Segurança de Sistemas Operacionais

Roberto Sadao Yokoyama

UTFPR-CP

Agosto, 2016

Até agora:

- Introdução à segurança computacional
- Segurança física

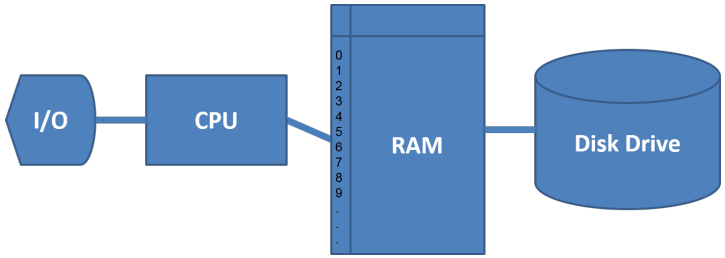
Aula de hoje: [1]

1. Revisão: Conceitos de Sistemas Operacionais
2. Segurança de Sistemas Operacionais
   - Segurança de processos

[1] Slides baseados no material do livro: Goodrich [1]

Um modelo de computador

Um sistema operacional tem que lidar com o fato de que um computador é composto por um CPU, memória de acesso aleatório (RAM), dispositivos de entrada/saída (E/S) e armazenamento a longo prazo.



Conceitos de SO

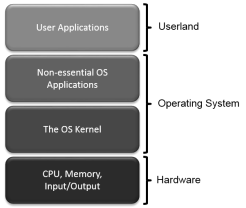
- Um sistema operacional (SO) fornece a interface entre os usuários de um computador e o hardware desse computador.
- Um sistema operacional gerencia as maneiras de as aplicações acessarem os recursos em um computador, incluindo suas unidades de disco, CPU, memória principal, dispositivos de entrada, dispositivos de saída e interfaces de rede.
- Um sistema operacional gerencia vários usuários.
- Um sistema operacional gerencia vários programas.

Multitarefa

- "Dar"a cada programa em execução uma "fatia"do tempo da CPU
- A CPU está executando tão rápido que, para qualquer usuário, parece que o computador está executando todos os programas em simultâneo

O Kernel

- O *kernel* é o componente central do sistema operacional. Ele trabalha com o gerenciamento de recursos de *hardware* de baixo nível, incluindo memória, processadores e dispositivos de entrada/saída (E/S), tais como um teclado, mouse ou tela de vídeo
- A maioria dos sistemas operacionais definem as tarefas associadas com o *kernel* em termos de uma metáfora de **camada**, com os componentes de *hardware*, tais como a CPU, memória e dispositivos de entrada/saída, na parte inferior e os usuários e aplicações na parte superior do modelo.



Entrada/Saída

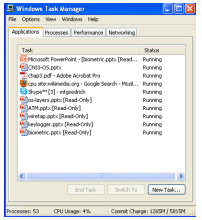
- Os dispositivos de E/S de um computador incluem coisas como o seu teclado, mouse, monitor de vídeo e placa de rede, bem como outros dispositivos mais opcionais, como um *scanner*, interface Wi-Fi, câmera de vídeo, portas USB, etc.
- Cada aparelho é representado em um sistema operacional usando um **driver de dispositivo**, que encapsula os detalhes de como deve ser feita a interação com esse dispositivo .
- A interface de programação de aplicativo (**API**), que os *drivers* de dispositivo apresentam aos programas de aplicação, permite a esses programas interagir com esses dispositivos em um nível razoavelmente alto, enquanto o sistema operacional faz o "trabalho pesado" de realizar as interações de baixo nível que fazem esses dispositivos realmente funcionar.

Chamadas de sistema

- Aplicações de usuário não se comunicam diretamente com componentes de *hardware* de baixo nível, e sim delegam essas tarefas para o *kernel* através de **chamadas do sistema**.
- Chamadas do sistema são geralmente contidas em uma coleção de programas, isto é, uma biblioteca como a biblioteca de C (libc), e elas fornecem uma interface que permite às aplicações usarem uma série predefinida de APIs que definem as funções para a comunicação com o *kernel*.
- Exemplos de chamadas do sistema incluem aquelas que realizam operações em arquivo de E/S (abrir, fechar, ler, escrever) e executar programas de aplicação (exec).

