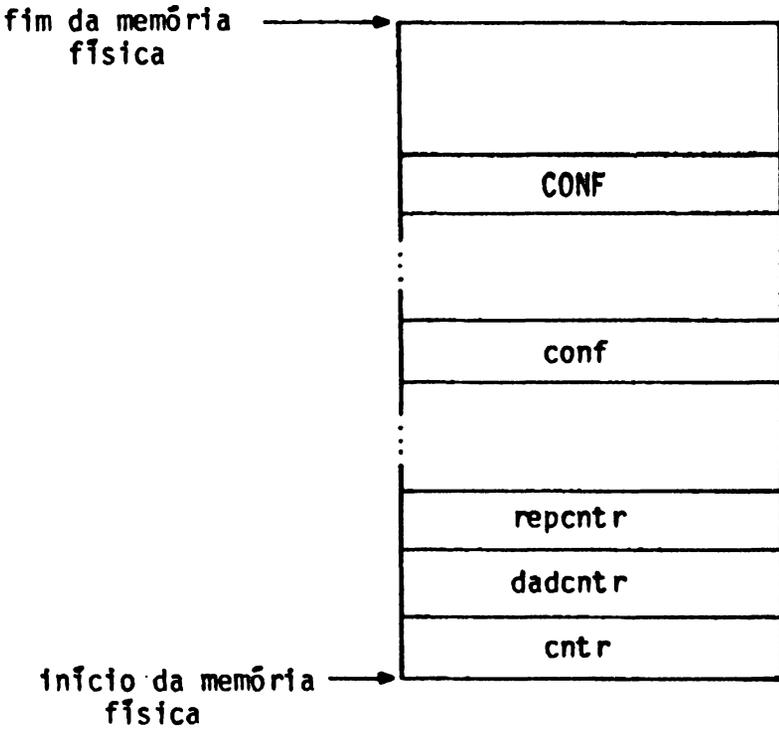


Figura 5-3
MAPA DE MEMÓRIA FÍSICA NA ATIVAÇÃO DOS CONTROLADORES

5.3.4.2 Carga do Núcleo

É enviada a mensagem <nucleo>..., ou seja, o nome do arquivo contendo o executável do Núcleo. Este arquivo é carregado no endereço zero de memória. Neste ponto, através de chamadas de configuração oferecidas pelo Núcleo, **conf** envia a este os dados relativos à sua configuração (parâmetros N). Isto faz com que o Núcleo dimensione a área de dados em função do ambiente que irá controlar.

5.3.4.3 Carga dos PMRs

É enviada a mensagem <pmr>..., que é o nome do arquivo contendo o módulo de carga do PMR, sendo carregado. Para cada PMR, **conf** lê o cabeçalho do módulo de carga do arquivo designado por <pmr> e obtém informações sobre o arquivo de dados <dadpmr>, se for o caso. Através de chamadas de configuração, oferecidas pelo Núcleo, **conf** solicita a este a criação de um PMR com tamanho $TPMR+TPARM+TDADOS$, onde:

- TPMR - Tamanho do programa;
- TPARM - Tamanho da Área de Parâmetros;
- TDADOS - Tamanho do arquivo de dados.

Na resposta o Núcleo fornece o endereço de memória onde o PMR deve ser carregado. Com esta informação, **conf** carrega e reloca o PMR, copiando em seguida a Área de Parâmetros e, posteriormente, o conteúdo do arquivo de dados. A Figura 5-4 mostra o mapa de memória após um PMR ser carregado por **conf**. Após a carga CONF, através do Núcleo, ativa o PMR passando o endereço da Área de Parâmetros.

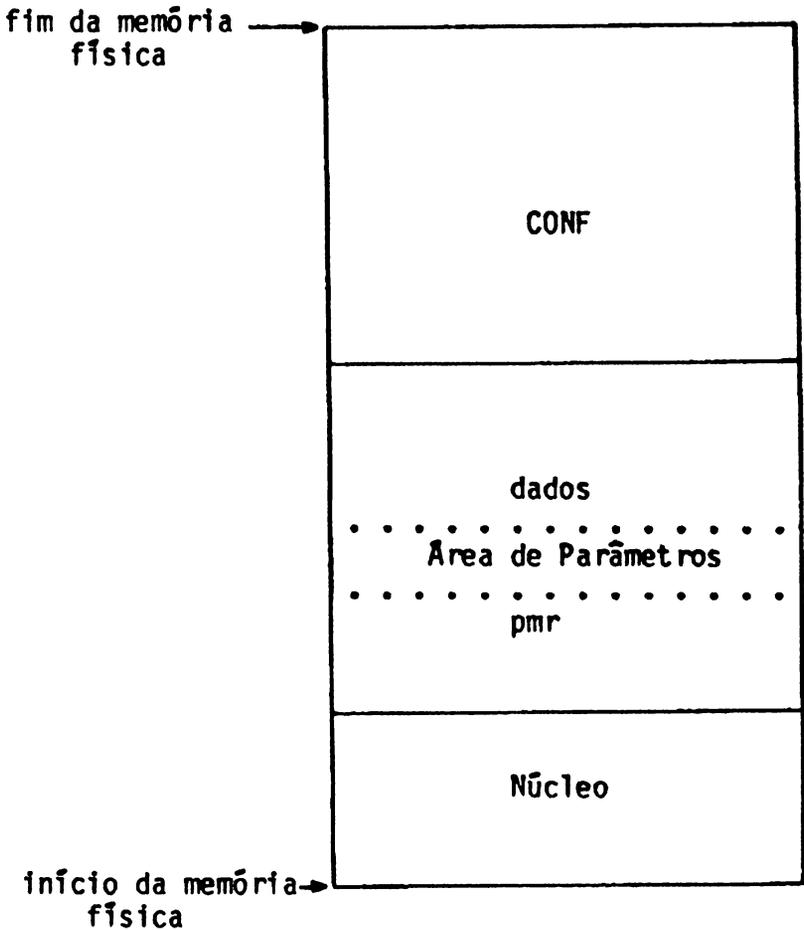


Figura 5-4
MAPA DE MEMÓRIA FÍSICA APÓS A CARGA DE PMR

5.3.4.4 Carga do Servidor

Por último acontece a carga do Servidor. É enviada a mensagem <servidor>..., que significa o nome do módulo de carga do mesmo.

Através dos parâmetros de configuração, **conf** calcula a área de memória adicional, que deverá ser adicionada ao Servidor, e ,através de chamadas de configuração oferecidas pelo Núcleo, cria um Processo de Máquina Virtual (PMV), levando em conta o tamanho da área adicional calculada.

5.3.5 FINAL DE OPERAÇÃO

Após a carga de todos os módulos, um mapa da memória física é semelhante ao mostrado na Figura 5-5. Neste ponto, **conf** desvia para o Núcleo, o que significa a entrada do SOX em operação. Deve-se lembrar que o SOX não tem conhecimento do programa CONF. Portanto, a área de memória inicialmente ocupada por **conf** é encaixada pelo SOX como área de memória livre.

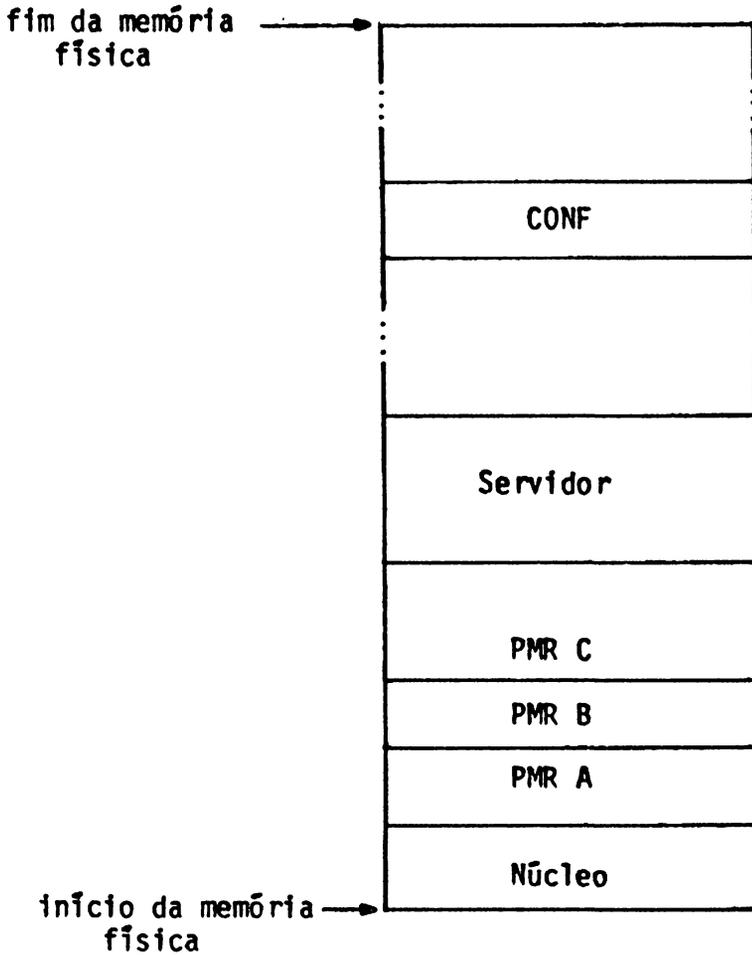


Figura 5-5
MAPA DE MEMÓRIA FÍSICA APÓS A CARGA DO SOX

5.4 VALORES-PADRÃO

As Tabelas 5-1 e 5-2, a seguir, apresentam os valores-padrão assumidos por **conf** para os parâmetros do Núcleo e do Servidor do SOX, não-declarados no arquivo de configuração. NPMR representa o número de PMRs, mais dois, a serem carregados.

5.4.1 PARÂMETROS DO NÚCLEO

Tabela 5-1
VALORES-PADRÃO DOS PARÂMETROS DO NÚCLEO

parâmetro	valor-padrão
N_nucleo	nucleo
N_nds	$(S_npmv + NPMR) * 2$
N_ndpc	$NPMR * 4$
N_ndpi	$NPMR * 4$
N_ndev	$(S_npmv + NPMR) * 2$
N_ndati	$NPMR * 4$

5.4.2 PARÂMETROS DO SERVIDOR

Tabela 5-2
VALORES-PADRÃO DOS PARÂMETROS DO SERVIDOR

pâramet ro	valor-padrão
S_serv	servidor
S_nodo	local
S_npmv	16
S_ntvol	4
S_ndalc	50
S_naca	40
S_ndsmc	8
S_ndsem	4
S_nsem	8
S_ndrsem	16
S_nlocf	20
S_nbuf1k	10
S_nbuf2k	0
S_nbuf4k	0
S_ndblc	8

5.5 MENSAGENS DE ERRO/ADVERTÊNCIA

Neste item apresentaremos as mensagens de erro/advertência que **conf** pode enviar. Será apresentada a mensagem seguida do motivo pelo qual ela foi enviada. Todas as mensagens são precedidas de **conf:**.

5.5.1 MENSAGENS DE ERRO

Estas mensagens são enviadas em caso de erro fatal e fazem com que os procedimentos de inicialização sejam abortados, devendo a máquina ser resetada. Antes de serem enviadas, uma linha é saltada na tela.

