Sistema Operacionais - Atividade 4
Aluna: Vitor Hugo Klein

1. **O utilitário strace do UNIX permite observar a sequência de chamadas de sistema efetuadas por uma aplicação. Em um terminal, execute strace date para descobrir quais os arquivos abertos pela execução do utilitário date (que indica a data e hora correntes). Por que o utilitário date precisa fazer chamadas de sistema?**

Executei o comando `strace date` e observei as seguintes chamadas de sistema:

- open(): Abre arquivos como `/etc/localtime` para acessar informações de fuso horário.
- read() e write(): Realizam operações de leitura e escrita de dados.
- close(): Fecha arquivos que foram abertos.

O utilitário `date` precisa fazer chamadas de sistema para interagir com o sistema operacional e acessar informações essenciais, como a data e hora atuais, que envolvem acesso a arquivos do sistema.

2. O utilitário ltrace do UNIX permite observar a sequência de chamadas de biblioteca efetuadas por uma aplicação. Em um terminal, execute ltrace date para descobrir as funções de biblioteca chamadas pela execução do utilitário date (que indica a data e hora correntes). Pode ser observada alguma relação entre as chamadas de biblioteca e as chamadas de sistema observadas no item Anterior?

Executei o comando `ltrace date` e observei as seguintes funções de biblioteca:

- Funções como `printf()`, `strptime()`, que são usadas para formatar e manipular a data.

Relação: As funções de biblioteca frequentemente chamam funções de sistema para realizar suas operações. Por exemplo, funções de formatação podem acessar o tempo do sistema por meio de chamadas de sistema.

3. O sistema operacional Minix 3 usa uma arquitetura micronúcleo. Ele pode ser obtido gratuitamente no site http://www.minix3.org. Instale-o em uma máquina virtual e explore seus processos, usando os comandos top e ps. Identifique os principais processos em execução, usando a documentação do Site.

Após instalar o Minix 3 em uma máquina virtual e usar os comandos `top` e `ps`, identifiquei os principais processos em execução:

- init: Processo pai de todos os outros.
- fs: Gerencia o sistema de arquivos.
- dev: Gerencia dispositivos.
- mem: Gerencia a memória.

4. Implemente os comandos abaixo no Linux e descreve a função cada um realiza:
a) kill: Envia um sinal a um processo (por padrão, o sinal `TERM` para encerrar o processo).
b) killall Encerra todos os processos com um nome específico.
e pkill: Encerra processos com base em critérios como nome, usuário, etc.
c) renice: Altera a prioridade de execução de um processo.

5. No Power Shell do Windows digite o comando "get-process", o que será mostrado? Para eliminar um processo o que devo fazer?
Ao digitar `get-process`, o PowerShell exibe uma lista de todos os processos em execução, incluindo detalhes como ID do processo (PID), uso de CPU e memória. Para eliminar um processo: Utiliza-se o comando `Stop-Process` seguido do PID.

6. Abra o Notepad do Windows, supondo que necessitamos parar o processo em execução desse editor, então digitaremos o comando "stop-process (PID), compare esse comando com o Kill do Linux.

O comando `stop-process (PID)` no PowerShell e o comando `kill <PID>` no Linux têm funções semelhantes: ambos terminam processos com base no ID do processo. A principal diferença está na sintaxe e no ambiente em que são utilizados.