

EA872 Laboratório de Programação de Software Básico

Atividade 7



Vinícius Esperança Mantovani

RA 247395

Entrega (limite): em sala /09/2023, 13:00

Atividade 1)

O que é execução em modo núcleo?

Resposta: A execução em modo núcleo é o modo de execução da CPU em que são executados o sistema operacional e processos associados a ele. Este modo, garante ao processo permissão e possibilidade de usar todas as instruções que a máquina é capaz de executar.

Atividade 2)

Qual é a diferença de execução em modo núcleo para execução em modo usuário?

Resposta: A diferença entre modo núcleo e modo usuário é que o modo núcleo tem acesso a todas as instruções que a máquina é capaz de executar, enquanto que o modo usuário tem acesso a apenas uma parte dessas instruções, sendo, portanto, mais limitado. Nesse sentido, enquanto o modo núcleo executa majoritariamente processos do sistema operacional, o modo usuário executa os demais processos, como diz o nome, principalmente processos do usuário. Assim, o modo usuário, por ser mais restrito, precisa recorrer ao sistema operacional por meio de rotinas (pedidos) para o SO executar uma determinada tarefa como um “printf”, por exemplo.

Atividade 3)

Qual é o objetivo de se ter estes dois modos de execução em um SO?

Resposta: A existência desses dois modos de execução é devida à necessidade de segurança ao sistema do computador. Isso porque, como o modo núcleo tem acesso a todas as instruções, ele tem controle muito grande sobre o hardware. Desse modo, caso um programa qualquer fosse executado com tamanho acesso, ele seria capaz de rejeitar o sistema operacional, tomando conta da máquina e a operando da maneira como lhe conviesse. Assim, temos essa distinção para restringir as instruções às quais os programas normais têm acesso,

de modo que eles não têm tanto poder quanto o sistema operacional e seus processos que são executados em modo núcleo.

Atividade 4)

O que são chamadas de sistema?

Resposta: As chamadas de sistema são a interface entre os programas aplicativos e o sistema operacional, de modo que, é por meio delas que os processos de programas aplicativos solicitam serviços de núcleo ao sistema operacional. Ou seja, as chamadas de sistema são chamadas que os programas que executam em modo usuário fazem para solicitar ao sistema operacional que execute uma determinada tarefa para ele (o programa usuário). O sistema operacional, por sua vez, opta por aceitar ou não a solicitação.

Atividade 5)

Dê um exemplo de situação em que um programa que você está fazendo precisa recorrer a uma chamada de sistema e um exemplo em que uma chamada deste tipo não é necessária.

Resposta: Um exemplo de situação em que um programa precisa recorrer a uma chamada de sistema seria a necessidade de executar um processo de input ou output, como: a impressão de uma string no terminal ou em um arquivo de saída.

Já um exemplo de situação em que um programa não precisa recorrer a uma chamada de sistema, seria: um programa que calcula o fatorial de um número. Isso porque, as operações matemáticas são acessíveis para programas em execução no modo usuário.

Atividade 6)

Um processo é criado em Unix por meio de uma de duas chamadas possíveis: `fork()` e `exec()`. Qual é a principal diferença entre elas?

Resposta: A principal diferença entre `fork()` e `exec()` é que: `fork()` gera um novo processo (processo-filho) exatamente igual ao processo-pai, que compartilham o texto, mas têm seus espaços separados na memória para dados e pilha (espécie de bifurcação), enquanto que, a rotina `exec()`, por sua vez, é responsável por sobrescrever um processo-filho com informações de um novo processo que se deseja executar no lugar do anterior.

Atividade 7)

A memória ocupada por um processo é dividida em partes, que são gerenciadas de formas distintas. Que partes são estas e qual é a principal característica de cada uma?

Resposta: Essas partes e a principal característica de cada uma delas são:

- 1) Segmento de instruções: contém os códigos executáveis do programa, é, portanto, estático e compartilhável entre processos.
- 2) Segmento de dados do usuário: contém os dados do programa, portanto é dinâmico, no sentido que os dados são alterados durante a execução do programa e é, também, não compartilhável entre processos, pois depende do processo.
- 3) Segmento de dados do sistema e de pilha: contém as variáveis de ambiente no qual o programa é executado e as chamadas feitas por este processo, sendo, também, portanto, não compartilhável.

Atividade 8)

Por que se diz que há uma árvore de processos em Unix?

Resposta: diz-se isso porque, como tem-se um sistema de processos sendo sempre gerados de um processo-pai, temos uma estrutura abstrata semelhante à de uma árvore, que se ramifica e expande a cada criação de processo. Assim, por termos sempre um processo originador, pai, e possivelmente vários filhos, temos essa estrutura dita árvore de processos.

Atividade 9)

Quando um processo é criado como cópia de outro, como o SO consegue distinguir entre eles?

Resposta: O sistema operacional distingue entre eles por meio de seus identificadores. Ou seja, cada processo contém um número inteiro de identificação usado pelo SO para distinguir entre eles.

Atividade 10)

Qual a diferença entre os estados “Ativo” e “Em execução” de um processo?

Resposta: As diferenças entre eles são que, enquanto um processo “Ativo” não está sendo executado, ele está pronto para sê-lo, mas aguarda competindo para ser executado, um processo “Em execução” não apenas está pronto para ser executado, mas está, no presente momento, sendo executado pela CPU, ocupando-a.