"erro na leitura da Seção de Informações."

Se ocorrer erro na leitura da Seção de Informações do volume de carga.

"erro na leitura do BCA raiz."

Se ocorrer erro na leitura do BCA raiz do volume de carga.

"erro no acesso ao arquivo conf.sox."

Se ocorrer erro na procura ou leitura do BCA do arquivo de configuração.

"erro na leitura de arquivo."

Se ocorrer erro na leitura do arquivo de configuração.

"erro de análise do arquivo de configuração,

No trecho "cccccccccccc". Se ocorrer algum erro de sintaxe em um trecho do arquivo de configuração, este será mostrado na tela.

"erro de sintaxe em parametro <R_>!"

Se algum parâmetro "R_" possuir erro de sintaxe. Alguns exemplos deste erro são: omissão de algum parâmetro obrigatório (nome do PMR, nome do núcleo e do carregador do controlador, se existir), omissão do sinal de "=", omissão de alguma pontuação (, (vírgula), : (dois pontos) ou ; (ponto e vírgula)), etc.

"parametro R_ nao e <e> ou <i> ."

Se algum parâmetro R_ não for seguido da letra e, d ou i, etc.

"foi atingido num. maximo de PMRs configuraveis."

Se número de parâmetros R_ ultrapassar máximo permitido. Este número está definido por **conf**, atualmente em 32.

"foi atingido num. max. portas por PMR."

Se número de portas (correspondente ao número de tempos passados) ultrapassar máximo permitido. Este número está definido por **conf**, atualmente em 8.

"foi atingido num. max. controladores configuraveis."

Se número de parâmetros R_ ultrapassar máximo permitido. Este número está definido por **conf**, atualmente em 8.

"dispositivo raiz do sistema nao foi configurado."

Parâmetro, S_draiz não foi citado no arquivo de configuração.

"Maior tipo configurado ultrapassou max. permitido."
Este valor está limitado atualmente em 0xEF.

"nenhum PMR foi configurado com tipo igual ao do dispositivo raiz."
Auto-explicativo.

"erro no acesso ao arquivo <nucleo>."
Se ocorrer algum erro na procura ou leitura do BCA do arquivo com módulo de carga do Núcleo.

"erro na leitura do cabeçalho do Nucleo."
Se ocorrer algum erro na leitura do cabeçalho do módulo de carga do Núcleo.

"erro na leitura de arquivo."
Se ocorrer erro na carga do Núcleo.

"erro na carga de controlador associado a <pmr>."
Se ocorreu algum erro na carga de controlador associado a um PMR imprescindível.

"erro no acesso ao arquivo <pmr>."
Se ocorrer algum erro na procura ou leitura do BCA do arquivo com módulo de carga de um PMR imprescindível.

"tipo de relocacao invalido"
Se existirem informações de relocação no módulo de carga do <PMR> sendo carregado, que não sejam em relação à palavra longa (4 octetos). Deve ser revista a montagem/ligação do módulo.

"erro na carga de <pmr>."

Se ocorrer algum erro na leitura do cabeçalho ou na relocação de um PMR imprescindível.

"erro no acesso ao arquivo <dadpmr>."

Se ocorrer erro na procura ou leitura do BCA do arquivo com dados para configuração de um PMR imprescindível.

"erro na leitura de arquivo."

Se ocorrer erro na carga de um PMR imprescindível ou de seus dados de configuração.

"erro no acesso ao arquivo <servidor>."

Se ocorrer erro na procura ou leitura do BCA do arquivo contendo o módulo de carga do Servidor.

"erro na leitura do cabeçalho do Servidor."

Se ocorrer erro na leitura do cabeçalho do módulo de carga do Servidor.

"erro na leitura de arquivo."

Se ocorrer erro na carga do Servidor.

5.5.2 MENSAGENS DE ADVERTÊNCIA

As mensagens a seguir não interrompem os procedimentos de inicialização, nem saltam uma linha na tela antes de serem enviadas.

"parâmetro S xxx ou N xxx com valor inconsistente; assumido padrão"

Auto-explicativo.

"erro na carga de <nucctr>."

Se ocorrer algum dos seguintes erros no processo de carga em algum controlador: não for encontrado ou não for lido o arquivo contendo o módulo de carga do núcleo do controlador. Não for encontrado ou não for lido o arquivo contendo dados para configuração do núcleo do controlador, se existir. Não for encontrado ou não for lido o arquivo contendo o módulo de carga do representante do controlador, se existir. Não for encontrado ou não for carregado para execução, o configurador do controlador.

"violacao da area de CONF por configurador de <nucctr>"

Auto-explicativo.

"erro na configuracao de <nucctr>"

Se ocorrer algum erro na execução do configurador do controlador.

"<pmr> e controlador associado não foram carregados."

Se ocorreu algum erro na carga de controlador associado a um PMR prescindível.

"<pmr> nao foi carregado."

Se ocorrer algum dos erros citados na carga de um PMR, sendo que este é prescindível.

CAPÍTULO 6

SPOOL DE SAÍDA

6.1 INTRODUÇÃO

O spool do SOX é composto por uma série de comandos que efetuam o controle necessário ao compartilhamento de periféricos de saída de uso exclusivo.

Ao administrador compete selecionar e iniciar os periféricos de saída que serão utilizados e agrupá-los em classes lógicas, a fim de maximizar a ocupação destes.

A iniciação de um periférico consiste em associar a ele um nome, chamado de **impressora lógica**, e definir as suas características. É este nome que será usado pelos usuários do spool para direcionar a saída dos seus relatórios.

Impressoras lógicas que possuem as mesmas características podem ser agrupadas em **classes de impressoras**, permitindo ao usuário não precisar direcionar a saída de seus relatórios para uma impressora específica, mas a qualquer uma que esteja desocupada de determinada classe. Isto permite um maior aproveitamento dos recursos disponíveis.

O administrador pode ainda definir uma **saída default** para aqueles usuários que não direcionarem seus relatórios. Esta saída pode ser uma **classe de impressora** ou uma **impressora lógica**.

A **saída default** pode ser definida ainda pela variável do ambiente do usuário **LPDEST**, que pode ser uma **classe de impressora** ou uma **impressora lógica**. Se presente no ambiente, tem precedência sobre a **saída default** definida pelo administrador para o spool.

6.2 ARQUIVOS AUXILIARES DO SPOOL

/usr/spool/info

Contém a saída default dos relatórios e as associações entre periféricos e impressoras lógicas, e suas características, feitas pelo administrador do sistema. As características de uma associação são: a impressora lógica, o periférico de saída, o estado da impressora, a classe a que pertence, o programa impressor, o número do formulário de impressão e se é desejado ou não cabeçalho antes de cada impressão.

/usr/spool/fila

Contém os pedidos de impressão feitos pelos usuários do spool. Possui entre outras características: nome do arquivo que será impresso, número de cópias, número do formulário de saída, estado do pedido, usuário que fez o pedido, data do pedido, saída da impressão (classe ou impressora-destino), se usuário deseja ser notificado quando a sua impressão terminar, as opções passadas para o programa impressor, o título para o relatório, etc.

/usr/spool/erro

Contém a data, a descrição e o número indicativo do erro (ERRNO) ocorrido nos comandos **lps** (escalonador de pedidos) e **lpd** (impressor de arquivos). São erros pouco usuais como erro em abertura, leitura e gravação dos arquivos, ou erro na execução do programa impressor.

/usr/spool/arqs

Diretório que contém os arquivos que serão impressos (cópias ou associação dos arquivos originais).

Esses arquivos devem ser modificados somente pelos comandos do spool.

6.3 OS COMANDOS

lpi - criando e alterando o spool

O comando **lpi** faz a associação entre um periférico de saída e uma impressora lógica, definindo características desta associação. Quando uma associação é criada, assume-se que a impressora está no estado **ATIVO**, ou seja, a impressora está apta a imprimir (em linha, com papel, etc). Pode-se criar uma associação e colocar a impressora no estado **DESATIVO**, usando-se a opção "-i".

Quando a primeira associação é feita, o comando **lpi** cria todo o ambiente necessário para o funcionamento do spool: cria diretórios, arquivos, tabelas de memória compartilhadas pelos outros comandos e ativa o escalonador de pedidos de impressão **lps**.

Para cada associação feita são definidos: a classe da impressora, se ela passa a ser a saída-default dos relatórios, o formulário de impressão, se imprime cabeçalho antes de cada relatório e o programa impressor. O comando **lpi** permite ainda alterar qualquer uma dessas características, desde que a impressora esteja no estado **DESATIVO**.

lpr - pedindo uma impressão

O comando **lpr** cria uma associação ou cópia do arquivo a ser impresso no diretório /usr/spool/arqs e coloca o registro de impressão na fila de saída (/usr/spool/fila) com suas características: se usuário quer ser notificado após o término da impressão, o formulário de impressão, o número de cópias, a impressora ou classe destino, o título do relatório, se quer ignorar o caractere de "pula pagina" no início do arquivo que vai ser impresso e as opções que devem ser passadas para o programa impressor.

lpa - alterando um pedido de impressão

O comando **lpa** permite ao usuário alterar as características de um pedido de impressão ou da própria saída do relatório.

lpl - consultando informações

O comando **lpl** dá uma visualização de todo o spool, mostrando o estado das associações existentes, o estado dos pedidos de impressão e o estado do spool.

O estado do spool pode ser **ATIVO**, indicando que o escalonador de pedidos (**lps**) está ativado e a tabela de memória compartilhada entre os comandos do spool existe, ou **DESATIVO**, caso contrário.

O estado de um pedido de impressão pode ser **CORRENTE**, quando ele está sendo impresso naquele momento, **FRON**T**O**, quando está para ser impresso ou **PRESO**, quando o usuário mandou parar a impressão na linha corrente via comando **lpa**.

O estado de uma impressora pode ser **ATIVA**, quando ela está apta a imprimir, **DESATIVA**, caso contrário, ou **DESATIVANDO**, indicando que ela passará para **DESATIVA** tão logo a impressão **CORRENTE** termine.

lps - escolhendo quem vai ser impresso

O comando **lps** seleciona para cada impressora no estado **ATIVO**, e que não esteja imprimindo, um pedido para ser impresso no momento.

lpd - imprimindo o arquivo

O comando **lpd** é quem efetivamente imprime um arquivo, tratando as ações feitas sobre a impressão, via comando **lpa**. Este comando pode executar um outro programa impressor para efetuar a impressão, se assim foi definido quando foi feita a associação para aquela impresora.

6.4 INICIANDO/FINALIZANDO O SPOOL

Quando a primeira associação é feita, todo o ambiente do spool é criado, como descrito anteriormente no comando **lpi**.

Quando o sistema é ativado, o fluxo **/etc/rc** (ou qualquer outro), pode conter todas as associações que devem existir no ambiente.

Por exemplo:

```
lpi -c c11 imp1 /dev/lp0
```

```
lpi -c c11 imp2 /dev/lp1
```

```
lpi -d c11
```

que define duas impressoras ativas da classe **c11** e define ainda a classe **c11** como a saída default dos relatórios.

