

ORGANIZAÇÃO DO SISTEMA OPERACIONAL SOMBRA

R. S. DE OLIVEIRA *, C. M. DA COSTA **

SUMARIO

Este trabalho apresenta o sistema operacional multiusuário SOMBRA. Este sistema incorpora uma série de características presentes em outros sistemas operacionais modernos, tais como: linguagem de comando com construções de alto nível, sistema de arquivos hierárquico organizado em forma de árvore, possibilidade da execução de programas em background e spooling de saída para impressoras.

ABSTRACT

This paper describes a multi-user operating system called SOMBRA. This system exhibits many features which are found in other modern operating systems, such as: a command language with high level constructs, a hierarchical file system organized as a tree, background execution of programs and output spooling to the printers.

* Engenheiro Eletricista (PUCRS, 1983). Mestre em Ciência da Computação (UFRGS, 1987); áreas de interesse: programação concorrente e sistemas operacionais; Curso de Pós-Graduação em Ciência da Computação da UFRGS; Av. Osvaldo Aranha, 99 - Porto Alegre - RS - CEP 90000.

** Analista de Sistemas (PUCRS, 1978). Mestre em Ciência da Computação (UFRGS, 1985); áreas de interesse: programação concorrente e sistemas operacionais; Professor Adjunto do Departamento de Informática/Curso de Pós-Graduação em Ciência da Computação da UFRGS; Av. Osvaldo Aranha, 99 - Porto Alegre - RS - CEP 90000.

INTRODUÇÃO

O Sistema Pascal Concorrente /HAN 77/ cria um ambiente de programação concorrente de alto nível, onde é possível implementar sistemas operacionais que podem ser transportados para outras máquinas com relativa facilidade. O Curso de Pós-Graduação em Ciência da Computação (CPGCC) da UFRGS dispõe do Sistema Pascal Concorrente implementado em um minicomputador LABO 8034 /MED 81/.

O sistema operacional SOM /BRA 84/ foi especificado para criar um ambiente interativo, multiusuário, tendo como base o Sistema Pascal Concorrente. Entretanto, não foi implementado.

Entre os projetos em andamento no CPGCC está a construção de uma máquina multiprocessadora, baseada em dois microprocessadores MC68000 da Motorola. Atualmente, um protótipo desta máquina, conhecida como máquina SUMUS, está em fase de depuração.

Como parte deste mesmo projeto, foi prevista a implementação do Sistema Pascal Concorrente na máquina SUMUS e a implementação de um sistema operacional multiusuário para este ambiente, tendo como ponto de partida a especificação do sistema SOM. O projeto ainda prevê que este sistema operacional deve ser implementado no LABO 8034 e posteriormente transportado para a máquina SUMUS.

O presente trabalho descreve a especificação e implementação do SOMBRA, um sistema operacional multiusuário, escrito em Pascal Concorrente no LABO 8034. Este sistema apresenta diversas alterações em relação a especificação do SOM.

O SOMBRA foi criado para satisfazer um dos objetivos do projeto descrito. Para tanto, ele deverá ser transportado para a máquina SUMUS, tão logo o Sistema Pascal Concorrente esteja lá disponível.

Devido às restrições impostas pela pequena memória principal do LABO 8034 (se comparada com a da máquina multiprocessadora), não é possível executar o SOMBRA completo e ainda dispor de um espaço razoável para a execução dos programas de usuário. Isto nos leva à execuções parciais do SOMBRA no LABO 8034, onde componentes específicos são testados separadamente. Testes de conjunto são efetuados na medida em que o tamanho da memória permite.

Abaixo estão enumeradas algumas características do sistema operacional:

- Multiusuário;
- Linguagem de Comando com construções de alto nível;
- Sistema de Arquivos organizado em árvore;
- Execução de programas em background;
- Spooling de saída para impressoras;

ORGANIZAÇÃO DO SOMBRA

Esta seção mostra uma visão geral do sistema, através da apresentação de suas características gerais, e de sua organização. Os diversos componentes do SOMBRA são descritos funcionalmente.

De uma forma geral, o SOMBRA pode ser dividido em três grandes partes:

- Os processos envelope;
- Os programas seqüenciais;
- Os subsistemas do programa concorrente.

