Alocação de memória e gerenciamento de processos

Isaque de Araújo

Trabalho de Arquitetura de Plataformas/2003

isaquearaujo22@gmail.com

Alocação de memória, consiste no processo de solicitar memória durante o processo de execução de um programa de computador. Nos sistemas operacionais, um processo é a forma de apresentar um programa em execução.

Alocação de Memória

Quando o programa requisita um bloco de memória, o gerenciador o disponibiliza para a alocação. Existem vários tipos de alocação de memória:

Alocação Estática: Decisão tomada quando o programa é compilado.

Quando o programa é executado o Sistema operacional o lê e cria um processo, sendo o programa uma noção estática e o processo o programa em execução, que é criado em armazenamento primário e após isso recebe um espaço na memória. O espaço de memória é dividido em varias partes, uma das partes se chama segmentos de memória, que armazena dados estáticos, e outro se chama segmento de código que guarda instruções do programa. Quando o programa é executado o registrador PC apontará para determinado endereço do segmento de código do processo, que se chama TEXT. Para que se realize a alocação estática o compilador deve saber o total de memória que está livre, mandar esta informação para o SO para que este crie um segmento de dados.

Alocação Contígua Simples: implementado nos primeiros sistemas operacionais -a memória principal é dividida em duas partes, uma para o sistema operacional e a outra para o programa do usuário . -não permite a utilização eficiente dos recursos do sistema, pois apenas um usuário pode dispor destes recursos. -todos os programas estão limitados ao tamanho da memória principal disponível para o usuário.

Segmentação de Programas: dividir o programa em módulos. -execução independente de cada módulo, utilizando a mesma área de memória (overlay). -a grande vantagem da utilização desta técnica consiste em se poder executar programas maiores do que a memória física disponível.

Alocação Dinâmica: Decisão é adiada até a execução. (Permite Swapping)

Os objetos alocados dinamicamente podem ser criados e liberados a qualquer momento, em qualquer ordem, o que difere dos objetos locais das funções, que são criados e destruídos em uma ordem específica. Dado isto, é preciso organizar a memória para objetos dinâmicos de uma forma que possibilite o gerenciamento do tempo de vida dos objetos por parte do programador. A memória reservada para objetos dinâmica costuma ser chamada de heap, existem várias formas de organizar um heap. Em linguagens sem gerenciamento automático (linguagem C), da memória dinâmica, uma organização usual do heap é uma lista encadeada de blocos livres, porém este tipo de organização pode ter problemas devido à fragmentação dos blocos. Já em linguagens com gerenciamento automático de memória dinâmica (Java), a organização do heap depende da parte do sistema de tempo de execução encarregada deste gerenciamento. Este componente é normalmente chamado de coletor de lixo.

Alocação Local:

Este processo de alocação é usado para variáveis que são locais a funções e sub-rotinas. Isso significa que o processo em execução deve manter acessível as variáveis locais da função ou procedimento que está executando no momento. Além disso, pelas propriedades do escopo em blocos, também devem estar acessíveis as variáveis de blocos mais externos. Em linguagens que permitem a definição de funções aninhadas, acessando as variáveis de quaisquer funções definidas externamente à função atualmente em execução. Como uma função pode chamar outras funções, um número arbitrário de funções pode estar no meio de sua execução em um determinado momento, mesmo que apenas uma esteja realmente sendo executada, isso indica que o contexto de várias funções deve ser mantido enquanto as mesmas não concluíram sua execução.

Gerenciamento de Processos

Nos sistemas operacionais, um processo é a forma de representar um programa em execução. É o processo que utiliza os recursos do computador - processador, memória, etc - para a realização das tarefas para as quais a máquina é destinada. (ALECRIM, 2005) Um processo pode ser descrito como parte de um programa que está aparentemente rodando.

Este aparente existe somente pelo fato de que determinado processo pode entrar e sair diversas vezes do processador em um único segundo, e em um determinado momento ele pode não estar no processador e mesmo assim aparentemente estar rodando. Como qualquer sistema de compartilhamento de tempo o Linux consegue dar a impressão de execução simultânea dos processos, separando um espaço bastante curto de tempo para cada um deles. Para ter sucesso nesta tarefa ele segue uma série de regras que não desperdiça tempo de hardware com operações desnecessárias e consegue escolher qual processo deve ser executado naquele exato momento.

O que decide essa escolha no kernel é o escalonador de processos, que em grande parte é responsável pela produtividade e eficiência do sistema. Mais do que um simples mecanismo de divisão de tempo, ele é responsável por uma política de tratamento dos processos que permite os melhores resultados possíveis.

Composição de um processo

O sistema operacional lida com uma infinidade de processos e, por isso, é necessário ter meios que permitam controlá-los. Para isso, os processos contam com um conjunto de características, dentre as quais:

Proprietário do processo;

Estado do processo (em espera, em execução, etc)

Prioridade de execução

Recursos de memória

O trabalho de gerenciamento de processos precisa contar com as informações acima e com outras de igual importância para que as tarefas sejam executadas da maneira mais eficiente. Um dos meios usados para isso é atribuir a cada processo um PID.

Escalonamento Circular com Prioridades: implementa o conceito de fatia de tempo e de prioridade de execução associada a cada processo. Neste escalonamento, um processo permanece no estado de execução até que termine seu processamento, ou voluntariamente passe para o estado de espera (interrupção por E/S), ou sofra uma preempção por tempo ou prioridade. A principal vantagem deste escalonamento é permitir um melhor balanceamento no uso do processador, com a possibilidade de diferenciar o grau de importância dos processos através da prioridade (o Windows e o Unix utilizam este escalonamento.

PID e PPID

Um PID (Process Identifier) é um número de identificação que o sistema dá a cada processo. Para cada novo processo, um novo número deve ser atribuído, ou seja, não se pode ter um único PID para dois ou mais processos ao mesmo tempo.

Os sistemas baseados em Unix precisam que um processo já existente se duplique para que a cópia possa ser atribuída a uma tarefa nova. Quando isso ocorre, o processo "copiado" recebe o nome de "processo pai", enquanto que o novo é denominado "processo filho". É nesse ponto que o PPID (Parent Process Identifier) passa a ser usado: o PPID de um processo nada mais é do que o PID de seu processo pai.

UID e GID

Conforme já mencionado, cada processo precisa de um proprietário, um usuário que seja considerado seu dono. A partir daí, o sistema saberá, através das permissões fornecidas pelo proprietário, quem pode e quem não pode executar o processo em questão. Para lidar com os donos, o sistema usa os números UID e GID. O Linux gerencia os usuários e os grupos através de números conhecidos como UID (User Identifier) e GID (Group Identifier). Como é possível perceber, UID são números de usuários e GID são números de grupos. Os nomes dos usuários e dos grupos servem apenas para facilitar o uso humano do computador. Cada usuário precisa pertencer a um ou mais grupos. Como cada processo (e cada arquivo) pertence a um usuário, logo, esse processo pertence ao grupo de seu proprietário. Assim sendo, cada processo está associado a um UID e a um GID. Os números UID e GID variam de 0 a 65536. Dependendo do sistema, o valor limite pode ser maior. No caso do usuário root, esses valores são sempre 0 (zero). Assim, para fazer com que um usuário tenha os mesmos privilégios que o root, é necessário que seu GID seja 0.

Referências

http://pt.slideshare.net/virginiabetiatto/gerncia-de-processos

http://pt.wikipedia.org/wiki/Aloca%C3%A7%C3%A3o_de_mem%C3%B3ria

http://pt.wikipedia.org/wiki/Gerenciamento_de_mem%C3%B3ria

http://www.jvasconcellos.com.br/unijorge/wp-content/uploads/2011/04/ger_memoria.pdf