A desativação do spool pode ser incluída no fluxo **/etc/shutdown**:

```
lpi -t
```

que eliminará o arquivo **/usr/spool/info** e desativará o escalonador dos pedidos de impressão **lps**.

Quando uma queda de força ocorre, todas as associações devem ser refeitas como descrito acima, e as impressões que estavam no estado **CORRENTE** serão reimpressas.

Quando o escalonador de pedidos **lps** for desativado, via comando **kill** por exemplo, nenhum relatório será impresso. Para ativá-lo novamente, o comando:

lpi

deve ser dado. Ele ativará o comando **lps** e atualizará a tabela de memória com seu novo identificador de processo.

6.5 O PROGRAMA IMPRESSOR

O programa impressor de uma impressora lógica é aquele que lê o arquivo a ser impresso e escreve na impressora. Ele pode ainda interpretar os caracteres dos registros lidos no arquivo, fazendo com que aconteçam determinadas ações, pré-estabelecidas na impressão. Por exemplo, um programa que aceita como entrada arquivos gerados em formato ASA e cuja saída destina-se a uma impressora que não aceita este formato, interpreta o primeiro caractere de cada linha lida e executa a ação que este caractere estabelece (avança linha, retrocede linha, avança página, etc.).

Se o programa impressor for omitido no comando **lpi**, quando se faz uma associação, assume-se que ele é o **lpd**. O comando **lpd** lê o arquivo a ser impresso, separa as linhas (terminadas por "\n" ou "\f") e as páginas (terminadas por "\f") e imprime linha a linha. A cada linha impressa, verifica se existe alguma ação pedida pelo **lpa** a ser feita sobre a impressão e executa esta ação.

Se outro programa impressor for especificado, o **lpd** prepara todo o ambiente para este programa, ou seja, imprime cabeçalho, se assim for pedido, e coloca em uma área de memória que será acessível a este programa o registro com as características da impressão (**/usr/include/sys/spool.h**). O **lpd**, então, executa o programa impressor e fica esperando o retorno dele.

Se diferente de zero, significa que houve erro, e uma mensagem será colocada no arquivo **/usr/spool/erro**. Se não houve erro, **lpd** notifica o usuário do término da impressão, se assim for pedido, elimina o registro de impressão do arquivo **/usr/spool/fila**, torna a impressora disponível para outra impressão e elimina o arquivo que foi impresso do diretório **/usr/spool/arqs**.

A área de memória com as características da impressão tem o seguinte formato:

```
struct regimp {
    char r_idp;
    char r_nome_arq[200];
    int r_ncopia;
    int r_totlinha;
    int r_totpagina;
    int r_acao;
    int r_comp_acao;
    int r_linha_corrente;
    int r_pagina_corrente;
    int r_nlinhas_pagina;
    int r_periferico[200];
    ;
};
```

É desta área que o programa impressor obtém as informações sobre o arquivo que vai ser impresso. A saber:

r_ipd - Identificador do processo do programa impressor.

r_nome_arq - Caminho do arquivo a ser impresso.

r_ncopia - Número de cópias pedidas do relatório.

r_totlinha - Total de linhas do arquivo na hora do pedido de impressão. As linhas são contadas a cada caracter "\n" e "\f" encontrados.

r_totpagina - Total de páginas do arquivo na hora do pedido de impressão. As páginas são contadas a cada caractere "\f" encontrado.

r_acao - Ação que o comando lpa pediu para ser feita sobre a impressão. Possui um dos seguintes valores (arquivo de include /usr/include/sys/spool.h):

AVANCA_LINHA - Indica que quer avançar linhas na impressão. O número de linhas está em r_comp_acao.

AVANCA_PAGINA - Indica que quer avançar páginas na impressão. O número de páginas está em r_comp_acao.

VOLTA_LINHA - Indica que quer retroceder linhas na impressão. O número de linhas está em r_comp_acao.

VOLTA_PAGINA - Indica que quer retroceder páginas na impressão. O número de páginas está em r_comp_acao.

PRENDE_IMPRESSAO - Indica que quer a impressão parada na linha corrente.

SOLTA_IMPRESSAO - Indica que quer que a impressão continue a partir da linha corrente.

REIMPRIME_ARQ - Indica que quer recomeçar a impressão a partir da primeira linha.

r_linha_corrente - É iniciado com o total de linhas do arquivo. O programa impressor pode decrementar este campo a cada linha impressa. O comando **lp** busca neste campo o número de linhas que ainda faltam para terminar a impressão.

r_pagina_corrente - É iniciado com o total de páginas do arquivo. Nenhum comando se utiliza deste campo, a não ser eventualmente o próprio programa impressor do usuário.

n_linhas_pagina - Número de linhas por página de impressão. De uso exclusivo do programa impressor do usuário. Nenhum comando do spool faz uso deste campo.

r_periferico - Nome do dispositivo que o programa impressor deve abrir antes de iniciar a impressão.

O comando **lpd** ativa o programa impressor passando nos primeiros argumentos as opções que o usuário passou para ele via opção "o" do **lpr**. Logo após estas opções está uma chave para obtenção do endereço da área descrita acima. Este parâmetro é uma variável do tipo **long** e é passado para a função **getspool()** (definida no `include /usr/include/sys/spool1.h`), que devolverá o endereço desta área.

```
struct regimp *getspool(chave)  
long chave;
```

A utilização desta área é opcional. Caso o programa impressor não queira se utilizar desta área, após a chave são passados o nome do arquivo a ser impresso, o periférico de saída e o número de cópias pedidas, nesta ordem. Mas o usuário deve ter em mente que o comando **lp** imprimirá sempre a mesma linha como linha corrente, porque o campo **r_linha_corrente** da estrutura **regimp** não está sendo atualizado.

Exemplos:

- 1) Programa impressor é um shell-script que chama o cat para fazer a impressao:

```
cat <$2 > $3
```

- 2) Programa impressor é um programa C (não está preparado para receber opções):

```
#include "/usr/include/fcntl.h"
#include "/usr/include/sys/spool.h"

main(argc, argv)
int argc;
char **argv;
{

long chave = atol (argv[1]);
int ncopia = atoi (argv[4]);
char buf[512];
int nl,nb,nlp;
struct regimp *ptabmem;

ptabmem = getspool(chave);

if ((nl = open (argv[2], O_RDONLY)) <= 0)
    exit(1);

if ((nlp = open (argv[3], O_RDONLY)) <= 0)
    exit(1);

while ((nb = read (nl, buf, 512)) > 0) {
    write (nlp, buf, nb);
    if (ptabmem->r_acao != NAO_TEM_ACAO) {
```

```

        .
        .
        .
        ptabmen->r_acao = 0;
    }
}
exit (0);
}

```

6.6 AMBIENTE DE INSTALAÇÃO

A seguir são apresentados todos os diretórios e arquivos que fazem parte do spool, com seu dono, grupo e modo respectivamente:

DIRETÓRIOS E ARQUIVOS	DONO GRUPO MODO
/usr/spool	bin bin drwxr-xr-x
/usr/spool/arqs	bin bin drwxr-xr-x
/usr/spool/info	bin bin -rw-----
/usr/spool/fila	bin bin -rw-----
/usr/spool/erro	bin bin -rw-rw-rw-
/usr/spool/lpd	root bin -rws-----
/usr/spool/lps	bin bin -rws-----
/usr/bin/lpl	bin bin -rwsr-xr-x
/usr/bin/lpr	bin bin -rwsr-xr-x
/usr/bin/lpl	bin bin -rwsr-xr-x

Administrador do Sistema S0X

/usr/bin/lpa		root bin -rwsr-xr-x
/usr/include/sys/spool1.h		bin bin -rwxr-xr-x

Se os modos ou donos dos comandos e diretórios forem alterados, eles podem não mais funcionar.

DIRETÓRIO BÁSICO DO SOX

Este anexo descreve a estrutura básica da árvore hierárquica do volume-raiz do SOX.

Esta estrutura deve normalmente estar presente em todo do Sistema SOX. Os comandos e processos fornecidos com o SOX supõem a estrutura definida a seguir.

Aquí será descrita apenas a estrutura básica, pois a estrutura completa é dependente de máquina (X-10, SOX-PC, SOX-500) e de particularidades da instalação.

O diretório / é o raiz do sistema de arquivos. A seguir serão descritas as finalidades dos demais arquivos da estrutura.

/bin - Diretório que contém os diversos comandos de usuário do sistema. Normalmente, usuários não devem criar arquivos nesse diretório, embora estes possam ler e executar os arquivos aqui contidos.

/etc - Diretório que contém comandos de uso do administrador e arquivos de controle do sistema. Normalmente não deve estar acessível aos usuários.

/dev - Diretório que contém os arquivos especiais bloco e caractere (dispositivos de E/S) do sistema. Normalmente, usuários não devem criar arquivos neste diretório.

- /tmp** - Diretório que contém os arquivos temporários de trabalho dos comandos do sistema ou de outros processos do sistema. Não deve ser usado pelos usuários. Normalmente, o administrador deve providenciar para que este diretório seja esvaziado sempre que o sistema entrar em operação.
- /mnt** - Este é um arquivo tipo diretório, que está sempre vazio. É usado como "ponte" para montagem de volumes. O administrador pode criar quantos arquivos deste tipo forem necessários, dando os nomes mnt0, mnt1, etc.
- /usr/bin** - Diretório de uso dos usuários e do sistema. Pode conter comandos do sistema e comandos criados pelos usuários.
- /usr/tmp** - Diretório de uso dos usuários e do sistema. Contém os arquivos temporários dos usuários. Normalmente, o administrador deve providenciar para que este arquivo seja esvaziado sempre que o sistema entrar em operação.
- /servidor** - Arquivo que contém o executável do módulo Servidor SOX. Este arquivo é acessado pelo Configurador, ao ser feita a carga do sistema. Não deve ser acessado por usuário algum. Caso o administrador deseje alterar seu conteúdo, deve se referenciar ao Capítulo 5 deste manual.
- /nucleo** - Arquivo que contém o executável do módulo Núcleo do SOX. Comportamento idêntico ao arquivo anterior.

- /pmrxxx - Arquivo que contém o executável dos módulos PMRs (programas de controle de dispositivos) do SOX. Comportamento idêntico ao anterior.
- /dadxxx - Arquivo que contém dados de configuração do PMR correspondente. Pode existir ou não (ver Capítulo 5).
- /conf - Arquivo que contém o executável do Configurador do SOX.
- /conf.sox - Arquivo que contém os parâmetros para o Configurador do SOX. Para se alterar este arquivo, deve-se referenciar ao Capítulo 5 deste manual.

Página intencionalmente em branco.

ANEXO B

ARQUIVOS DE ADMINISTRAÇÃO

Este anexo contém a descrição dos arquivos que devem ser utilizados e conhecidos pelo Administrador do SOX.

Os arquivos estão organizados em ordem alfabética e nomeados apenas pelo último nome do caminho, quando for o caso. O nome completo do caminho é fornecido no item "Arquivos".

Algumas estruturas de controle importantes foram incluídas neste anexo, mesmo não correspondendo a arquivos.

Nome

BCA - Formato de um BCA (Bloco de Controle de Arquivo).