PROCESSO ENVELOPE

Processo envelope é o elemento ativo do sistema que executa as tarefas requisitadas pelo usuário. Cada terminal é atendido por um processo envelope. O que este processo faz essencialmente é carregar e executar programas seqüenciais, e atender as chamadas de sistema destes programas.

A primeira tarefa do processo envelope é efetuar a conexão de um usuário. Isto é feito através da carga e execução de um programa seqüencial adequado. Uma vez tendo sido reconhecido pelo sistema, o usuário poderá dar início a sua sessão de trabalho. O processo envelope então carrega e executa o interpretador da Linguagem de Comando.

A Linguagem de Comando é a interface entre o usuário e o sistema. É através dela que o usuário determina as tarefas a serem realizadas pelo sistema. A linguagem pode ser usada, por exemplo, para listar o conteúdo de um diretório, para listar os usuários correntemente conectados ao sistema, para executar um programa seqüencial específico. O interpretador da linguagem, ele próprio um programa seqüencial, executa as tarefas determinadas pelo usuário através de chamadas de sistema ou através da execução de um outro programa seqüencial em overlay.

As chamadas de sistema oferecidas pelo SOMBRA estão listadas abaixo, juntamente com uma descrição sumária de cada uma delas.

OPEN - Abre arquivo para acesso, inclusive arquivos de impressão;

CLOSE - Fecha arquivo previamente aberto, inclusive arquivo de impressão;

WRITE - Escreve em um arquivo previamente aberto, inclusive arquivo de impressão. O acesso é feito a registro, cujo tamanho é definido na abertura e pode variar entre 2 e 512 bytes;

READ - Lê de arquivo previamente aberto. O acesso é feito a registro, cujo tamanho é definido na abertura e pode variar entre 2 e 512 bytes;

GETCHAR - Lê de arquivo previamente aberto. O acesso é feito a caractere;

PUTCHAR - Escreve em arquivo previamente aberto, inclusive arquivo de impressão. O acesso é feito a caractere;

RUN - Executa programa seqüencial através de overlay;

MARK - Determina a posição atual da área de heap do programa;

RELEASE - Libera área de heap do programa;

GETTIME - Obtém o dia e hora atuais;

ACCEPT - Lê caractere do teclado;

DISPLAY - Coloca caractere na tela;

IOTRANSFER - Reservada para uso futuro;

RUNJOB - Inicia execução em background de um programa seqüencial;

SENDMSG - Envia mensagem para outro terminal;

GETTERM - Obtém o estado de todos os terminais do sistema e dos processos envelope destinados à execução em background;

GETDISK - Obtém o estado de todas as unidades de disco do sistema;

GETUSER - Obtém o número do terminal e a identificação do usuário conectado;

MAKEDIR - Cria diretório;

DELETE - Remove arquivo, diretório ou ligação;

RENAME - Troca o nome de arquivo, diretório ou ligação;

LISDIR - Obtém o conteúdo de um diretório;

FILESTATUS - Obtém ou altera características de um arquivo;

CDIR - Posiciona o diretório corrente;

READCD - Obtém a identificação do diretório corrente;

LINK - Estabelece ligação para arquivo ou diretório;
MOUNT - Monta volume em unidade de disco;
UNMOUNT - Desmonta volume;
BLOCK - Bloqueia um outro terminal;
UNBLOCK - Libera terminal para uso;
WILLBLOCK - Determina se o terminal será ou não bloqueado;
BLOCKMYSELF - Bloqueia o próprio terminal;
SETREADY - Comunica conexão de usuário;
STOPPGM - Aborta execução de programa seqüencial em outro terminal;
SETDATE - Acerta a data no relógio do sistema;
SETCLOCK - Acerta a hora no relógio do sistema.

Uma vez terminada a conexão, através de comando adequado, dado pelo usuário ao interpretador, o processo envelope executa uma limpeza em suas estruturas de dados e passa a esperar uma nova conexão, sendo este o seu ciclo de trabalho ao longo da existência do sistema.

PROGRAMAS SEQUENCIAIS

Um programa seqüencial é carregado e executado pelo processo envelope, dispondo de determinadas chamadas de sistema para realizar sua tarefa.