Processos

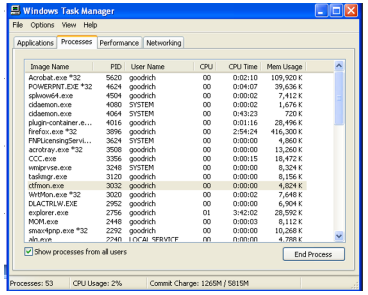
- Um processo é uma instância de um programa que está sendo executado no momento
- Inicialmente, o conteúdo real de todos os programas é armazenado no armazenamento persistente, como um disco rígido.
- A fim de ser executado, um programa deve ser carregado na memória de acesso aleatório (RAM) e identificado exclusivamente como um processo.
- Desta forma, várias cópias do mesmo programa podem ser executadas como processos diferentes.
- Por exemplo, nós pode ter várias cópias do MS Powerpoint aberto ao mesmo tempo.



Processos ID

- Cada processo em execução em um determinado computador é identificado por um inteiro não negativo único, chamado de ID de processo (PID).
- Por meio do PID de um processo, podemos então pode associar seu tempo de CPU, uso de memória, usuário ID (UID), nome do programa, etc

  ex. de comandos relacionados: *pstree; ps aux; top*

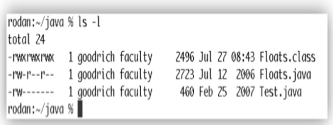


Sistemas de arquivos

- Um sistema de arquivos é uma abstração de como a memória externa, não-volátil do computador é organizada.
- Sistemas operacionais geralmente organizam arquivos de forma hierárquica em **pastas**, também chamadas de *diretórios*
- Cada pasta pode conter arquivos e/ou subpastas
- Assim, um volume ou unidade, consiste de uma coleção de pastas aninhadas que formam uma árvore.
- A pasta de nível superior é a raiz desta árvore e também é chamada de pasta raiz.

Permissões de arquivo

- Permissões de arquivos são verificadas pelo sistema operacional para determinar se um arquivo é legível, gravável ou executável por um usuário ou grupo de usuários.
- Em sistemas operacionais similares a UNIX, possuem uma matriz de permissão de arquivo. Essa matriz é representação de quem pode fazer o que no arquivo.
- Os arquivos têm permissões de proprietário (*owner*), determinam as permissões do criador do arquivo. As permissões do grupo (*group*), que determina permissões para usuários do mesmo grupo. Por fim, a classe outros (*others*), que determinam as permissões de usuários que não são nem o dono do arquivo, nem pertencem ao mesmo grupo



Gerenciamento de memória

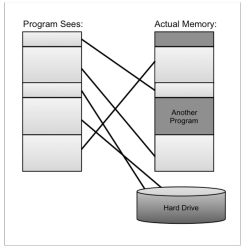
- A memória RAM de um computador é o seu "espaço de endereçamento"
- Ele armazena o código de um programa em execução, seus dados de entrada e sua memória de trabalho.
- Para qualquer processo em execução, é organizado em diferentes segmentos, que separa as diferentes partes do espaço de endereço
- As questões de segurança de memória exigem que nunca devemos misturar estes segmentos diferentes

Organização da memória

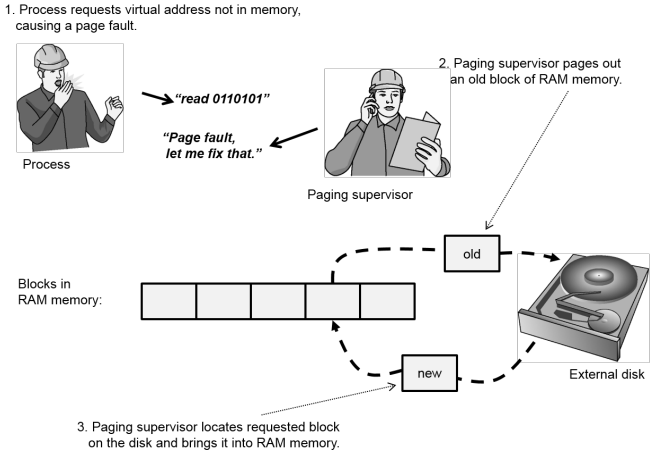
- **Texto** (código): este segmento contém o código de máquina real (binário) do programa.
- **Dados**: Este segmento contém as variáveis estáticas do programa, que foram inicializadas no código do programa.
- **BSS**: Neste segmento, que é denominado por um acrônimo antiquado de *block started by symbol*, contém variáveis estáticas que são não inicializadas.
- **Heap**: Neste segmento, que também é conhecido como o segmento dinâmico, armazena dados gerados durante a execução de um processo
- **Pilha**: Este segmento abriga uma estrutura de dados em pilha que cresce para baixo e é usada para controlar a estrutura de chamada de sub-rotinas (por exemplo, os métodos em Java e funções em C e seus argumentos).