Sintaxe

```
#include sys/types.h
#include sys/aca.h
```

Descrição

A entidade arquivo no SOX é implementada através de uma estrutura de controle chamada BCA. A identificação de um arquivo se dá pelo índice de sua estrutura de controle dentro da área do volume reservada para conter os BCAs. O índice começa de 1, sendo que o BCA de índice 1 é reservado e o BCA de índice 2 corresponde ao arquivo diretório-raiz do sistema de arquivos.

Quando um arquivo é aberto, esta estrutura de controle é copiada em memória, passando a constituir uma área chamada Área de Controle de Arquivo (ACA). A estrutura ACA é composta da estrutura BCA e outras informações que são necessárias durante o tempo em que o arquivo estiver aberto. As estruturas BCA e ACA têm sua definição dada no arquivo /usr/include/sys/aca.h.

Arquivos:

```
/usr/include/sys/aca.h
```

Nome

checklist - Arquivo de parâmetros para o fsck.

Descrição

Se o comando **fsck** for executado sem parâmetros, ele vai obter deste arquivo os arquivos especiais que contêm os sistemas de arquivos que devem ser verificados/recuperados. Cada linha deste arquivo deve conter um arquivo especial. O **fsck** vai automaticamente processar o sistema de arquivos contido em cada linha do arquivo checklist. Ver descrição do comando **fsck** no "Manual de Referência Comandos SOX".

Arquivos

/etc/checklist

Nome

conf - Configurador.

Descrição

Este arquivo contém o configurador do SOX. Ver descrição do configurador no Capítulo 5 deste manual.

Arquivos

/conf

Nome

conf.sox - Arquivo de parâmetros de configuração.

Descrição

Este arquivo contém os parâmetros de configuração do SOX. A descrição deste arquivo se encontra no Capítulo 5 deste manual.

Arquivos

/conf.sox

Nome

console - Terminal-console do sistema.

Descrição

Este é um arquivo que representa o console do sistema. Normalmente está associado a um arquivo especial caractere, associado a um determinado terminal. Os diálogos entre os diversos comandos e processos de operação e o administrador do sistema serão feitos através desse arquivo.

Arquivos

/dev/console

Nome

dir - Formato de uma entrada no diretório.

Sintaxe

```
#include <sys/types.h>
#include <sys/dir.h>
```

Descrição

Um arquivo tipo diretório tem o mesmo comportamento de um arquivo regular, exceto que nenhum usuário (nem mesmo Superusuário) pode realizar operações de escrita sobre ele. Este arquivo é composto de entradas, com o formato dado pela seguinte estrutura:

```
#define DIRSIZ 14
struct direct
{
    ino_t  d_ino;
    char  d_name[DIRSIZ];
};
```

Por convenção, a primeira entrada de um arquivo diretório é `.` (ponto), que representa o próprio arquivo, e a segunda entrada é `..` (ponto-ponto), representando o diretório-pai deste arquivo (o imediatamente acima na árvore hierárquica). Para o diretório-raiz do volume raiz, a entrada `..` (ponto-ponto) tem o mesmo significado da entrada `.` (ponto).

Arquivos:

```
/usr/include/sys/dir.h
```

Nome

erro - Arquivo de comunicação do sistema.

Descrição

Este arquivo é usado para comunicação entre o usuário e o sistema. É através dele que o processo-console e o comando **ip** comunicam-se com o sistema.

Normalmente este arquivo deve ter permissões tais, que somente os Superusuário tenham acesso, evitando assim que usuários normais respondam a uma intervenção.

Arquivos

/dev/erro

Nome

gettydefs - Tabela de controle do **getty**.

Descrição

Este arquivo contém uma tabela para controlar a execução do **getty**. Sua descrição pode ser encontrada no Capítulo 2 deste manual, na descrição do comando **getty**.

Arquivos

/etc/gettydefs

Nome

group - Arquivo que nomeia os grupos.

Descrição

group contém, para cada grupo, as seguintes informações:

- . Nome do grupo;
- . Senha criptografada;
- . Valor numérico do identificador de grupo;
- . Lista de usuários permitidos no grupo, separados por vírgula.

Este é um arquivo ASCII. Os campos são separados por dois pontos. Cada grupo é separado do próximo por um caractere **newline**.

Este arquivo pode ser usado, por exemplo, para mapear os valores numéricos dos identificadores de grupo em seus respectivos nomes.

Arquivos

/etc/group

Nome

inittab - Tabela de controle do **init**.

Descrição

Este arquivo contém uma tabela para controlar a execução do **init**. Sua descrição pode ser encontrada no Capítulo 2 deste manual, na descrição do comando **init**.

Arquivos

/etc/inittab

Nome

issue - Tela de apresentação do sistema.

Descrição

Este é um arquivo ASCII que contém o texto a ser impresso, antes do pedido de abertura de sessão. Este arquivo é lido pelo processo getty e impresso em todo terminal sempre que estiver sendo ativado. A existência deste arquivo não é obrigatória.

Arquivos

/etc/issue

Nome

lost+found - Diretório para arquivos perdidos.

Descrição

Este arquivo tipo diretório contém como entradas arquivos que são considerados perdidos na execução do comando **fsck** (verifica inconsistências em volumes), ou seja, arquivos cujo BCA está marcado como ocupado, mas que não possuem referências no diretório. Como não existe uma maneira de descobrir os nomes destes arquivos, **fsck** os cadastra no diretório **lost+found** com um nome que é a representação ASCII de seus ibcas correspondentes, completando com zeros à esquerda até cinco dígitos. A identificação do arquivo, após sua inserção em **lost+found**, deve ser feita através de um comando **ls -l**, para que se possa localizar o dono deste e informá-lo, para que este decida seu destino.

Como a criação de uma nova entrada em um diretório, por um utilitário, poderia causar complicações desnecessárias (alocação de um novo bloco para o diretório, por exemplo), principalmente em um volume que se considera danificado logicamente, achou-se por bem que **fsck** não deveria criar novas entradas em **lost+found**. **fsck** deve aproveitar entradas existentes e livres, já que um diretório no SOX apenas cresce e, quando uma entrada é removida, apenas o campo relativo ao ibca é zerado, permanecendo o diretório com o tamanho antigo.

O procedimento correto antes da execução de **fsck** é criar o diretório `lost+found` (se não existe) na raiz do volume sobre o qual **fsck** irá atuar, e inserir nele entradas que logo após são removidas, mas que garantam um espaço para os arquivos achados na verificação. Pode-se criar um arquivo de comandos, por exemplo, de nome `/etc/mklost+found`, para realizar esta função automaticamente. Um possível conteúdo para este arquivo é dado a seguir:

```
mkdir /lost+found
cd /lost+found
i=1
while [ $i -lt 30 ]
do
    echo > $i
    i=`expr $i + 1`
done
rm *
```

Note-se que a criação de `/lost+found`, na primeira linha é opcional, se o diretório já existe, e que o número de arquivos a serem inseridos neste, no exemplo 30, pode ser devidamente alterado para o número de entradas necessárias para abrigar arquivos perdidos.

Arquivo

```
/lost+found
```

Nome

mnttab - Tabela de volume montados.

Sintaxe

```
#include <sys/types.h>
#include <mnttab.h>
```

Descrição

Este arquivo contém a tabela de volumes montados no sistema, através da chamada mount. A seguinte estrutura define cada entrada nesta tabela:

```
struct mnttab
{
    char    mt_dev[10];      /* dispositivo */
    char    mt_filsys[10];  /* diretório */
    short   mt_ro_flg;      /* flags de abertura */
    time_t  mt_time;        /* data */
};
```

O significado de cada campo é o seguinte: `mt_dev` é o nome (terminado por NULL) do arquivo no diretório `/dev`, associado ao dispositivo onde o volume está montado; `mt_filsys` é o nome (terminado por NULL) do diretório que representa a raiz do volume montado. `mt_ro_flg` representa o flag de abertura do volume (Teitura somente ou leitura/escrita) e `mt_time` representa a data da montagem do volume.

Arquivos

`/etc/mnttab`

Nome

motd - Mensagem do dia.

Descrição

Este é um arquivo que tem sua definição dada por convenção. Cabe ao administrador do sistema decidir, levando em consideração as necessidades de sua instalação, se vai ou não seguir esta convenção. O SOX não impõe restrições sobre este arquivo.

Normalmente esse arquivo contém uma mensagem, que deve ser curta, que será impressa nos terminais sempre que um usuário abrir uma sessão. O administrador deve sempre, no início do dia, atualizar este arquivo (usando o **ed** ou qualquer outro comando equivalente) para que os usuários possam ser informados de fatos importantes ocorridos na instalação.

Para que este arquivo seja impresso, é necessário que seja incluído no arquivo `/etc/profile` um comando para tal. Por exemplo,

```
cat /etc/motd
```

Arquivos

```
/etc/motd
```

Nome

news - Arquivo de mensagens para os usuários.

Descrição

Este é um arquivo tipo diretório que é usado pelo comando **news**. Ver descrição desse arquivo no "Manual de Referência Comandos SOX", na descrição do comando **news**.

Arquivos

/etc/news

Nome

null - Arquivo nulo.

Descrição

Este arquivo representa um dispositivo-fantasma. Dados escritos neste arquivo são simplesmente descartados, enquanto que leituras sobre este arquivo retornam sempre 0.

Arquivos

/dev/null

Nome

passwd - Arquivo de cadastro de usuários.

Descrição

O arquivo /etc/passwd contém, para cada usuário, as seguintes informações:

- . Nome;
- . Senha criptografada (pode estar vazio);
- . Valor numérico do identificador de usuário;
- . Valor numérico do identificador do grupo (po de estar vazio);
- . Comentário (pode estar vazio);
- . Diretório inicial de trabalho;
- . Interpretador de comandos a ser usado (pode estar vazio).

Este é um arquivo ASCII, composto de entradas. Cada entrada, correspondendo a um usuário, é separada da outra pelo caractere newline. Os campos de cada entrada são posicionais, separados dos outros pelo caractere : (dois pontos).

Este arquivo é usado pelo programa **login** para verificar a validade dos nomes e senhas das sessões fornecidas pelos usuários. Pode ser usado também, por exemplo, para mapear o valor numérico do identificador de usuário em seu respectivo nome. Nome é uma cadeia que identifica um usuário. Sua composição segue as mesmas regras para nome de arquivos.

A descrição do uso desse arquivo pode ser encontrada no Capítulo 4 deste manual.

Arquivos

/etc/passwd

Nome

profile - Arquivo de comandos para início de sessão.

Descrição

Se o arquivo `/etc/profile` existe no sistema de arquivos, o interpretador de comandos-padrão do `SOX` (`/bin/sh`) irá executar este arquivo, sempre que for ativado através do programa `login`. Este arquivo deve conter comandos para o interpretador para, por exemplo, inicializar ou alterar variáveis do ambiente, para listar o arquivo `/etc/motd` (mensagem do dia), etc.

Além disto, se o arquivo `.profile` existe no diretório corrente inicial de uma sessão, será também executado como descrito acima.

O primeiro arquivo é usado para a execução de procedimentos que devem atingir todos os usuários do sistema, enquanto o segundo deve ser usado para procedimentos particulares a um determinado usuário.

