Sistemas Operacionais

**SOLARIS**

**Introdução**

Em um sistema de computação, a destinação final do conteúdo de qualquer tipo de memória é o processador, também conhecido comoCPU (central processor unit). Isto é, o objetivo final de cada uma das memórias (ou do sistema de memória) é armazenar informações destinadas a serem, em algum momento, utilizadas pelo processador. Ele é o responsável pela execução das instruções, pela manipulação dos dados e pela produção dos resultados das operações.

SPARC (Scalable Processor ARChitecture) é uma arquitetura de processador aberta, baseada em RISC (Reduced Instruction Set Computer), inclui processadores de diversas empresas e estão presentes em dispositivos que vão desde câmeras digitais a grandes servidores UNIX. É uma arquitetura controlada e gerenciada por uma fundação internacional não governamental fundada em 1989. Sua filosofia, desde o nascimento, é de um padrão aberto para promover

inovação, flexibilidade e encorajar a competição. SPARC foi criada nos laboratórios da Sun Microsystems Inc, baseado na pioneira pesquisa da Universidade da Califórnia em Berkeley sobre arquitetura RISC. O primeiro produto baseado na arquitetura SPARC foi produzido pela Sun e Fujitso em 1986 e em 1987 a Sun lançou a primeira workstation baseada em um processador SPARC. Entre os diversos produtos da arquitetura SPARC podemos citar Computação Militar, Fotografia Digital, Computação Aeroespacial, Imagens Médicas, Computação Empresarial, Notebooks, Telecomunicações, Armazenamento Empresarial,

Computadores Pessoais, Rede, Eletro-Eletrônicos para Casa. As ações operativas do processador são realizadas nas suas unidades funcionais: na unidade aritmética e lógica-ULA (Aritmetic and Logic Unit), na unidade de ponto flutuante-UFP (Float Point Unit-FPU) ou talvez em uma unidade de processamento vetorial. No entanto, antes que a instrução seja interpretada e as unidades da CPU sejam acionadas, o processador necessita buscar a instrução de onde ela estiver armazenada (memória cache ou principal) e armazená-la em seu próprio interior, em um dispositivo de memória denominado registrador de instrução. Em seguida a este armazenamento da instrução, o processador deverá, na maioria das vezes, buscar dados da memória (cache, principal ou mesmo de unidades de disco em fita) para serem manipulados na ULA. Esses dados também precisam ser armazenados em algum local da CPU até serem efetivamente utilizados. Os resultados de um processamento (de uma soma, subtração, operação lógica, etc.) também precisam, às vezes, ser guardados temporariamente na CPU,

ou para serem novamente manipulados na ULA por uma outra instrução, ou para serem transferidos para uma memória externa à CPU. Esses dados são armazenados na CPU em pequenas unidades de memória denominadas registradores.

**Gerenciamentos de processos**

O SOLARIS trata os threads em nível de usuário e de Kernel da mesma forma e possui multiprocessamento simétrico. Inicialmente, quando se ativa o sistema, é criado o processo

0, que por sua vez cria o processo 1, conhecido como init, que é o pai de todos os processos (Adão dos processos), cada processo seja ele pai ou filho, possui seu espaço de endereçamento, evitando que um processo interfira nas variáveis de outro.

Todo processo no sistema SOLARIS é identificado pelo processo-pai, é uma espécie de batismo, existindo uma forma de conversarem entre si, por intermédio demensagens, e a saída de um processo é direcionada para a entrada do outro, formando uma espécie de cadeia de processos.

Os processos no SOLARIS possuem duas estruturas: a tabela de processos, com informações como número de processos, modo e prioridade e a estrutura de usuário, com informações como quem criou o processo, quem está usando, etc.

A política de escalonamento de processos é preemptiva, utilizando um misto de múltiplas filas, contador de programa e troca de contexto. A primitiva de sincronização é a utilização de monitores e semáforos.

**Gerenciamento de memória**

O Kernel do sistema operacional é o maior responsável pelo gerenciamento de memória no Solaris, alocar e desalocar memória para os processos quando eles precisarem, bem como gerenciar a troca entre a memória principal e o disco (swap) quando na memória principal não couber o tamanho do processo.

O escalonamento de memória define as prioridades dos processos, levando-se em consideração o tempo de execução acumulado. Os processos que passaram mais tempo em execução terão menos prioridade que os processos que ainda não foram executados.

