

FLUXO DE CONTROLE EXCEPCIONAL RESUMO

Uma sequência de transferências de controle, é chamada de fluxo de controle, onde essas transferências ocorrem de forma interna no programa. Mas existem as mudanças bruscas e que não ocorrem internamente, os chamados fluxos de controle excepcional, que são mudanças no fluxo de controle que estão fora das ramificações normais e das chamadas de procedimento. Esses são recursos do SO para lidar com processos, memória virtual e E/S, aqui hardware e software cooperam para fornecer o mecanismo fundamental de exceção de baixo nível.

O SO usa exceções para suportar uma forma de fluxo de controle excepcional, a mudança de contexto do processo. As Exceções são uma forma de fluxo de controle excepcional implementadas parte por hardware e parte por SO, e é a resposta à alguma alteração no estado do processador. As exceções são subdivididas em classes, as *Interrupções* são captadas por um pino no chip do processador, ocorrem como resultado à sinais de E/S externos ao processador, *Trap e Chamadas do Sistema* são captadas pelo SO, existindo ainda *Falhas e Anulação*.

Nos Sistemas Linux, são 256 (0 a 255) tipos de exceções e os valores indicam na tabela, qual a classe da exceção para que uma rotina possa processar o evento. O tratamento de erro é extremamente importante, no entanto, deixa o código bem mais robusto e por isso programadores terminam não fazendo.

Falando um pouco de Processos, um programa ao ser executado é instanciado no contexto de processos e o processador executa essas instâncias de forma a causar a ilusão ao usuário de que aquele programa é o único em execução. Cada processo possui três estados: *em espera*, quando está na memória aguardando a vez de ser executado, *executando*, quando finalmente acessa o processador e realiza a tarefa, ou *finalizado*, que é o terceiro estado. Isso porque existem vários níveis de prioridade e eles alternam o acesso ao processador. Além disso, existem os fluxos simultâneos, que ocorrem quando dois processos são executados na mesma fatia de tempo, por exemplo, o processo A sendo iniciado quando o processo B está em execução, isso define uma situação de concorrência.

JADSON ANDRÉ P. SILVA

Ainda no contexto de Processos, temos os termos *pai* e *filho*, consiste no fato de que um processo pode criar outro(s) processo(s), este é o processo pai, enquanto os processos gerados, são os filhos e esses também podem gerar outro(s) processo(s), sendo processos filhos quase uma cópia do processo pai, com exceção do PID. Cada processo é identificado por um número inteiro, o PID, através do identificador é possível controlar cada um dos processos (encerrar, colocar para dormir, alterar o nível de prioridade, entre outros), e sendo esse um processo filho, é possível identificar por qual processo ele foi gerado.

Processos filhos devem ser mortos assim que o processo pai é finalizado, pois mesmo fora da área de execução consomem memória, desta forma, se o processo pai não coletar os filhos antes de finalizar, o kernel do SO realizará esse processo de coleta.

Os Sinais são outra forma de fluxo de controle excepcional, que permite que os processos e o kernel interrompam outros processos, de forma mais clara, o processo é interrompido com a chegada de um sinal, os registradores são salvos e o manipulador de sinal é ativado e após o manipulador retornar, a atividade que foi interrompida é continuada. Esse sinal é uma mensagem que indica ao processo que algum evento ocorreu no sistema e cada evento tem o tipo específico de mensagem. Assim o processo pode reagir ao sinal de três maneiras distintas: ignorar, “matar” o próprio processo, ou capturar e manipular o sinal.

Um sinal deixa de ser pendente e passa a ser entregue, quando o processo executa baseado no sinal. Chama-se de sinal pendente, quando um sinal foi enviado, porém ainda não foi recebido, podendo existir apenas um sinal pendente do tipo específico, desta forma, os demais serão descartados, outro caso é quando um processo bloqueia o sinal, que só será recebido após o desbloqueio. Descartar ou bloquear sinais são alguns dos problemas de manuseio de sinais, que ocorrem quando o programa captura múltiplos sinais.

Um outro problema citado brevemente, é a complexidade da programação simultânea. Programar fluxos que leem e escrevem no mesmo espaço de armazenamento é um desafio e tem como resultado um comportamento não-determinístico, o grande problema disso está em como sincronizar o fluxo. Em cima disso, vemos alguns exemplos que invocam a função *fork()*, mas o resultado ainda é ineficiente, pois em alguns casos ainda é possível que ocorra problemas.

JADSON ANDRÉ P. SILVA

Outra forma de fluxo de controle excepcional são Saltos Não-Locais, essa já é no nível de usuário e transfere o controle de uma função diretamente para outra que está em execução, isso sem ter que passar pela sequência de chamada e retorno. Os saltos não-locais são fornecidos por duas funções (*setjmp* e *longjmp*), onde a primeira salva o ambiente e a segunda capta esse ambiente no buffer salvo e fica disparando um retorno para a função *setjmp*. Essa aplicação evita que o programa seja encerrado, fazendo com que o erro seja decodificado e manipulado em um local do código.

Por fim, temos as Ferramentas para Manipulação de Processos, que são nada mais que ferramentas (comandos), fornecidos para monitoramento e manipulação dos processos. São importantes para o usuário domésticos, mas ainda mais importante ao programador, pois tende a facilitar no desenvolvimento, caso precise fazer algum tipo de alteração no sistema.