Arquivos

`/etc/profile`
`.profile`

Nome

rc - Arquivo de comandos para ativação do sistema.

Descrição

Este é um arquivo que tem sua definição dada por convenção. Cabe ao administrador do sistema decidir, levando em consideração as necessidades de sua instalação, se vai ou não seguir esta convenção. O SOX não impõe restrições sobre este arquivo.

Normalmente este arquivo contém comandos para o interpretador de comandos-padrão do sistema. Para que ele seja executado, deve existir uma entrada no arquivo /etc/inittab referente a ele, estando a entrada marcada, para que seja executada sempre que o sistema for carregado ou que mudar de nível de operação (sempre com espera).

Como tarefas típicas que podem ser incluídas neste arquivo, podemos citar a remoção de arquivos temporários, a montagem de volumes, a execução de tarefas de manutenção/administração tais como salvamento de volumes, etc.

Arquivos

/etc/rc

Nome

setinf - Seção de informação de um volume.

Síntaxe

```
#include <sys/types.h>  
#include <sys/setinf.h>
```

Descrição

Todo volume no SOX deve conter, em um endereço físico fixo (no octeto de ordem 1024 no dispositivo de residência do volume), uma área que será considerada a Seção de Informação do volume. Esta área contém todas as informações necessárias para que o SOX possa manter e administrar o sistema de arquivos contido no volume. Qualquer inconsistência nesta área, encontrada pelo SOX, fará com que o volume seja considerado inválido, não sendo possível a sua montagem.

O conteúdo dessa área é dado pelo arquivo /usr/include/sys/setinf.h.

Arquivos:

/usr/include/sys/setinf.h

Nome

shutdown - Arquivo para desativação do sistema.

Descrição

Este é um arquivo de comandos para o interpretador de comandos, e deve ser usado para se desativar o sistema. Devem ser incluídos neste arquivo todos os comandos para implementar os procedimentos de desativação necessários à instalação.

A seguir, é dado um exemplo de um conteúdo possível para esse arquivo:

```
cd /
/etc/wall<< !
O sistema vai sair de operação em 1 minuto.
Por favor, fechem suas sessões.
!
sleep 60
/etc/sfo
```

Se for necessário ao administrador executar tarefas em modo monousuário, antes de desativar o sistema, o arquivo pode ser alterado para:

```
cd /  
/etc/wall<<!  
O sistema vai sair de operação em 1 minuto.  
Por favor, fechem suas sessões.  
!  
sleep 60  
killall  
init s
```

Depois disto, o administrador executa o que precisar, tendo o cuidado de executar o comando `/etc/sfo` antes de desligar o sistema.

Arquivos

`/etc/shutdown`

Nome

termio - Interface de terminal.

Sintaxe

```
#include <termio.h>

ioctl (n1, comando, arg)
struct termio *arg;

ioctl (n1, comando, arg)
int arg;
```

Descrição

Esta sessão descreve a interface para linha de comunicação assíncrona, independente de hardware, oferecida pelo SOX.

Os usuários desta interface serão tanto o operador dos terminais quanto programadores de aplicação, que escrevem dados na tela e/ou lêem dados do teclado de terminais. Os recursos oferecidos ao operador são aqueles de utilidade indiscutível, não importando o tipo de aplicação: correção de erros elementares de digitação, cancelamento da aplicação (abandono), interrupção temporária da saída de caracteres e sinalização de fim de entrada.

Recursos adicionais podem ser definidos, desde que programados dentro das aplicações (ver comando stty).

É oferecida também, a possibilidade de configurar conversões sistemáticas de caracteres digitados, a fim de contornar limitações na geração de caracteres existentes em muitos terminais, e também conversões dos caracteres produzidos pelas aplicações e destinadas à tela (ou teleimpressora). Como, por exemplo, conversão maiúscula-minúscula, conversão (ver XCASE), conversão CR-NL, etc.

A interface oferece dois modos de operação selecionáveis pelas aplicações - modo canônico e modo "raw". No modo canônico, os caracteres são agrupados em linhas. O terminador-padrão de uma entrada é o caractere NL. Existem, ainda, terminadores adicionais (EOL e EOF) que podem ser configurados. Os terminadores NL e EOL são sempre retornados para as aplicações, juntamente com os outros caracteres digitados. O modo canônico destina-se a aplicações que lêem linhas de tamanho variável, constituídas de caracteres representáveis na tela.

O modo "raw" destina-se a usos não elementares da interface. Como característica geral, apenas o fato de que linhas de entradas não são separadas por terminadores. O término de uma linha de entrada pode ser dado, por exemplo, quando já foi lido um número predeterminado de caracteres ou quando um intervalo de tempo predeterminado foi expirado.

Cada função disponível para o operador é acionada pela digitação de um único caractere. As funções são as seguintes:

ERASE : Elimina o último caractere da entrada corrente, caso haja algum.

KILL : Elimina todos os caracteres da entrada corrente. Quando um erro é percebido tardiamente, pode ser mais rápido usar a função **KILL** do que utilizar várias vezes a função **ERASE**.

NL : Terminador-padrão de uma entrada no modo canônico.

EOL : Terminador opcional de uma entrada no modo canônico. O tratamento é mesmo do **NL**.

EOF : Terminador opcional de uma entrada. No modo canônico, as entradas terminadas por **NL** ou **EOL** não provocam nunca um retorno de tamanho lido zero, já que estes terminadores são incluídos na fila de entrada. Quando é digitado um **EOF**, a entrada é considerada terminada, mas o **EOF** não é incluído na fila de entrada. Desta forma, quando é digitado no início da entrada, a leitura é respondida com tamanho lido nulo, que é a indicação de fim de arquivo.

INTR : Ao receber a tecla **INTR**, o sistema é comunicado da ocorrência do sinal **SIGINT** e as filas de entrada e saída de caracteres são limpas. Normalmente, o sinal **SIGINT** provoca o cancelamento de todos os processos da sessão, exceto o interpretador de comandos.

QUIT : Gera um sinal **SIGQUIT**. É idêntico ao tratamento feito para a tecla **INTR**.

STOP (XOFF) : Quando é digitada a tecla **XOFF**, a transmissão para o terminal é suspensa. É utilizada para evitar que uma mensagem desapareça da tela pelo efeito de rolamento.

START (XON) : Cancela o efeito de suspensão da transmissão para o terminal provocada por uma tecla **XOFF**, digitada anteriormente.

MIN : Caractere de controle quando modo "raw" ativado (ver caracteres de controle).

TIME : Caractere de controle quando modo "raw" ativado (ver caracteres de de controle).

Quando qualquer uma das funções acima está habilitada pela aplicação corrente, a tecla associada não entra na fila de entrada. As duas únicas exceções são os terminadores NL e EOL, que são sempre colocados na fila e lidos pela aplicação.

Quando qualquer uma das funções acima está inibida pela aplicação corrente, as teclas associadas a elas são tratadas como caracteres de dados comuns. Entram, portanto, na fila de entrada e são lidos pela aplicação corrente.

As teclas associadas às funções ERASE, KILL e EOF podem ser incluídas na fila de entrada e lidas pela aplicação corrente, mesmo se as funções estiverem habilitadas. Basta que o operador as faça preceder pelo caractere \ . São configuráveis pelas aplicações as teclas associadas às funções INTR, QUIT, KILL, EOF e EOL.

O elemento fundamental para interface com um terminal é a estrutura de dados onde ficam todos os parâmetros de configuração, seleção de funções habilitadas e caracteres configuráveis. Esta estrutura está definida no arquivo /usr/include/termio.h e é manipulada pela chamada ao sistema `ioctl()`.

```
ioctl(n1, comando, ptermio)
int n1;
int comando;
struct termio *ptermio;
```

Onde:

n1 - Número lógico do terminal;
comando - Código do comando;
ptermio - Ponteiro para a estrutura termio.

O código do comando identifica a operação de controle desejada. As operações são as seguintes.

- TCGETS** - Lê estrutura termio do terminal e copia para ptermio.
- TCSETS** - Altera a estrutura termio do terminal, conforme a estrutura apontada por ptermio. A alteração é imediata.
- TCSETAW** - Altera a estrutura termio do terminal, conforme a apontada por ptermio, após esperar que todos os caracteres da fila de saída sejam escritos.

TCSETAF - Altera a estrutura termio do terminal, conforme a apontada por ptermio, após es perar que todos os caracteres da fila de saída sejam escritos e descarta todos os caracteres da fila de entrada.

Um outro modo de se usar a função foctl para termi nal é da seguinte forma:

```
foctl (n1, comando, arg)
int n1;
int comando;
int arg;
```

Onde:

n1 - Identificador do terminal;
comando - Código do comando;
arg - Argumento.

Os comandos são:

TCSBRK - Espera que todos os caracteres da fila de saída sejam escritos e então, se arg igual a 0, envia um **break** ao terminal.

TCXONC - Se arg igual a 0, suspende a escrita. Se arg for positivo, recomeça a escrita.

TCFLSH - Se arg igual a 0, descarta todos os caracter es da fila de entrada;
se arg igual a 1, descarta todos os caracter es da fila de saída;
se arg igual a 2, descarta os caracteres de ambas as filas.

SETGRP - Define identificador de grupo e habilita recepção de sinais SIGINT, SIGQUIT e SIGHUP.

RESETGRP - Invalida identificador de grupo e inibe recepção de sinais SIGINT, SIGQUIT e SIGHUP.

OBTIPTV - Obtém tipo do terminal de vídeo.

A estrutura de controle de terminal tem o formato abaixo.

```
typedef struct termio {
    unsigned c_iflag; /* parâmetros de entrada */
    unsigned c_oflag; /* parâmetros de saída */
    unsigned c_cflag; /* parâmetros configuração */
    unsigned c_lflag; /* parâmetros de modos */
    char c_line; /* disciplina da linha */
    unsigned char c_cc[8]; /* caracteres de /*
                                /* controle */
};
```

PARÂMETROS DE ENTRADA

A variável `c_iflag` define o tratamento que deve ser efetuado sobre todo caractere vindo do terminal. Estes tratamentos correspondem às funções de conversão dos caracteres digitados, controle de erros e de fluxo. Portanto, não devem ser modificados por aplicações independentes do terminal real. Os parâmetros são passados ligando ou desligando um bit específico da variável `c_iflag`. São eles:

IGNBRK

Se ligado, toda ocorrência de **break** é ignorada. Se desligado, o tratamento de **break** depende do campo BRKINT.

BRKINT

Se BRKINT ligado (e IGNBRK desligado) sempre que ocorrer um **break**, um sinal SIGINT é gerado e ambas as filas de entrada e saída serão descartadas. Se BRKINT e IGNBRK estão desligados, o **break** é considerado um caractere nulo com erro de **framing**.

IGNPAR

Se ligado, os caracteres com erro de paridade ou de **framing** serão ignorados

PARMRK

Só tem significado se IGNPAR desligado. Se PARMRK desligado, todo caractere com erro de **framing** ou paridade é representado na fila de entrada pelo caractere nulo. Esta configuração de parâmetros só serve para aplicações em que o caractere nulo não possa ocorrer de outra forma. Aplicações que precisem de transparência absoluta têm que ligar PARMRK.