Existe uma técnica que seleciona alguns processos da memória a fim de liberar espaço, chamada swapping, bem como, o SOLARIS usa paginação por demanda, evitando o carregando do processo completo para a memória principal, de tempos em tempos o SO percorre as páginas do processo, procurando o bits de acesso, se for igual a zero, é desprezado, se for igual a um, é carregado na memória

**Gerenciamento de entrada e saída**

Todos os requerimentos de entrada e saída são trabalhados sincronicamente, ou seja, um processo que solicita uma entrada, por exemplo, é suspenso a partir do momento dessa solicitação e liberado quando a entrada tiver sido completada. A gerência de entrada e saída no Solaris é implementada por drivers, sendo necessário um driver para cada dispositivo. Esses drivers são acoplados ao sistema operacional e, uma vez acrescentado um novo dispositivo, um driver correspondente será acoplado ao kernel.

O Solaris trabalha com dois tipos de drivers de entrada e saída:

* driver de bloco, onde a transmissão é feita por blocos e normalmente está associada a dispositivos com altas taxas de transferência entre esse dispositivo e a memória;
* e driver de terminal, cuja transmissão é feita caractere por caractere e é usado em dispositivos mais lentos.

No caso do driver de bloco, sempre que um processo solicita uma transferência, o kernel verifica se o bloco já está na memória ou não e, em seguida, o sistema transfere o bloco solicitado para o dispositivo de entrada e saída. Blocos frequentemente utilizados tendem a permanecer na memória, reduzindo, portanto, o tráfego de entrada e saída.

O driver de terminal é utilizado por todos os dispositivos que não se ajustam ao modelo de blocos. Contudo, a maioria dos dispositivos que possuem a interface estruturada para o driver de bloco, também possui a interface de terminal.

O acesso aos dispositivos de entrada e saída é integrado ao sistema de arquivos através de arquivos especiais. Esses arquivos podem ser acessados da mesma forma que qualquer outro arquivo, utilizando primitivas de leitura e gravação.

**Sistemas de arquivo**

O sistema de arquivos do Solaris é baseado em uma estrutura de diretórios em árvore, não existindo dependência entre a estrutura lógica desses diretórios e o local onde os arquivos estão fisicamente armazenados. Esse modelo permite que uma estrutura seja formada por diferentes discos, inclusive em estações remotas.

Utilizando uma arquitetura denominada Virtual File System (VFS), o Solaris proporciona uma interface padrão para diferentes tipos de sistemas de arquivos, uma vez que essa arquitetura permite ao kernel do sistema controlar operações básicas como ler, escrever ou listar arquivos, sem que seja necessário um conhecimento do tipo de sistema de arquivos, tanto pelo usuário quanto pelo programa.

Existem três tipos de arquivos no Solaris: diretórios, que podem conter arquivos ou outros diretórios; arquivos regulares, contendo qualquer tipo de dado que o usuário deseje; e arquivos especiais, que, como já visto, estão associados a dispositivos de entrada/saída (locais ou remotos).

O Solaris suporta três tipos de sistema de arquivos: sistema de arquivos baseados em disco, que podem ser escritos em diferentes formatos e são armazenados fisicamente em discos flexíveis, discos rígidos ou CD-ROMs; sistema de arquivos virtual, baseados em memória para proporcionar acesso ao núcleo do sistema sem utilizar espaço em disco; e sistema de arquivos baseado em rede, que são acessados através da rede, por intermédio do diretório/export

Existem dois tipos de sistema de arquivos basead os em rede, o Network File System (NFS) e o Remote File Sharing (RFS).

O NFS habilita computadores e arquiteturas diferentes – utilizando diferentes sistemas operacionais – a compartilhar arquivos através de uma rede. Dessa forma, qualquer computador tem acesso aos arquivos de outro computador. A diferença entre o NFS e o RFS, é que, enquanto o primeiro gera um sistema de arquivos genérico, este último provém uma cópia exata deum sistema de arquivos UNIX.

Por ser um sistema operacional multiusuário, o Solaris necessita de segurança para o sistema de arquivos.

Cada arquivo apresenta um nível de proteção definido pela categoria do usuário. Todo arquivo ou diretório possui um user que pertence a um grupo. Qualquer usuário que não seja dono do arquivo e não per tença ao respectivo grupo, enquadra-se na categoria others. Por fim, o administrador do sistema, chamado de root, tem acesso irrestrito a todos os arquivos. Dependendo da categoria do usuário, três tipos de acesso podem ser concedidos, read, write ou execute.