~~Além dos programas seqüenciais responsáveis pelo reconhecimento do usuário e pela interpretação da Linguagem de Comando, são necessários utilitários, tais como editor de texto, compiladores, programa para cópia de arquivos, programa para impressão de arquivos, e assim por diante.~~

SUBSISTEMAS DO PROGRAMA CONCORRENTE

O processo envelope, para executar as chamadas de sistema requeridas pelos programas seqüenciais, pode utilizar os serviços dos demais subsistemas do programa concorrente. Estes subsistemas geram o compartilhamento de recursos entre os diversos processos envelope.

Os sete subsistemas existentes são responsáveis pela gerência das seguintes áreas:

- Controle do tempo;
- Controle de volumes;
- Controle de terminais;

- Troca de mensagens;
- Sistema de Arquivos;
- Impressão;
- Execução subterrânea.

O sistema operacional mantém um relógio interno, atualizado a cada segundo. É possível, através das chamadas de sistema apropriadas, consultar o relógio, além de acertar dia e hora. O subsistema de controle do tempo é responsável por sua atualização, além de utilizá-lo no controle do tempo de execução de programas seqüenciais que podem, opcionalmente, ter um limite de tempo de execução, após o qual são abortados.

O subsistema de controle de volumes mantém tabelas sobre o estado de cada unidade de disco. Informações como quais unidades de disco estão ativas, qual a identificação do volume montado, entre outras, são mantidas nestas tabelas. Toda vez que uma unidade de disco é ativada ou desativada, estas tabelas são atualizadas.

O subsistema de controle de terminais mantém tabelas sobre o estado de cada terminal. Informações como que usuários estão conectados, em que terminais, entre outras, são mantidas nestas tabelas. Toda vez que um usuário inicia ou termina uma sessão de trabalho, estas tabelas são atualizadas.

Dentro do sistema operacional existe o subsistema que fornece o serviço de troca de mensagens entre usuários correntemente conectados. Este serviço permite que usuários enviem e recebam pequenas mensagens através do próprio sistema.

O Sistema de Arquivos implementa uma estrutura de diretórios em árvore, permitindo facilidades do tipo link e diretório corrente /PET 85/. Ele é responsável pela gerência de espaço em disco e pela criação, remoção e alteração de diretórios e arquivos de dados. O Sistema de Arquivos também é responsável pelo controle do acesso á arquivos, bem como pelo método de acesso aos dados de um arquivo, inclusive arquivos compartilhados entre usuários.

O subsistema de impressão realiza as funções de um output spooler /DEI 84/. O texto a ser impresso é colocado em disco pelo processo envelope e

posteriormente enviado para a impressora por este subsistema. Este mecanismo permite que diversos programas seqüenciais realizem suas impressões simultaneamente, mesmo que o sistema disponha de uma única impressora.

Existe, como recurso do sistema, um conjunto de processos envelopes que não estão vinculados a nenhum terminal. Um programa seqüencial pode, através da chamada de sistema adequada, alocar um destes processos e determinar que programa seqüencial ele deverá executar. Este tipo de execução é denominada execução em background. O sistema operacional não implementa nenhuma forma de comunicação ou sincronização entre processos envelopes.

O SISTEMA DE ARQUIVOS

O Sistema de Arquivos do SOMBRA está organizado na forma de uma árvore. Os nodos da árvore correspondem a arquivos e diretórios do sistema. Os diretórios contêm nomes e endereços de outros arquivos ou diretórios. Arquivos são utilizados para armazenar dados do próprio sistema ou de usuários. As folhas da árvore do Sistema de Arquivos correspondem a arquivos ou diretórios vazios. Os nodos intermediários da árvore são necessariamente diretórios.

Cada arquivo ou diretório possui um nome e uma identificação. O nome pode ter de 1 a 8 caracteres. O nome serve para identificar o arquivo ou diretório dentro do seu diretório pai na árvore, ou seja, dentro do diretório onde ele aparece como entrada. Entretanto, é possível a existência de dois diretórios ou arquivos distintos, com o mesmo nome, mas localizados em diferentes diretórios. Para identificar completamente um diretório ou arquivo dentro da árvore, utiliza-se o caminho sobre a árvore que, a partir da sua raiz, leva até o diretório ou arquivo em questão. Este caminho é denominado de identificação do diretório ou arquivo. Uma notação própria é utilizada para escrever uma identificação. Os nomes dos diretórios que formam o caminho a partir da raiz, da árvore, juntamente com o nome do diretório ou arquivo considerado, são dispostos em seqüência, separados por uma barra à esquerda (\). Uma barra à esquerda deve preceder o resultado da concatenação.