Se PARMRK ligado, erros de recepção são representados na fila de entrada pela seqüência 0xFF 0x00 0xXX, onde 0xXX é o caractere com erro de paridade ou **framing**.

INPCK

Se desligado, toda ocorrência de erro de paridade é ignorada. Então, caracteres com erro de paridade não se distinguem dos corretos. Pode parecer que, ao ligar PARMRK, a aplicação apenas transfere a ambiguidade do caractere nulo para o caractere 0xFF, mas a ambiguidade é resolvida pelo parâmetro ISTRIP.

ISTRIP

Se ISTRIP estiver ligado, todo caractere recebido é convertido para 7 bits, pelo apagamento do bit mais significativo. Se PARMRK e ISTRIP ligados, o caractere 0xFF só pode aparecer na fila de entrada se ocorrer erro de recepção, evitando a ambiguidade mencionada no item anterior. Se PARMRK ligado e ISTRIP desligado, os caracteres 0xFF recebidos sem erro são duplicados na fila de entrada e a ambiguidade desaparece também.

INLCR

Se INLCR ligado, caracteres NL recebidos são convertidos para CR.

IGNCR

Se IGNCR ligado, caracteres CR recebidos são ignorados.

ICRNL

Só tem sentido se IGNCR desligado. Se ICRNL ligado, caracteres CR recebidos são convertidos para NL.

IUCLC

Se IUCLC ligado, maiúsculas recebidas são convertidas para minúsculas.

IXON

Se IXON ligado, habilita o controle de fluxo xon/xoff no sentido sistema→terminal, ou se ja, o operador pode digitar **XOFF** para interromper o fluxo de dados para a tela, e digitar **XON** para prosseguir. Se IXON desligado, **XON** e **XOFF** são vistos como caracteres comuns de dados.

IXANY

Sõ tem sentido se IXON ligado. Se IXANY ligado, quando o fluxo de dados estiver suspenso por ter sido digitado **XOFF**, a transmissão para o terminal recomeça quando qualquer caractere é digitado, não necessariamente **XON**. Se ambos IXON e IXANY estão ligados, o caractere **XOFF** funciona alternadamente como **XON** e **XOFF**.

IXOFF

Se IXOFF ligado, habilita controle de fluxo xon/xoff no sentido terminal→sistema. Então, quando a fila de entrada cresce além de 250 caracteres, é enviado um xoff ao terminal e, quando a fila recomeça a ser consumida por leituras, e o número de caracteres na fila desce abaixo de 200 caracteres, é enviado um xon autorizando o terminal a prosseguir a transmissão. Se IXOFF desligado, a fila de entrada fica no máximo com 256 caracteres e, em seguida, passa a ignorar os caracteres recebidos.

PARÂMETROS DE SAIDA

A variável `c_oflag` define o tratamento que deve ser realizado sobre todos os caracteres destinados ao terminal, sejam caracteres vindos nas chamadas de escrita emitidas pelas aplicações ou ecoados. Os parâmetros são passados ligando ou desligando um bit específico da variável `c_oflag`. São eles:

OPOST

Se OPOST desligado, nenhum dos parâmetros descritos adiante é considerado.

OLCUC

Converte maiúsculas destinadas ao terminal para minúsculas.

ONLCR

Converte caracteres NL destinados ao terminal para a seqüência CR-NL.

OCRNL

Converte caracteres CR destinados ao terminal para NL.

ONOCR

Ignora caracteres CR destinados ao terminal, se o cursor estiver no início de uma linha (coluna 0).

ONLRET

Se ONLRET ligado, é considerado que o cursor passa para o início da próxima linha, após a transmissão de um caractere NL. Se desligado, é considerado que o cursor permanece na mesma coluna, na próxima linha. Esta distinção tem importância se estão sendo substituídos caracteres de tabulação por brancos (ver TABDLY - TAB3).

OFILL

Se OFILL ligado, caracteres de enchimento serão usados para provocar retardo. Se OFILL desligado, o retardo será temporizado.

OFDEL

Só tem sentido se OFILL ligado. Se OFDEL ligado, o caractere DEL (0x7F) será usado para enchimento. Se OFDEL desligado, será usado o caractere nulo.

Em terminais cuja comunicação visual com o operador se dá através de impressão sobre papel, o tratamento de alguns caracteres de controle inclui movimentos mecânicos longos do carro de impressão ou do papel. Muitos destes terminais não são capazes de sincronizar os movimentos mecânicos com a recepção dos caracteres seguintes do fluxo de dados, resultando em perda destes caracteres ou sua impressão durante o movimento, prejudicando a formatação do texto.

Alguns terminais de vídeo antigos não são capazes de efetuar o tratamento de um caractere em paralelo com a recepção dos caracteres seguintes. Então, a recepção de um caractere cujo tratamento é lento (FF - alimentação de formulário, por exemplo, provoca a perda dos caracteres que chegam durante o tratamento. Para suprir tais deficiências podem ser configurados retardos no fluxo de dados, logo após a transmissão de qualquer um dos caracteres seguintes: NL, CR, TAB, BS, VT e FF. Os retardos podem ser feitos através de temporização ou do envio de caracteres nulos, ou do envio de caracteres 0xFF, chamados de caracteres de enchimento. A tabela a seguir dá os parâmetros para cada tipo de caractere:

PARÂMETRO CARACTERE	VALOR DO PARÂMETRO	ENCHIMENTO	TEMPORIZAÇÃO (ms)
NLDLY / NL (avanço de linha)	NL0	0	0
	NL1	2	100
CRDLY / CR (retorno de carro)	CR0	0	0
	CR1	2	50 * c (1)
	CR2	4	100
	CR3	4	150
TABDLY / TAB (tabulação horizontal)	TAB0	0	0
	TAB1	2	50 * c (1)
	TAB2	2	100
	TAB3	(2)	-
BSDLY / BS (volta caractere)	BS0	0	0
	BS1	1	50
VTDLY / VT (tabulação vertical)	VT0	0	0
	VT1	(3)	2000
FFDLY / FF (alimenta formulário)	FF0	0	0
	FF1	(3)	2000

- (1) c é o deslocamento em colunas.
 (2) Expande em brancos.
 (3) Não é permitido enchimento.
 Deve sempre temporizar.

PARÂMETROS DE CONFIGURAÇÃO

A variável `c_cflag` define a configuração da linha de comunicação com o terminal. Os parâmetros são os seguintes:

CBAUD

Configura a velocidade de transmissão de bits na linha, nos dois sentidos de transmissão, em bits por segundo. O valor nulo é uma exceção, pois equivale a um comando de desconexão. Para esta versão da interface, o valor 15 correspondente a 38400 bps não está disponível, assumindo o valor 14 correspondente a 19200 bps. Os outros valores de CBAUD e as velocidades associadas são:

1:	50	bps
2:	75	bps
3:	110	bps
4:	134	bps
5:	150	bps
6:	200	bps
7:	300	bps
8:	600	bps
9:	1200	bps
10:	1800	bps
11:	2400	bps
12:	4800	bps
13:	9600	bps
14:	19200	bps

CSIZE

Indica o número de bits em que são representados os caracteres na linha.

- 0: 5 bits
- 1: 6 bits
- 2: 7 bits
- 3: 8 bits

CSTOPB

Se ligado, indica que o sinal **stop** deve ter a duração de transmissão de 2 bits. Se desligado, deve durar um tempo igual ao tempo de transmissão de 1 bit.

CREAD

Se CREAD ligado, deve ser habilitada a recepção na linha, o que significa que o terminal ligado à interface é capaz de transmitir dados. Para um terminal do tipo teleimpressora, por exemplo, CREAD deve ficar desligado, desde que o controle de fluxo xon/xoff esteja não-ativo.

PARENB

Se PARENB ligado, será acrescentado um bit de paridade aos bits de informação de cada caractere. Se PARENB desligado, apenas o número de bits definido em CSIZE será transmitido.

PARODD

Sõ tem sentido se PARENB ligado. Se PARODD ligado, o bit de paridade será ligado, se o número de bits de informação ligados for par. Se PARODD desligado, o bit de paridade será ligado, se o número de bits de informação for impar.

HUPCL

Se HUPCL ligado, deve ser cancelada a conexão quando o terminal for liberado pelo último processo usuário.

CLOCAL

Se CLOCAL ligado, é suposto que o terminal está ligado localmente, sem modem. Se CLOCAL desligado, ou seja, está sendo usada linha remota. Quando o terminal for desligado, é gerado um sinal SIGHUP, e todos os comandos pendentes e seguintes, são respondidos com a condição "fim de arquivo" (zero caracteres lidos ou escritos) até o próximo comando CLOSE, quando então a condição de "fim de arquivo" será desativada, e os comandos subseqüentes voltam a ser tratados normalmente.

DISCIPLINA DE LINHA

O parâmetro `c_line` define a funcionalidade apresentada na variável `c_lflag`. Trata-se de um artifício para que funções dedicadas, a uma determinada aplicação sejam acrescentadas, a partir de funções básicas, definidas nos subitens anteriores. Apenas as funções associadas à disciplina de linha 0 são padronizadas. Os projetistas de sistemas têm toda liberdade para utilizar os bits da variável `c_lflag` para outros valores de `c_line`.

A variável `c_lflag` define funções que constituem uma camada de software de nível mais alto, apoiada sobre as funções definidas em `c_lflag`, `c_oflag` e `c_cflag`. Os campos da variável `c_lflag` para a disciplina de linha 0 são:

ISIG

Se ligado, habilita o reconhecimento dos sinais SIGINT e SIGKILL. Então, todo caractere recebido do terminal, depois de eventualmente tratado conforme os parâmetros de entrada (`c_lflag`), é testado contra os caracteres configurados para geração de sinais. Se ISIG desligado, não há possibilidade de geração de sinais, a partir do teclado.

ICANON

Se ICANON ligado, habilita o modo canônico. Então, para qualquer caractere e, depois de aplicar os tratamentos conforme os parâmetros de entrada (c iflag), verifica se o caractere resultante é um terminador de entrada (NL, EOL ou EOF) ou um caractere de edição (ERASE ou KILL), e realiza a função correspondente em caso positivo. Se ICANON desligado, habilita o modo **raw**, no qual não há terminadores. As entradas são terminadas por tamanho ou fim de temporização (ver caracteres de controle - MIN e TIME).

XCASE

Só tem sentido se ICANON ligado. Se XCASE ligado, habilita a conversão, após a aplicação dos tratamentos dos parâmetros de entrada (c iflag) aos caracteres recebidos, ou antes da aplicação dos tratamentos dos parâmetros de saída (c oflag) aos caracteres destinados ao terminal. Em terminais antigos é comum a impossibilidade de geração dos seguintes caracteres especiais: acento grave, barra vertical, til, abre-chave e fecha-chave. A conversão permite a geração destes caracteres, a partir de outros caracteres especiais, precedidos pelo caractere barra invertida. Para que este possa ser gerado, é preciso precedê-lo de si mesmo. Ver tabela de conversão na página seguinte.

