

Sistemas operacionais: Conceitos e Mecanismos

Capítulo 2: Estruturas de um SO

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

Elementos do sistema operacional

Os elementos de um sistema operacional são diversos tendo suas funcionalidades e objetivos que se complementam. Alguns destes principais componentes são:

- Núcleo: responsável por gerenciar recursos que o hardware disponibiliza utilizados quando é requisitado pelas aplicações.
- Código de inicialização: responsável por inicializar o hardware reconhecendo dispositivos instalados e realizando testes e configurações para uso posterior e realizar a tarefa importante de carregar o núcleo do sistema operacional em memória e iniciar sua execução
- Drivers: módulos de código específicos para acessar hardware periféricos ao hardware principal, existem diversos drivers para cada tipo de dispositivo, como portas USB, placas gráficas entre outras.
- Programas utilitários: para facilitar o uso do sistema computacional é utilizado programas que realizam funções que complementam o núcleo, como configurar dispositivos, interpretador de comandos , terminal , interface gráfica, entre outros.

Elementos de hardware

Para que um sistema operacional possa cumprir suas funcionalidades com eficiência e confiabilidade é necessário que o hardware forneça funcionalidades básicas.

- **Arquitetura do computador:** formado por um ou mais processadores que são responsáveis de execuções de instruções das aplicações. Possui barramentos de dados que leva a informação a ser transferida entre o processador e a memória ou um periférico, instruções, controle que indica que operação deve ser realizada e endereço aponta a operação de escrita ou leitura. Possui uma unidade lógica aritmética que é responsável por realizar as operações lógicas em conjunto com os registradores que armazenam informações requisitadas para operação. Esta arquitetura é básica para um computador, podendo ter maior complexidade de acordo com a aplicação
- **Interrupções e exceções:** a interrupção é iniciada quando é requisitada o fluxo de exceção é detido e executa uma rotina de endereço predefinido que contém a rotina de tratamento de interrupção que executa as ações necessárias para atender o dispositivo que a requisitou. Situações que são geradas pelo próprio processador podem deter a execução usando o mesmo mecanismo de interrupções mas são denominadas exceções, que podem ser instruções inexistentes ou com operando inválidos, cálculos como divisão por zero ou erros em códigos podem disparar essas exceções no processador que resulta no processo da ativação da rotina de tratamento de exceções tendo a mesma analogia da rotina de interrupções.
- **Níveis de privilégio:** códigos de máquina devem ser diferenciados quando se trata de interagir com o hardware para poder gerenciar os aplicativos a partir de restrição para o hardware para que não ocorra interferência em configurações do gerenciamento que podem impactar no sistema por completo. Para que problemas como este não ocorra existe a necessidade de diferenciar privilégios de execução dos diferentes tipos de softwares por níveis de privilégios de execução, sendo controlados por flags especiais na CPU, que podem ser combinados de acordo com o código que está sendo executado e cada processador possuem níveis de privilégios diferentes mas os básicos são como: nível de núcleo que um código em andamento neste nível o processador sempre possui recursos disponíveis, disponibilizando todas as instruções que dispõe de maneira a ser executada pelos recursos internos de hardware. O nível de usuário possui apenas um grupo separado de instruções que o processador e registradores disponibilizam, como instruções de categorias “perigosas” sendo caso do RESET que realiza a inicialização do processador e o acesso de portas de entrada/saída, nestes casos o código em andamento tente executar as instruções de tipo “perigosas” o hardware compreende a situação e gera uma exceção sendo assim realiza o desvio para rotina de tratamento de exceção.

Chamadas de sistema

Os sistemas operacionais definem as chamadas de sistema para qualquer operação que rodeiam o acesso a recursos de baixo nível que são periféricos, arquivos, alocação de memória, entre outros, para abstrações lógicas também é realizada chamada de sistema que realiza o procedimento de finalizar tarefas ou como tanto inicializar-las, entre outros. Alguns exemplos de chamada de sistema são:

- Gestão de processos: criar, carregar código, finalizar, esperar, ler/mudar atributos.
- Gestão de memória: alocar, liberar, modificar áreas de memória.
- Gestão de arquivos: criar, remover, abrir, fechar, ler, escrever, ler/mudar atributos
- Comunicação: criar/destruir canais de comunicação, receber/enviar dados.
- Gestão de dispositivos: ler/mudar configurações, ler/escrever dados.
- Gestão do sistema: ler/mudar data e hora, desligar/suspender/reiniciar o sistema.

O que diferencia o núcleo do restante do sistema operacional?

O núcleo é a parte mais importante do sistema operacional, ele é o coração do sistema, sendo o responsável por toda a gerência dos recursos do hardware utilizados pelas aplicações. Além de operar com privilégio sobre os outros componentes de um sistema operacional ele também implementa as principais abstrações utilizadas pelos programas.

Quais as diferenças entre interrupções, exceções e traps?

A interrupção, exceção e trap são rotinas especiais desenvolvidas para interromper o funcionamento do processador em devida operação e direcionar o fluxo de execução para um endereço pré determinado, onde se encontra a rotina de interrupção que deve ser executada, e em seguida retorna o processador para o processo ao qual ele estava anteriormente.

Essas instruções são necessárias para casos onde é necessário realizar uma ação imediatamente mesmo com outra aplicação em execução, como o click de um mouse ou o recebimento de um pacote de rede.

A diferença entre elas seria o que origina essa interrupção, quando se vem de dispositivos como um mouse ou teclado são tratadas como interrupções, quando é proveniente do próprio processador como um erro em alguma instrução é chamado de exceção e no caso das traps seria uma forma de utilizar a interrupção como ferramenta para acessar o privilégio no nível de kernel quando houver a necessidade estando em nível de aplicação.

O comando em linguagem C fopen é uma chamada de sistema ou uma função de biblioteca? Por quê?

O fopen é uma função de biblioteca, pois ao executar o mesmo a aplicação realiza uma chamada de sistema, porém não é o fopen quem realiza a chamada diretamente, ele na verdade chama uma função a qual essa sim invoca uma chamada de sistema. Como por exemplo no caso do Unix, onde o fopen da biblioteca C chama o open, que seria a verdadeira chamada de sistema do sistema operacional.

Coloque na ordem correta as ações abaixo, que ocorrem durante a execução da função `printf("Hello world")` por um processo (observe que nem todas as ações indicadas fazem parte da sequência).

- (h) O processo chama a função `printf` da biblioteca C.
- (c) A função de biblioteca `printf` recebe e processa os parâmetros de entrada (a string "Hello world").
- (d) A função de biblioteca `printf` prepara os registradores para solicitar a chamada de sistema `write()`.
- (g) Uma interrupção de software é acionada.
- (a) A rotina de tratamento da interrupção de software é ativada dentro do núcleo.
- (i) A operação de escrita no terminal é efetuada ou agendada pela rotina de tratamento da interrupção.
- (j) O controle volta para a função `printf` em modo usuário.
- (b) A função `printf` finaliza sua execução e devolve o controle ao código do processo.

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?

O comando `date` tem a necessidade de realizar chamadas de sistema pois o mesmo busca o horário diretamente do hardware, no relógio da BIOS, também chamado de relógio do hardware, então é preciso realizar a chamada de sistema para ter acesso a essa informação e sincronizar a data do sistema com a data do hardware.