O Sistema de Arquivos do SOMBRA foi projetado para tirar proveito das características da situação mais genérica. Em outras palavras, sua

estrutura foi organizada para uma instalação alvo onde estão disponíveis diversas unidades de disco, e estas unidades podem ser acessadas simultaneamente. A impossibilidade de acesso simultâneo às unidades de disco, ou a existência de uma única unidade, constituem casos particulares da estrutura proposta. Cada unidade de disco permite o acesso a um volume, que o SOMBRA supõe removível. O fato do volume ser realmente removível ou não é indiferente para o sistema.

Cada volume tem o seu conteúdo organizado como uma subárvore completa do Sistema de Arquivos. Isto significa que um diretório não pode fazer referência a outro diretório ou arquivo colocado em outro volume. Todo volume é autocontido, e desconhece a existência de outros volumes.

Como os volumes são considerados removíveis, eles podem ser montados e desmontados. Montar um volume presente em uma unidade de disco significa trazer para memória os seus descritores, o que torna o seu conteúdo acessível. Por outro lado, ao desmontar um volume, os seus descritores que estavam na memória são salvos no volume, o que impede o acesso ao seu conteúdo, até que ele seja montado novamente.

O diretório raiz da árvore do Sistema de Arquivos contém uma entrada para cada volume montado. Na verdade, ele não pode ser acessado como um diretório comum e, ao contrário dos demais diretórios, não é salvo em disco. Ele somente indica quais unidades estão ativas no momento, e qual o nome do volume montado em cada unidade. Ele não é mantido pelo Sistema de Arquivos, mas pelo Controle de Volumes.

A identificação de um arquivo pode indicar o volume ao qual pertence o arquivo. Por exemplo, o arquivo cuja identificação é mostrada abaixo deverá ser procurado no volume SOMBRA1, se ele estiver montado:

```
\SOMBRA1\ROMULO\TESE.CPI
```

A identificação de um arquivo ou diretório é formada por um caminho sobre a árvore do Sistema de Arquivos. Até agora, este caminho sempre foi descrito como iniciando na raiz da árvore, com o primeiro nome identificando o volume em questão. O mecanismo de diretório corrente permite que este caminho não inicie na raiz da árvore, mas em um diretório qualquer. O usuário, através de comando apropriado, pode transformar um diretório qualquer da árvore, em seu diretório corrente. A partir deste momento, ele pode identificar arquivos ou diretórios a

partir de caminhos que iniciam na raiz da árvore, ou em seu diretório corrente. Se a barra inicial estiver presente na identificação de um arquivo ou diretório, então o caminho inicia na raiz da árvore. Se a barra estiver ausente, então o caminho inicia no diretório corrente do usuário.

O Sistema de Arquivos do SOMERA permite que o usuário estabeleça ligações ("links") para arquivos ou diretórios. Estabelecer uma ligação significa criar em um diretório uma entrada que aponte para um arquivo ou diretório já existente. O nome associado com esta entrada pode ou não ser igual ao nome original do arquivo ou diretório apontado. As características dele continuam as mesmas, o seu dono continua o mesmo, apenas foi criado um novo caminho sobre a árvore para chegar-se até ele. Um arquivo ou diretório somente é removido da árvore quando não existir mais nenhuma ligação para ele. O Sistema de Arquivos exige que as ligações fiquem restritas a cada volume. Além disto, o diretório onde é introduzida a ligação como entrada deve estar em um nível mais baixo do que o diretório ou arquivo alvo da ligação. Isto impede a ocorrência de ciclos dentro do Sistema de Arquivos, o que dificultaria os algoritmos de procura e remoção de arquivos e diretórios.

A LINGUAGEM DE COMANDO

A Linguagem de Comando tem por objetivo ser a interface entre o usuário e o sistema operacional. Através dela o usuário requisita serviços do sistema operacional e recebe o resultado de tais serviços.