SEQUENCIA DIGITADA	CARÁCTERE ASSUMIDO
\ -	-
\ !	!
\ ~	~
\ ({
\)	}
\ \	\
\ x	X

Onde X é a maiúscula de x

ECHO

Se ECHO ligado, os caracteres recebidos do terminal são ecoados, se houver entrada corrente, ou memorizados para serem ecoados no momento em que for iniciada uma entrada. Se ECHO desligado, não há eco de caracteres.

ECHOE

Só tem significado se ICANON ligado. Se ECHOE desligado, o caractere associado à função ERASE é ecoado sem transformações, além daquelas definidas nos parâmetros de entrada (c_1flag). O parâmetro ECHOE deve ser ativado apenas para terminais de vídeo. Se ECHOE e ECHO ligados, o caractere ERASE não é ecoado e no seu lugar é ecoada a sequência BS-SP-BS, que apaga da tela o último caractere digitado. Se o último caractere digitado, antes do ERASE, foi TAB, é ecoada uma sequência de caracteres BS, em número apropriado, para que o cursor volte à posição em que estava, quando foi digitado o caractere TAB.

Se ECHOE ligado e ECHO desligado, a sequência que ecoa o caractere ERASE é SP-BS. Esta regra destina-se a terminais com eco local. O caractere ERASE deve ser configurado como BS, que é ecoado localmente. Então, apenas é complementada a sequência que apaga da tela o último caractere digitado. Se o último digitado antes do ERASE foi o caractere TAB, também deve ser descontado da sequência de caracteres BS aquele que já foi ecoado localmente.

ECHOK

Só tem sentido se ICANON ligado. Se ECHOK ligado, o caractere KILL não é ecoado, e no seu lugar é ecoado NL. Se ECHOK desligado, o caractere KILL não é substituído por NL no tratamento do eco.

ECHONL

Só tem sentido se ICANON ligado. Se ECHONL ligado, o terminador NL é ecoado mesmo se ECHO desligado. Este parâmetro também se destina a terminais dotados de eco local. Suponhamos um terminal com eco local, onde a tecla **ENTER** gera CR. Para que o operador possa terminar uma entrada com a tecla **ENTER**, é preciso que o parâmetro ICRNL esteja ligado. Quando o operador digita **ENTER**, a entrada é considerada terminada, mas localmente foi ecoado CR e o cursor voltou ao início da linha. Daí a necessidade de se ecoar NL para que o cursor avance para a linha seguinte.

NOFLSH

Só tem sentido se ISIG ligado. Se NOFLSH ligado, os caracteres das filas de entrada e saída não são descartados quando ocorrem os sinais SIGINT e SIGQUIT. Tem interesse para aplicações que capturam estes sinais e não são canceladas. Se NOFLSH desligado, ambas as filas são limpas quando ocorre os sinais SIGINT e SIGQUIT.

CARACTERES DE CONTROLE

O vetor `c_cc []` contém os caracteres configuráveis da interface com o usuário. Vimos que as funções podem ser individualmente inibidas pelos parâmetros `passa` dos pela variável `c_lflag`. As aplicações dispõem de uma segunda maneira de inibir uma função. Basta associar a esta função um caractere que não pode ser gerado pelo terminal, ou que não pode ser recebido. Neste último caso está o caractere `0xFF`, se os caracteres têm 7 bits ou menos (CSIZE) ou se estão sendo sistematicamente convertidos para 7 bits (ISTRIP ligado).

As posições dentro do vetor são as seguintes:

- 0 - caractere INTR
- 1 - caractere QUIT
- 2 - caractere ERASE
- 3 - caractere KILL
- 4 - caractere EOF ou MIN
- 5 - caractere EOL ou TIME
- 6 - reservado
- 7 - reservado

As duas últimas posições são reservadas. Se `ICANON` ligado, as posições 4 e 5 do vetor contém os caracteres configurados como terminadores EOF e EOL.

Se `ICANON` desligado (modo `raw` ativo), as posições 4 e 5 do vetor contém dois valores, `MIN` e `TIME`, que selecionam o tipo das entradas seguintes.

Se $MIN=0$ e $TIME=0$, as leituras serão imediatamente respondidas com os caracteres já digitados no momento da leitura. Se nenhum caractere tiver sido digitado antes da leitura, ela será respondida com tamanho lido nulo.

Se $MIN=0$ e $TIME>0$, a entrada é terminada quando é digitado um caractere, ou quando expira uma temporização de $TIME$ décimos de segundo.

Se $MIN>0$ e $TIME=0$, as entradas serão de tamanho fixo igual a MIN caracteres.

Se $MIN>0$ e $TIME>0$, a entrada termina quando é recebido MIN caracteres ou quando um intervalo de $TIME$ décimos de segundos se expira, sem que tenha sido recebido o caractere seguinte de uma rajada. A temporização se inicia quando o primeiro caractere é recebido e é reiniciada a cada caractere recebido.

Nome

tty - Terminal de controle da sessão.

Descrição

Este arquivo representa, para cada processo, o terminal de controle associado ao grupo de processo ao qual o processo em questão pertence, se houver. O terminal de controle de um grupo de processos é o terminal responsável pela geração dos sinais **SIGQUIT**, **SIGUP** e **SIGINT** para os processos do grupo. O primeiro terminal (arquivo especial correspondente a um terminal) aberto por um processo líder de grupo, que não seja terminal de controle de nenhum outro grupo, torna-se o terminal de controle do processo. O terminal de controle é herdado pelo filho, durante uma chamada **fork**, e mantido durante uma chamada **exec**. Um processo pode se dissociar de um terminal de controle, trocando o grupo ao qual ele pertence, através da chamada **setpgrp**. Um terminal de controle fica associado a um grupo até que o processo líder do grupo morra.

Arquivos

/dev/tty

Nome

utmp, wtmp - Registro de usuários do sistema.

Sintaxe

```
#include <sys/types.h>
#include <utmp.h>
```

Descrição

Estes arquivos contêm informações sobre os usuários do sistema. O arquivo **utmp** mantém informações sobre os usuários criados pelo processo **login** que estão no sistema num determinado momento, enquanto o **wtmp** mantém um histórico sobre todos os usuários. O arquivo é composto de entradas de formato fixo, dado pela seguinte estrutura:

```
#define    UTMP_FILE    "/etc/utmp"
#define    WTMP_FILE    "/etc/wtmp"
#define    ut_name      ut_user

struct exit_status
{
    short e_termination; /* status de termino */
    short e_exit;        /* status do exit */
};

struct utmp
{
    char    ut_user[8];   /* nome da sessão */
    char    ut_id[4];     /* id da linha em */
                          /* /etc/inittab */
    char    ut_line[12]; /* nome do disp. */
    short   ut_pid;      /* id. do processo */
    short   ut_type;     /* tipo de entrada */
}
```

```

    struct exit_status ut_exit; /* Status exit      */
                                /* de um proc. assinalado */
                                /* DEAD_PROCESS      */
    time_t   ut_time; /* tempo criação ent. */
};

/*      Definições para ut_type      */
#define EMPTY      0
#define RUN_LVL    1
#define BOOT_TIME  2
#define OLD_TIME   3
#define NEW_TIME   4
#define INIT_PROCESS 5
/* Processo produzido por init */
#define LOGIN_PROCESS 6
/* Um proc. getty esperando login */
#define USER_PROCESS 7
/* Um proc. do usuário */
#define DEAD_PROCESS 8
#define ACCOUNTING  9
#define UTMATYPE    ACCOUNTING
/* Maior valor legal para ut_type */

/* Cadeias ou formatos especiais usados no campo */
/* ut_line quando ut_type não relacionado a      */
/* processos.                                    */
/* Nenhuma cadeia para o campo ut_line podera ter */
/* mais que 11 caracteres + um NULL no seu      */
/* tamanho.                                       */

#define RUNLVL_MSG    "nivel oper %c"
#define BOOT_MSG     "carga sist"
#define OTIME_MSG    "data ant"
#define NTIME_MSG    "data corrig"

Arquivos
    /etc/utmp
    /etc/wtmp

```

Nome

volume - Formato de um volume.

Descrição

Um volume é o meio de residência dos sistemas de arquivos do SOX. A unidade básica na qual é dividido é chamada bloco. Um bloco tem seu tamanho definido ao ser criado o volume. Na atual versão do SOX, o único tamanho de bloco possível é de 1024 octetos.

Para que o sistema possa administrar o sistema de arquivos é necessária a existência de algumas áreas de controle, que são as seguintes:

- . Reservado para o boot

Área reservada. É usada para conter o programa inicial de carga, quando necessário. Esta área tem tamanho de 1024 octetos.

- . Seção de informação

É a área que contém as informações que definem o volume. Esta área é de vital importância. Estando danificada, o volume será considerado não-válido.

. Área de BCAs

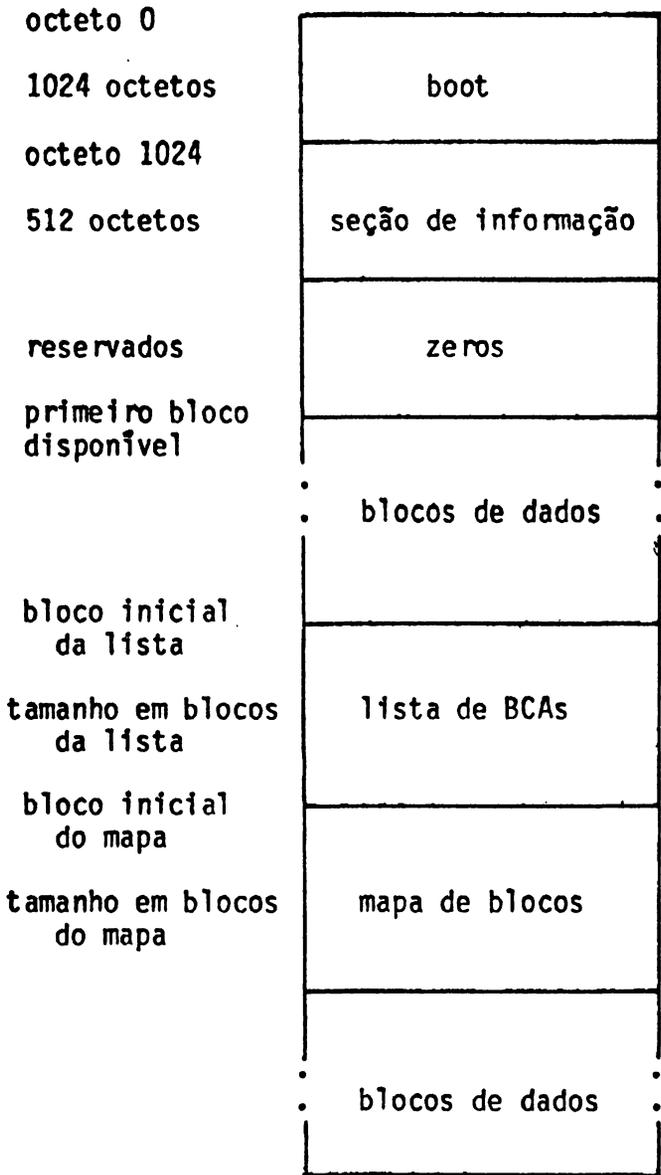
Um BCA (Bloco de Controle de Arquivo) é um descritor de um arquivo. Esta área, que é contígua, contém todos os BCAs do sistema de arquivos contido nesse volume. Os BCAs são identificados pelo seu índice nessa área (ibca - numerados à partir de 1 - sendo que o BCA de índice 1 é reservado).

