Sistemas operacionais: Conceitos e Mecanismos

Capítulo 15: Uso da memória

Integrantes do grupo: Jefferson Botitano e Leonardo Ludvig
Professor: Arliones Stevert Hoeller Junior
Disciplina: Sistemas operacionais

INSTITUTO FEDERAL
Santa Catarina

Campus São José

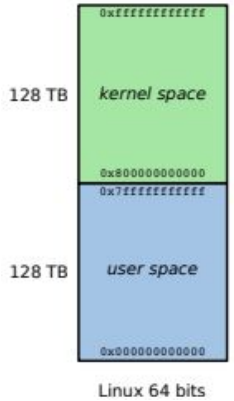
Espaço de endereçamento virtual de um processo

Organização: Divido entre duas partes uma inicial para o processo e final para o núcleo do sistema operacional,as flags de controle da tabela de páginas sinalizam as páginas do processo como user e as páginas do núcleo não o são.

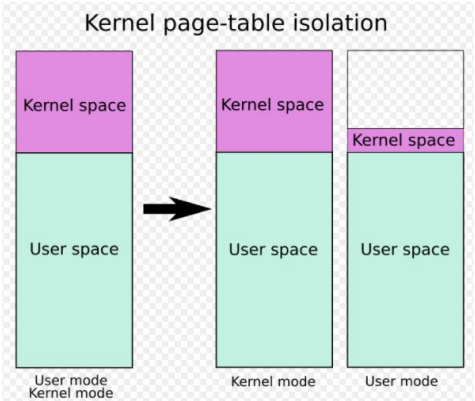
Relação entre código do núcleo e do processo: As flags realizam a separação da região de endereçamento entre parte do processo e parte do núcleo e quanto a relação aos códigos presentes nestes endereçamento o código presente na região do processo não tem acesso a parte do núcleo mas o código do núcleo pode acessar as páginas do processo.

Bug de segurança Meltdown: Esse bug permite que a um código de usuário ignorar flags da tabela de páginas e ler partes da memória do núcleo,tendo possibilidade de expor dados sensíveis lidos/escritos por outros processos, como senhas ou chaves de criptografia.

Solução bug de segurança Meltdown: Retirar o núcleo do espaço de endereçamento dos processos como por exemplo no linux que possui Kernel Page-table Isolation (KPTI) que mantém apenas um pequeno conjunto de páginas do núcleo no espaço de endereços do processo para que receber alguma chamada de sistema e interrupções e para o restante da memória do núcleo se usa uma tabela de páginas a parte.



Linux 64 bits



A memória de um processo

INSTITUTO FEDERAL
Santa Catarina

Organização:O sistema operacional implementa cada processo de uma forma que pareça uma “cápsula” de memória isolada dos demais processos,sendo assim uma área de memória exclusiva onde quem tem conhecimento dela é apenas o processo e o núcleo onde armazena as informações necessárias para sua execução e cabe ao núcleo armazenar a lista de seções de memória de cada processo.

Seções de memória de um processo

- **TEXT:** Situada no inicio do espaço de endereçamento de um processo, possui tamanho fixo e é a parte responsável por conter o código binário a ser executado pelo processo, gerado durante a compilação e ligação com as bibliotecas e armazenado no arquivo executável.
- **DATA:** Contém as variáveis estáticas inicializadas, ou seja, as variáveis que já estão definidas com valor inicial no código fonte do programa e são utilizadas do início até o fim da execução de um processo, esses valores pré-definidos são armazenadas no arquivo executável e são carregados para a seção DATA da memória quando o processo inicia.
- **BSS:** É a seção que contém as variáveis que não precisam ter seu valor inicial já pré-definido no arquivo executável, sendo chamadas de variáveis estáticas não-inicializadas.
- **HEAP:** Uma seção com alocação de memória dinâmica, justamente por ser a responsável por conter as variáveis dinâmicas do processo, tendo o seu final definido por um ponteiro chamado Program Break que pode ter seu tamanho aumentado ou diminuído através de chamadas de sistema.
- **STACK:** Responsável por gerenciar o fluxo de execução nas chamadas de função e também para armazenar os parâmetros, variáveis locais e o valor de retorno das funções, é a estrutura que mantém a pilha de execução de um processo. Podendo ter o seu tamanho fixo ou dinâmico, geralmente iniciado nos endereços maiores e crescendo “para baixo” na área de endereçamento da memória de um processo.

Alocação de variáveis

INSTITUTO FEDERAL
Santa Catarina

Variáveis são espaços de memória reservados e nomeados utilizados pelos programas em execução para armazenar informações, como um “Int cont” em linguagem C indica uma área de memória de 4 bytes que pode armazenar um número inteiro e cujo nome é cont. Em linguagens de programação as variáveis podem ser alocadas na memória dessas três formas:

- **Alocação estática:**A variável recebe um espaço de memória que é definido durante a compilação do programa e para o espaço de memória RAM é reservado no início da execução do processo e mantido até o término do processo.
- **Alocação automática:**Por padrão as variáveis dentro de uma função são alocados automaticamente na pilha de execução do programa e o espaço usado para armazenar essa variáveis é alocado somente quando a função é chamada e quando termina a função o espaço utilizado é liberado.
- **Alocação dinâmica:**O processo faz requisitos de espaços que necessita para armazenar seus dados, utiliza durante a execução e no término os libera ou quando não forem mais necessários durante a execução.