Abaixo estão algumas das construções de alto nível oferecidas pela linguagem:

- Expressões formadas através dos 16 operadores definidos;
- Construções do tipo sequência de comandos;
- Construções do tipo se-então-senão;
- Construções do tipo laço com teste no início, teste no fim ou com número previamente determinado de iterações;
- Variáveis.

Além das construções citadas, a linguagem oferece comandos com as seguintes funções:

- Obtém a data atual;

- Obtém a hora atual;
- Obtém o estado das unidades de disco;
- Obtém o estado dos terminais e processos envelope destinados a execução em background;
- Envia mensagem;
- Obtém ou altera diretório corrente;
- Lista conteúdo de diretório;
- Cria diretório;
- Remove arquivo, diretório ou ligação;
- Dispara a execução de um lote de comandos;
- Dispara a execução de um lote de comandos em background;
- Executa um programa seqüencial;
- Realiza a desconexão do usuário;
- Aborta programa seqüencial em andamento;
- Monta volume;
- Desmonta volume;
- Bloqueia terminal;
- Libera terminal;
- Acerta o relógio do sistema.

A Linguagem de Comando é implementada por um programa seqüencial, seguindo a filosofia de máquina virtual de Brinch Hansen /HAN 77/. Os comandos do usuário são traduzidos em operadores de uma máquina virtual, que posteriormente serão interpretados. Esta filosofia permitiu que métodos utilizados na construção de compiladores /AHO 77/ /CAL 79/ fossem usados na tradução dos comandos em código da máquina virtual.

CONCLUSÕES

O sistema operacional SOMBRA, descrito neste trabalho, é composto basicamente por um programa concorrente, e pelo programa seqüencial que implementa a Linguagem de Comando.

O compilador Pascal Concorrente disponível no sistema operacional SOLO não permite a compilação de módulos em separado. A pequena memória do LABO 8034 impõe sérias restrições a capacidade do compilador. Devido a estas limitações, não foi possível realizar a integração do sistema.

O interpretador da Linguagem de Comando, programado em Pascal Seqüencial, foi depurado em um IBM-PC. Para isto, foi programado um conjunto de

rotinas capaz de simular, parcialmente, as chamadas de sistema do SOMBRA. O ILC não utiliza nenhuma facilidade própria do IBM-PC, podendo ser transportado para o SOMBRA sem nenhuma alteração. Graças a modularidade do ILC, a substituição do SOMBRA por um ambiente simulado foi necessária somente para a depuração da fase de interpretação do código virtual. O programa fonte do ILC ocupa aproximadamente 3000 linhas.

Apesar de codificado, o programa concorrente teve que ser dividido em versões parciais para que pudesse ser compilado e testado. Ele ocupa aproximadamente 5500 linhas. Diversas versões parciais foram criadas, ao mesmo tempo em que foram desenvolvidos programas e rotinas para auxílio à depuração.

Finalmente, é importante salientar que o transporte dos programas concorrente e seqüencial do LABO 8034 para a máquina SUMUS não deverá apresentar problemas. O uso de linguagens de alto nível confere elevada portabilidade ao sistema. Isto se aplica a qualquer máquina onde venha a ser implementado o Sistema Pascal Concorrente.

BIBLIOGRAFIA

- /AHO 77/ AHO, Alfred & ULLMAN, Jeffrey. Principles of compiler design. New Jersey, Addison-Wesley, 1977.
- /BRA 84/ BRANDÃO, M. F. R. Especificação de um sistema operacional multiusuário em Pascal Concorrente. Porto Alegre, PGCC-UFRGS, 1984. (Dissertação de mestrado).
- /CAL 79/ CALINGAERT, Peter. Assemblers, compilers, and program translation. Potomac, Computer Science Press, 1979.
- /DEI 84/ DEITEL, H. M. An introduction to operating systems. New Jersey, Addison-Wesley, 1984.
- /HAN 77/ HANSEN, P. B. The architecture of concurrent programs. Englewood Cliffs, Prentice-Hall, 1977.
- /MED 81/ MEDEIROS, G. C. R. Implementação do sistema Pascal Concorrente no LABO 8034. Porto Alegre, PGCC-UFRGS, 1981. (Dissertação de mestrado).
- /PET 85/ PETERSON, J. & SILBERSCHATZ, A. Operating system concepts. New Jersey, Addison-Wesley, 1985.