Memória virtual

- Geralmente não há memória suficiente no computador para endereçar todos os processos em execução
- No entanto, o sistema operacional dá para cada processo em execução, a ilusão de que ele tenha acesso ao seu espaço de endereço completo (contíguos)
- Na realidade, essa visão é virtual, em que o sistema operacional suporta este ponto de vista, mas não é como a memória é organizada.
- Em vez disso, memória é dividida em páginas, e o sistema operacional mantém controle sobre quais páginas estão na memória e quais delas são armazenados no disco.

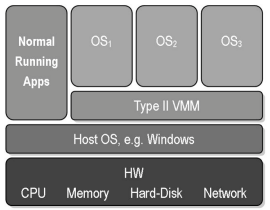


Falhas de página



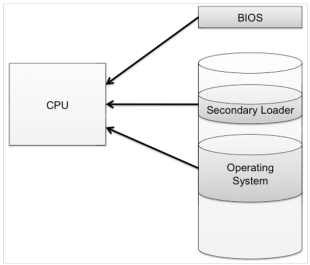
Máquinas virtuais

- Máquina virtual: permite a um sistema operacional executar sem contato direto com seu hardware adjacente. Por exemplo, um emulador de Windows em um Mac.
- Benefícios:
  - Eficiência de hardware
  - Portabilidade
  - Gerenciamento
  - Segurança



A sequência de inicialização

- A ação de carregar um sistema operacional na memória de um estado desligado é conhecida como *boot* ou inicialização.
- Quando um computador é ligado, ele primeiro executa código armazenado em um componente de *firmware*, conhecido como o BIOS (sistema básico de entrada/saída).
- Em sistemas modernos, o BIOS carrega na memória, o segundo carregador de inicialização, que trata de carregar o restante do sistema operacional na memória e, em seguida, passa o controle de execução para o sistema operacional.

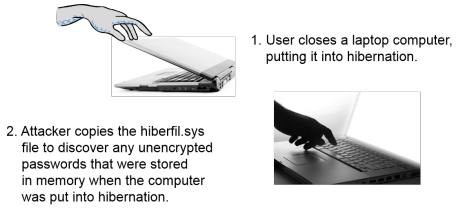


BIOS Passwords

- Um usuário mal-intencionado poderia potencialmente capturar a execução de um computador em vários pontos do processo de inicialização.
- Para evitar que um atacante atue nas primeiras etapas da inicialização, muitos computadores possuem uma senha de BIOS que não permite que um carregador de inicialização de segundo estágio seja executado sem autenticação apropriada.

Hibernação

- Máquinas modernas têm a capacidade de entrar em um estado desligado conhecido como hibernação.
- Ao entrar em hibernação, o sistema operacional armazena o conteúdo da memória da máquina em um arquivo de hibernação no disco (hiberfil.sys), de modo que o computador pode ser facilmente restaurado ao seu estado quando ligado novamente.
- Mas... sem as devidas precauções de segurança adicionais, hibernação expõe uma máquina para investigação forense potencialmente invasiva



Log de eventos

- Manter o controle de quais processos estão em execução, que outras máquinas que interagem com o sistema através da Internet, e se o sistema operacional tiver sofrido qualquer comportamento inesperado ou suspeito podem muitas vezes deixar pistas importantes não só para a solução de problemas comuns, mas também para determinar a causa de uma quebra de segurança.
- ex. */var/log*

Referências

[1] M. T. Goodrich and R. Tamassina. *Introdução à Segurança de Computadores*. Bookman, 2013.