Atribuição de endereços

INSTITUTO FEDERAL
Santa Catarina

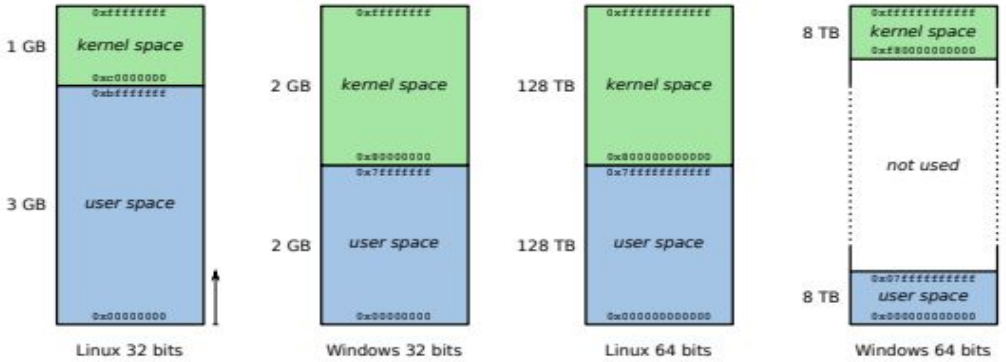
A realização de um programa utilizando diferente linguagem de programação, o programador usa nomes para variáveis e a atribuição de endereços durante essas definições de nomes podem ocorrer em diferentes partes do programa como:

- **Na edição:**O programador é responsável por indicar os endereços de cada variável e do código do programa na memória
- **Na compilação:**O compilador escolhe as posições das variáveis na memória.
- **Na ligação:**Compilador traduz o código fonte para binário e o ligador que analisa o arquivo de saída gerado pelo compilador em em conjunto a tabela de símbolos define os endereços de memória dos símbolos que resulta no programa executável.
- **Na carga:**É possível definir os endereços de variáveis e de funções durante a carga do código em memória para o lançamento de um novo processo.
- **Na execução:**Durante a execução do processo o processador emite endereços que serão convertidos em endereços efetivos a serem acessados na memória real.

2)Como é organizado o espaço de memória de um processo?

INSTITUTO FEDERAL
Santa Catarina

O espaço de memória de um processo é dividido entre o espaço de endereços para uso do processo, geralmente contendo os endereços iniciais enquanto a parte final é reservada para o núcleo do sistema operacional, não sendo possível para o código no espaço do usuário acessar as páginas de memória do núcleo mas sendo permitido acesso do código núcleo ao espaço de memória inteiro.



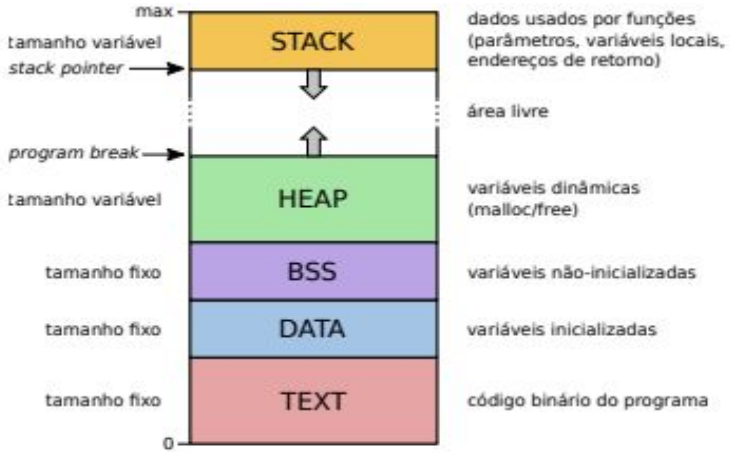


Figura 15.2: Organização da memória de um processo.

Internamente no espaço de usuário de um processo são definidos as seções de memória responsáveis armazena as informações necessárias para sua execução e cabe ao núcleo armazenar a lista de seções de memória de cada processo.

3)Sobre as afirmações a seguir, relativas ao uso da memória RAM pelos processos, indique quais são incorretas, justificando sua resposta:

INSTITUTO FEDERAL
Santa Catarina

(a) A área de memória TEXT contém o código-fonte compilado e executado pelo processo.
(b) A área de memória DATA é usada para armazenar todas as variáveis e constantes usadas pelo processo.
F. Apenas as variáveis inicializadas são armazenadas.
(c) A área de memória HEAP é usada para as alocações dinâmicas de memória, sendo usada através de funções como malloc e free.
(d) A área de memória STACK contém as pilhas do programa principal e das demais threads do processo.
F. Contêm somente as pilhas do programa principal.

4)Explique em que consiste a resolução de endereços nos seguintes momentos: codificação, compilação, ligação, carga e execução.

INSTITUTO FEDERAL
Santa Catarina

- **Codificação:**Define o endereço durante a execução do processo.
- **Compilação:**Durante compilação do código fonte o compilador determina as posições das variáveis na memória para evitar problemas,sendo assim todos os códigos fontes precisam ser reconhecidos durante a compilação.
- **Ligação:**Compilador traduz o código fonte para binário e o ligador que analisa o arquivo de saída gerado pelo compilador em em conjunto a tabela de símbolos define os endereços de memória dos símbolos que resulta no programa executável.
- **Carga:**Carrega os endereços do código do processo e define os endereços de memória que foram indicados no código.
- **Execução:**Durante a execução do processo o processador emite endereços que serão convertidos em endereços efetivos a serem acessados na memória real.