. Mapa de blocos

Uma área contígua, que contém um bit para cada bloco do volume, indicando se o bloco está livre (bit=1) ou ocupado (bit=0). O primeiro bit dessa área corresponde ao bloco 0, o segundo ao bloco 1 e assim por diante.

. Blocos de dados

Blocos que serão usados pelos diversos arquivos componentes do sistema de arquivos.



O primeiro bloco disponível corresponde ao menor múltiplo de bloco que contenha 1536 (1,5 K) octetos. Os blocos de 0 a este número são reservados. Os blocos iniciais e o tamanho das diversas áreas de controle do volume estão contidos na seção de informação.

Página intencionalmente em branco.

ANEXO C

DISCOS LÓGICOS E PARTIÇÕES

Este anexo contém a descrição dos discos lógicos e partições que devem ser conhecidos pelo administrador do SOX.

A apresentação é feita em três partes: discos lógicos e partições, formatação e criação de disco lógico.

Discos Lógicos e Partições

No SOX, um (volume de) disco físico é uma unidade de meio magnético de acesso direto, que pode ser associada a uma unidade física de disco. Um disco físico é dividido em cilindros, numerados de zero até o último cilindro do disco. Um cilindro é dividido em trilhas (uma por superfície de dados), numeradas de zero em diante, e cada trilha é composta de blocos físicos ou setores, numerados a partir de um. Assim, o disco físico abrange todo o espaço de dados endereçável do meio magnético.

O usuário lida normalmente com discos lógicos, e não com discos físicos - por exemplo: sistemas de arquivos são mantidos em discos lógicos. O disco lógico fornece uma visão uniforme dos diferentes discos existentes no sistema, permitindo que o usuário se abstraia de certas características dos discos físicos.

Em um disco flexível não há diferença entre disco lógico e disco físico, isto é, existe apenas uma maneira do usuário "enxergar" um disco flexível. Para os discos rígidos, entretanto, o Administrador do Sistema pode criar um ou mais discos lógicos que forneçam "visões" parciais do disco físico, ou que permitam utilizar volumes fisicamente distintos, como se fossem um só volume de disco.

Um disco lógico é composto de uma ou mais partições de dados, cada uma das quais é formada por um conjunto consecutivo de blocos de um dispositivo físico diferente. A unidade básica de utilização do disco lógico é o bloco (lógico), cujo tamanho é fixado em 1 kocteto por disco lógico.

A numeração (endereço) dos blocos de um disco lógico começa de zero e vai até o maior bloco deste disco, independente do número e da localização das partições que o compõem.

Um disco físico pode ter no máximo 16 partições, as sim definidas:

Partição	Uso
0-9	Partições de usuário, definidas pelo Administrador
10-12	Reservadas
13	Partição de manutenção
14	Partição de identificação
15	Disco físico

O número da partição identifica para o SOX a partição desejada. As partições 0 a 9 são destinadas a sistemas de arquivos SOX, podendo ser definidas através do comando **iv** (ver "Manual de Referência Comandos SOX"). As partições 10 a 12 são reservadas para uso futuro. A partição 13 ocupa o último cilindro do disco Winchester, e não é usada pelo SOX, sendo destinada à manutenção do próprio disco, por exemplo: como área de teste para programas de diagnóstico. A partição 14 ocupa o primeiro cilindro (cilindro 0) do disco Winchester, e contém informações sobre as demais partições do disco. As partições 13 e 14 são criadas automaticamente após o término com sucesso da formatação do disco físico. A partição 15 endereça o disco físico inteiro, englobando todas as demais partições, portanto ela deve ser utilizada com cautela.

A partição de identificação de um disco físico contém:

- Tabela de identificação das partições, com a localização (cilindro inicial), tamanho e disco físico e que pertence cada uma das partições do disco físico;
- Lista de defeitos do meio, contendo os endereços absolutos (cilindros, superfície e setor) de cada bloco defeituoso do disco físico;
- Diário de falhas de disco, onde são guardadas pelo SOX as ocorrências de erros e falhas nas operações com o disco. Estas informações podem ser recuperadas através de uma chamada `ioctl`.

Uma partição de usuário é identificada para o sistema através de um número que associa à unidade física de disco onde reside. Este número deve ser igual ao número da partição (0-9) mais uma base, que é igual a 64 (decimal) vezes o número da unidade física. Por exemplo, no X-10 as partições de usuário são identificadas pelos seguintes números:

unidade física	base	identificação das partições
0	0	0 a 9
1	64	64 a73
2	128	128 a137
3	192	192 a201

Formatação

A formatação é uma operação sobre o disco físico e consiste da escrita em todas as suas trilhas de informações que permitam o endereçamento de blocos físicos no meio. Todo o conteúdo prévio do disco, incluindo suas partições e discos lógicos, é destruído pela formatação.

A formatação bem sucedida de um disco Winchester deixa criadas no mesmo as partições de identificação e de manutenção. Se estas partições não puderem ser criadas, a formatação termina com erro e o disco é rejeitado pelo SOX.

Durante a formatação de um disco Winchester, o SOX monta uma lista de todos os setores que foram achados defeituosos no disco físico. Se a formatação for feita pelo comando **iv**, no final da operação esta lista é combinada com a que tiver sido eventualmente fornecida pelo usuário (através do parâmetro **defeitos**) e gravada na partição de identificação do disco.

Criação de Disco Lógico

Para criar um disco lógico, o usuário precisa fornecer o tamanho e a localização (em termos de unidade física) de cada partição de dados que irá compor esse disco. O comando **iv** utiliza o parâmetro **partições** para isso; esse parâmetro especifica uma lista

tamanho [[unidade tamanho] ...]

onde o primeiro valor é o tamanho da primeira partição (que reside na unidade associada ao parâmetro **lógico**) e seguem-se pares de valores, definindo o número da unidade física e o tamanho da respectiva partição. O SOX aloca partições de dados automaticamente em cada unidade para as partições definidas por esse parâmetro.

Os tamanhos são fornecidos em unidade de bloco lógico. O tamanho de cada partição é automaticamente acrescido de espaço para setores alternativos e é arredondado para a fronteira de cilindro mais próxima.

A existência de setores alternativos é normalmente transparente para o usuário. O SOX aloca um setor alternativo para cada setor defeituoso da partição, de modo que todo acesso ao setor defeituoso se transforma em um acesso ao setor alternativo. Se o disco lógico for criado pelo comando **iv**, o espaço solicitado pelo usuário será acrescido de um bloco para cada bloco defeituoso existente na partição e de uma reserva de pelo menos 10 blocos alternativa para futuros defeitos.

Nos discos flexíveis todo o espaço de dados do meio físico é utilizável para um sistema de arquivos, isto é, o volume físico é igual ao volume lógico. Nos discos rígidos, o Administrado deve criar um ou mais discos lógicos e associá-los a arquivos especiais para que possam ser utilizados. A associação é feita através do subtipo do dispositivo, armazenado no BCA do arquivo especial, que deve ser igual ao número de identificação da primeira partição do disco lógico.

O arquivo especial cujo subtipo de dispositivo módulo 16, for igual a 15 (isto é, cujo subtipo possuir os 4 bits menos significativos iguais a um), permite o acesso a todo disco físico. Por convenção, esse arquivo possui o nome `rdwcn`, onde "n" é igual ao número da unidade física. Os discos lógicos são convencionalmente denominados `rdwn`, onde `r` é usado para os arquivos do tipo caractere (modo `raw`) e "n" é igual ao número de identificação da partição de dados inicial do disco lógico.

Página intencionalmente em branco.

Bit Indicador

Qualquer bit, dígito, caractere ou sinal usado para indicar a ocorrência de algum estado. Em inglês, "flag".

Buffer pool

Conjunto de áreas de memória utilizadas para armazenamento temporário de informações nas operações entrada e saída, funcionando como uma extensão dos dispositivos associados.

Fim de temporização

Período de tempo que o sistema ou programa aguarda alguma resposta ou intervenção do usuário, antes de prosseguir com sua execução. Em inglês, "time-out".

Erro de framing

Ocorre erro de "framing" na recepção de um caractere, quando o sinal de fim de transmissão do mesmo não chega no momento esperado.

Programa autônomo

Programa que opera de forma independente. Em inglês, "stand-alone".

Página intencionalmente em branco.

ISR-52-850/87
UP APT PRES. VARGAS
DR-RJ

CARTA RESPOSTA COMERCIAL

NÃO É NECESSÁRIO SELAR

O SELO SERÁ PAGO POR

COBRA COMPUTADORES E SISTEMAS BRASILEIROS S/A
(DIV. ASSIST. SOFTWARE - DAS)

20299 RIO DE JANEIRO - RJ

--	--	--	--	--

CEP

Endereço:

Remetente:

COLE AQUI A ÁREA DO CARTÃO



COMPUTADORES E SISTEMAS BRASILEIROS S.A.

CARTÃO DE CADASTRO E COMENTÁRIOS DO USUÁRIO

Administrador do Sistema SOX

SX0389-03.0
abril/1989

Prezado leitor:

Suas críticas, comentários e sugestões são valiosos instrumentos para que a COBRA possa, cada vez mais, aprimorar a documentação de seus produtos. Solicitamos, portanto, sua colaboração em preencher este cartão, colocando-o em qualquer caixa de correio (porte pago). Obrigado.

AVALIAÇÃO DO MANUAL

Nos itens que se seguem, indique o conceito aplicável, comentando se assim o desejar:

CONCEITOS: ● Excelente (E) ● Bom (B) ● Regular (R) ● Precisa melhorar (P) ● Insuficiente/Inadequado (I)

ITEM	CONCEITO
APRESENTAÇÃO DO MANUAL (Encadernação, acabamento, formato)	
QUALIDADE GRÁFICA (Tipo de impressão, diagramação, papel utilizado)	
CLAREZA DAS INFORMAÇÕES (Linguagem empregada, facilidade de compreensão do texto)	
ABORDAGEM DOS ASSUNTOS (Organização, precisão e objetividade)	

ITEM	CONCEITO
DESENHOS, ILUSTRAÇÕES E EXEMPLOS (Suficiência, adequabilidade e clareza)	
UTILIDADE (Adequação à utilização pretendida, facilidade de consulta)	
EDITORAÇÃO E REVISÃO (Existência de falhas de impressão, paginação, erros ortográficos ou gramaticais)	
QUALIDADE GERAL DO MANUAL (Avaliação global do usuário)	

COMENTÁRIOS E SUGESTÕES

IDENTIFICAÇÃO DO USUÁRIO

Nome:	Profissão:
Empresa:	Cargo:
Endereço p/correspondência:	CEP:
	Tel.:

Av. Comandante Guarany, 447 - 22786 - Rio de Janeiro - RJ
(021) 542-9393 - Telex (021) 22420

COLE ESTA ABA NO VERSO DO CARTÃO

cobrã

Computadores e Sistemas Brasileiros S.A.