

Prova SB 02

9 de novembro de 2009

1 .

Caracterize a arquitetura de Von Neumann (0.5). Explique como um programa é executado em uma máquina com essa arquitetura (0.5). Conceitue gargalo de Von Neumann (0.5). Descreva uma técnica utilizada para minimizar um gargalo de Von Neumann

R:

A arquitetura de Von Neumann pode ser caracterizado por:

1. uma CPU na qual temos uma ULA, um banco de registradores, um program counter e um controlador de sequência.
2. periféricos de E/S
3. barramentos
4. memória - na qual são armazenados dados e instruções.

Um programa é executado na máquina através de um ciclo sequencial, no qual primeiro ocorre a busca da instrução, sua decodificação, busca de operandos, execução e, por fim, incremento do PC.

O gargalo de Von Neumann é um gargalo que ocorre pelo fato de não haver separação entre memória de dados e instruções, fazendo com que não se possa ler uma instrução ao mesmo tempo que se lê ou escreve um dado. A fim de minimizar esse gargalo, faz-se técnicas como pipelining, blocagem, bufferização, mecanismos de controle de E/S e memória cache.

2 .

Caracterize código absoluto e código relocável (0.5). Conceitue e exemplifique relocação estática (0.5). Qual a importância do código relocável? (0.5). Conceitue e explique como funciona um carregador de relocação.

R:

Código absoluto é um código que possui uma posição fixa em um endereço de memória. Ideal para programas de inicialização de sistemas - ou seja, programas de boot, que iniciam em um endereço de memória específico. Já código relocável é aquele que pode assumir qualquer posição de endereço de memória.

A relocação estática é um processo de relocação no qual é decidido em qual endereço de memória um código relocável será relocado. O endereço escolhido segue a base α , ou seja, o código será relocado contíguamente a partir do endereço $\text{offset} + \alpha$. Um exemplo ocorre quando se carrega um programa executável. O código do programa é alocado a um endereço que pode variar dependendo do contexto de execução e das áreas de memória disponíveis no sistema.

A importância de um código relocável vem da necessidade de saber o que pode ser relocado ou não, além de permitir melhor gerência de memória do sistema.

O carregador de relocação é um carregador que, a partir de um endereço livre fornecido pelo sistema operacional, é responsável por carregar e relocar o código para aquele endereço. Ele verifica o tamanho da memória suficiente para comportar o código, solicita uma área desse tamanho para o gerenciador de memória, obtém a resposta do sistema operacional, reloca os endereços relocáveis e transfere o controle para o programa carregado.

3 .

Conceitue montador (0.5) e ligador (0.5). Escolha um deles e descreva as estruturas de dados que utiliza e as funções dos seus dois passos (0.5). Se optar por montador, também descreva a função do pré-passo. Se optar pelo ligador, apresente um exemplo, baseado em lista encadeada reversa para resolução de referências externas (0.5).

R:

O montador é a parte responsável por transformar o código fonte assembly em um código objeto relocável, já o ligador é responsável por ligar as referências entre os códigos objetos relocáveis, criando o código de máquina relocável executável.

Escolho o montador. Algumas de duas estruturas de dados utilizadas são tabela de símbolo, lista encadeada reversa de referências externas e a tabela de relocação. Suas funções são:

1. Pré-passo - nessa fase, é passado o código fonte em assembly e seus arquivos. O montador irá verificar todas as diretrizes e macros de montagem. As macros têm seus nomes inseridos na tabela de símbolos e strings substituídas em cada uma de suas ocorrências - há também inclusão de arquivos e montagem condicional. O resultado é um código fonte puro em assembly.
2. Obtendo o resultado do pré-passo, a idéia é montar a tabela de símbolos. Em seguida, fazer análise léxica, sintática e semântica.
 - (a) a análise léxica envolve o mapeamento de conjuntos de caracteres em identificadores, delimitadores, palavras-chaves, números, dentre outros. Cada um desses conjuntos de caracteres são identificados em tipos de tokens e, para cada tipo, são anexadas informações. Os números são convertidos em forma binária. Estes dados são inseridos na tabela de símbolos.
 - (b) na análise sintática as sequências de tokens são verificadas a fim de verificar se possuem uma ordem correta.
 - (c) na análise semântica, os tokens são interpretados de forma a verificar se fazem sentido, são adicionadas informações implícitas e depois é gerado um código objeto intermediário.
3. Nessa fase, ocorre uma otimização opcional do código objeto intermediário.
4. Por fim, o código objeto intermediário é montado no código objeto relocável.

4 .

Conceitue relocação dinâmica (0.5). Descreva a abordagem de registro de realocação para implementar relocação dinâmica. Descreva o processo de mapeamento de um endereço lógico <p,d> em um endereço real em um ambiente com paginação para ilustrar as idéias (0.5). Compare a abordagem de paginação e apresentando as vantagens e desvantagens de cada uma delas. (0.5)

R:

null.

5 .

Conceitue sistema operacional e descreva suas funções (1.0). Seja um sistema operacional multiusuário. Descreva o tratamento de interrupção mascarável desde reconhecimento da fonte de interrupção de hardware (0.5). Determinação do endereço do handler da interrupção e execução propriamente dita criação salvaguarda de contexto para executar o código sobre o processo que estava executando (0.5). Qual o suporte de hardware necessário para permitir cascadeamento de interrupções (0.5).

R:

Sistema operacional é um sistema que permite gerenciar os recursos do computador e também fornecer uma interface amigável entre o hardware/firmware e o usuário.

No quesito de gerência de recursos do computador, entram gerência de interrupções, gerência de memória real e virtual, segurança e tratamento de erros, escalonamento de processos e suporte à utilização em rede.

No quesito de fornecer uma interface amigável, é possível fornecer uma interface de comandos ou então uma interface visual interativa em janelas, por exemplo.

O sistema de interrupção ocorre da seguinte maneira: verifica-se se há um sistema de interrupção mascarável na CPU. Caso tenha, envia um sinal alto (1) para o ACK, o qual vai passando por todas as fontes de interrupção, na ordem decrescente. Se uma fonte de interrupção pedir uma interrupção, o sinal é abaixado (ou seja, passa de 1→0) e sua identificação é colocada no bus de endereço. Ao final, o handler identifica e executa cada uma das interrupções, usando a identificação de cada interrupção para localizar no vetor de interrupções o endereço para a rotina da interrupção a ser executada.

O suporte que permite o cascadeamento das interrupções é justamente o sinal alto que passa por cada uma das fontes, em ordem decrescente.

6 .

Caracterize processo (0.5). Descreva os estados em que um processo possa um diagrama (grafo) para ilustrar a transição entre eles (1.0). Um escalonador é responsável por definir o processo a ser executado no momento. Qual o algoritmo de escalonamento é mais adequado? Porque? Descreva como esse algoritmo funciona e como pode contribuir para o objetivo geral de aumentar o throughput. (1.0).

R: