

Lista de Exercícios

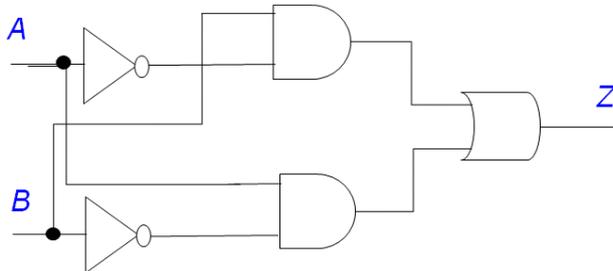
Organização de Computadores e Sistemas Operacionais

Sistemas de Informação CIn/UFPE

Prof. Sergio Cavalcante

Conceitos Gerais

1) Qual a tabela verdade do circuito abaixo?



2) Faça a subtração a seguir da forma como o computador faz: 54 – 23.

⊛ Subtração binária (54-23)

① Conversão da base decimal p/ binária

- Alternativa 1:
 $54 = 00110110$
 $23 = 00010111$

Completear os valores com 0 até 8 bits

- Alternativa 2:
 $54 = 00110110$
 $23 = 00010111$

Pega o quociente da última divisão e o resto das anteriores

② Transformação do 2º valor em negativo

- Inverte seus valores
 $23 = 00010111$
 \downarrow
 11101000
- Soma (+1)
 11101000
 $+1$
 $\hline -23 = 11101001$

③ Soma dos valores

$54 - 23 = 31$

$$\begin{array}{r} 00110110 \\ +11101001 \\ \hline 100011111 \end{array}$$

- 3) Qual o resultado de 01100110×01110110 ?
- 4) O que é um sistema operacional?
R.: É um gerenciador de recursos do computador que fornece uma interface amigável com o usuário
- 5) O que acontece quando ligamos o computador?
R.: 1. A CPU vai ao endereço inicial da memória principal (onde está o BIOS) e executa o POST, que testa todos os periféricos.
2. A CPU executa o Boot Loader, que vai ao setor 0 (zero) do disco buscar o Loader do S.O e o carrega na memória principal.
3. A CPU executa o Loader do S.O. que carrega o S.O. na memória principal e passa a executá-lo.
- 6) O que acontece quando ligamos um computador que tem dois sistemas operacionais como opção?
R.: 1. A CPU vai ao endereço inicial da memória principal (onde está o BIOS) e executa o POST, que testa todos os periféricos.
2. A CPU executa o Boot Loader, que vai ao setor 0 (zero) do disco buscar o software de seleção do S.O e o carrega na memória principal.
3. O usuário seleciona o S.O. e
3. A CPU executa o Loader do S.O. que carrega o S.O. na memória principal e passa a executá-lo.
- 7) O que é uma interrupção? E uma exceção?
R.: Interrupção é uma forma de um dispositivo de E/S chamar a atenção da CPU / S.O.. É algo desejado e que facilita a comunicação com os dispositivos (evita, por exemplo, o "busy-waiting"). Exceção é algo indesejado provocada por algum erro na execução de um processo. É gerada dentro da CPU e pode ocasionar erros fatais, que encerram a execução de um processo de forma abrupta.
- 8) O que acontece com o processo que está sendo executado quando ocorre uma interrupção ou uma exceção?

Processos:

- 9) Como um computador com um único processador faz para executar vários programas ao mesmo tempo? Explique como isso é feito.
- 10) O que é um escalonador?
R.: É uma parte do S.O. responsável por definir a ordem de execução dos processos baseado no algoritmo de escalonamento.
- 11) O que é e para que serve o contexto de um processo? O que ele contém?
R, O Contexto de um processo é o conjunto de informações necessárias para retomar a execução à partir do ponto em que este parou. Este também contém informações de contabilidade como, por exemplo, o tempo de uso da CPU pelo processo. O Contexto contém contador de programa, registradores, estado, pilha, espaço de endereçamento, arquivos abertos, processos filhos, informações de uso do contabilidade e uso do sistema.
- 12) Como o escalonador garante que um processo não usará o processador para sempre?
R.: Antes de colocar um processo em Rodando, o escalonador ajusta o temporizador para interromper a execução daquele processo caso passe o tempo limite permitido para a sua execução. Se essa interrupção ocorrer, a CPU põe o escalonador para escolher o próximo processo para executar.
- 13) O que é um processo? O que são threads? Quais partes do contexto das threads são compartilhados e quais não são?
R. Um processo é um programa em execução. Threads são partes de um processo que são

escalonadas individualmente. As partes do contexto que são compartilhadas são: espaço de endereçamento, arquivos abertos, processos filhos, informações de uso do contabilidade e uso do sistema. Não são compartilhados: contador de programa, registradores, estado e pilha.

14) Como se garante que um processo não invada o espaço de endereçamento de outro processo? Na segmentação, há alguma diferença?

R.: Através da comparação dos endereços gerados pela CPU com seus registradores Base e Limite, que definem o espaço destinado ao processo em execução. Se o endereço gerado for menor que a Base ou maior que o Limite, uma exceção é gerada e o processo é encerrado. Na segmentação, cada segmento (código, dados, pilha, etc) tem seus próprios registradores Base e Limite, aumentando a proteção entre os segmentos do mesmo processo.

15) O que ocorre quando não há nenhum processo do usuário no estado “Rodando”?

R.: O Escalonador põe o Processo Ocioso do Sistema para rodar.

16) Quais são as condições para que um processo esteja em cada um dos estados do escalonamento?

Rodando: O processo está em execução pela CPU

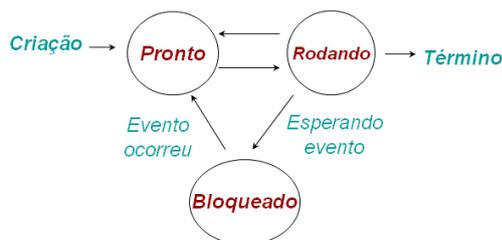
Pronto: O processo está aguardando que o escalonador o coloque em execução

Bloqueado: O processo está esperando o término de algum evento de Entrada/Saída para ir para o estado Pronto.

17) O que faz com que um processo deixe o estado “Rodando” direto para o estado “Pronto”?

18) Em que estado(s) do escalonamento é possível não ter nenhum processo? Por quê?

- a. No estado Bloqueado, caso nenhum processo esteja esperando por Entrada/Saída, e no Pronto, quando todos estão bloqueados esperando por Entrada/Saída.



Gerenciamento de Memória:

19) O que são e quais as diferenças entre memória principal e secundária? Dessas duas, qual é volátil e qual é não-volátil?

20) O que é e para que serve “memória virtual”?

21) O que é e para que serve a Hierarquia de Memória?

22) Descreva o Princípio da Localidade e sua importância.

23) O que é memória RAM, ROM, PROM (OTP), EPROM, EEPROM e Flash EPROM?

24) Como um processo pode ser relocado na memória principal?

25) O que é Espaço de Endereçamento?

26) Para que servem os registradores Base e Limite? Onde eles ficam?

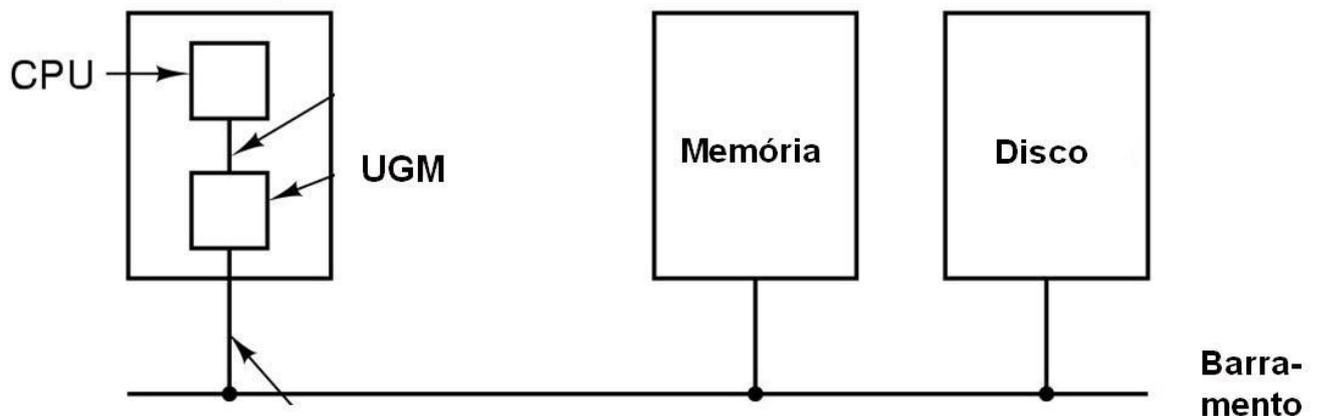
27) O que é paginação e o que é segmentação? Quais as características de cada uma?

28) Como uma informação da memória principal é passada para a CPU usando paginação? Explique o mecanismo usando a figura abaixo e considere que pode haver a necessidade de fazer a substituição de páginas.

R.: Passo 1. A CPU gera um endereço virtual que é enviado a UGM. Este endereço é dividido em Endereço da Página Virtual – EPV e Endereço do Caracter na Página – ECP. A UGM verifica se a informação de tradução do EPV para Endereço de Página Real – EPR se encontra no TLB, que contém as informações de tradução mais recentes. Se estiver, o endereço real EPR|ECP é usado pela UGM para buscar o dado na memória principal e entregar à CPU.

Passo 2. Se o TLB não contém a informação de tradução, a UGM acessa a Tabela de Páginas na memória principal e verifica se a informação de tradução está lá. Se estiver, a UGM atualiza o TLB e executa o Passo 1 novamente.

Passo 3. Se a informação não estiver na Tabela de Páginas, a UGM gera uma interrupção para a CPU que passa a executar o algoritmo de substituição de páginas, geralmente o NRU – Não Recentemente Usado, que irá copiar a página requisitada do disco para a memória principal, atualizar a Tabela de Páginas e a TLB e executar o Passo 1 novamente



Dispositivos de Entrada e Saída

29) Como é possível instalar um dispositivo que não era conhecido quando o sistema operacional foi projetado?

30) O que é um driver de dispositivo e para que serve?

R.: É um software que é parte do S.O. e é responsável por fazer a comunicação entre a Camada Independente de Dispositivo (ou Subsistema de E/S) e o Controlador de Dispositivo. Ele traduz comandos de alto nível para o controlador, corrige e informa erros, e pode ser instalado a qualquer momento.

31) O que são e para que servem Controladores de Dispositivos?

R.: São hardwares que controlam portas, barramentos e dispositivos, executando comandos de E/S.

32) Descreva barramentos e portas.

R.: Barramentos são conjuntos de condutores elétricos que fazem a comunicação de vários dispositivos. Portas fazem apenas a comunicação entre dois dispositivos e, por serem mais simples, têm protocolos de comunicação também mais simples.

33) Explique para que serve e como são implementados a E/S isolada e a E/S mapeada em memória.
 R.: Na E/S Isolada a forma de acesso aos dispositivos de E/S é diferente do acesso à memória. Necessita de instruções especiais que apenas o S.O. pode executar, impedindo que processos do usuário tenham acesso direto aos dispositivos de E/S, promovendo maior segurança ao sistema.

Na E/S Mapeada em Memória a forma de acesso aos dispositivos e à memória é igual. Tem a vantagem de ter uma implementação mais simples e barata, mas não garante a proteção da E/S Isolada.

34) Descreva E/S programada, E/S por interrupção e E/S por DMA.

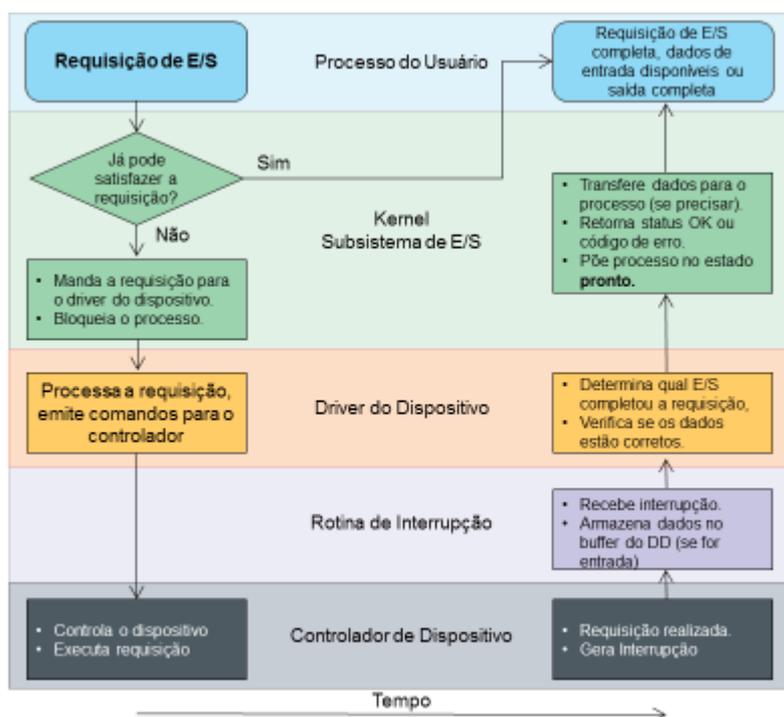
R.: Na E/S Programada um programa fica em loop perguntando ao dispositivo se pode enviar/receber o próximo dado até que tenha enviado/recebido todos. É extremamente ineficiente.

Na E/S por Interrupção, o S.O. envia/recebe um caracter do dispositivo e passa a executar outras atividades. Quando o dispositivo estiver pronto para outra comunicação, este gera uma interrupção fazendo com que o S.O. envie/receba o próximo caracter. Isto continua até que todos os caracteres tenham sido transmitidos.

Na E/S por DMA, um Controlador de DMA é programado para fazer toda a transferência de dados entre o dispositivo e a memória. Quando terminar, o Controlador de DMA gera uma interrupção que informa ao S.O. que terminou.

35) Quais as características que um driver de dispositivo informa para se conectar a um Sistema Operacional?

36) Diga o que acontece quando um processo faz uma chamada de E/S. Mostre usando as camadas do Sistema Operacional.



Exemplo de Chamada de E/S (2)

Discos e Memória Secundária:

37) Descreva como funciona e as características dos RAIDs 0, 1, 4 e 5.

115

7/R: os Raids são feitos para deixar o sistema mais rápido e eficiente.

Raid 0 apresenta as seguintes características:

- Não tem redundância;
- Os dados são distribuídos a nível de blocos entre discos;
- Alta taxa de transferência;
- Capacidade de acesso em paralelo.

caso →	A	B	C	D
	bloco 0	bloco 1	bloco 2	bloco 3
	bloco 4	bloco 5	bloco 6	bloco 7
	bloco 8	bloco 9	bloco 10	bloco 11
	bloco 12	bloco 13	bloco 14	bloco 15

Raid 1 apresenta as seguintes características:

- Tem dados duplicados em dois discos;
- Muito caro (custoso);
- Alta confiabilidade;
- Alta Redundância.

Disco A	B	C	D
bloco 0	bloco 1	bloco 0	bloco 1
bloco 2	bloco 3	bloco 2	bloco 3
bloco 4	bloco 5	bloco 4	bloco 5
bloco 6	bloco 7	bloco 6	bloco 7

Raid 4 Apresenta as seguintes características:

- Os dados são distribuídos ao nível de bloco entre os discos;
- permite o uso de um disco com blocos de paridade;
- Alta taxa de transferência;
- vários acessos permitidos por vez;
- gargalo na escrita dos discos da paridade.

Disco A	B	C	D	E
bloco 0	bloco 1	bloco 2	bloco 3	Par. 0-3
bloco 4	bloco 5	bloco 6	bloco 7	Par. 4-7
bloco 8	bloco 9	bloco 10	bloco 11	Par. 8-11
bloco 12	bloco 13	bloco 14	bloco 15	Par. 12-15

Obs: Par. significa paridade.

Raid 5 Apresenta as seguintes características:

- Os dados são distribuídos ao nível de bloco entre os discos;
- os blocos de paridade são distribuídos entre discos;
- Alta taxa de transferência;
- vários acessos permitidos por vez;
- Elimina o gargalo na escrita dos discos da paridade.

Disco A	B	C	D	E
bloco 0	bloco 1	bloco 2	bloco 3	Par. 0-3
bloco 4	bloco 5	bloco 6	Par. 4-7	bloco 7
bloco 8	bloco 9	Par. 8-11	bloco 10	bloco 11
bloco 12	bloco 13	bloco 14	bloco 15	

38) O número binário 0_000101 tem paridade 1. Qual o valor do bit que está faltando? Como você sabe o valor?

39) Se implementarmos um RAID 4 com apenas dois discos, qual tipo de RAID ele se torna? Por quê?
R.: Se torna o Raid 1, porque os blocos de paridade serão idênticos aos blocos de dados, visto que paridade de 0 é 0 e paridade de 1 é 1.

40) Se implementarmos um RAID 5 com apenas dois discos, qual tipo de RAID ele se torna? Por quê?
R.: Se torna o Raid 1, porque os blocos de paridade serão idênticos aos blocos de dados, visto que paridade de 0 é 0 e paridade de 1 é 1.

Programação:

Introdução à Linguagem C

- 1) Faça um programa para ler um número do usuário e determinar se este número é par ou não par. Imprima o resultado.
- 2) Faça um programa que recebe 3 valores inteiros do usuário e armazena nas variáveis **a**, **b** e **c**. Com esses 3 números calcule o MMC dos 3 valores. Imprima o resultado.
- 3) Faça um programa que recebe 3 valores inteiros do usuário e armazena nas variáveis **a**, **b** e **c**. Faça uma função "**maior**" que recebe três valores e retorna o maior valor entre eles. Na função "**main**", use a função "**maior**" para calcular o maior valor entre as variáveis **a**, **b** e **c**, e imprima o resultado.
- 4) Implemente um programa que solicita e recebe valores do usuário, calcula a função $v = v0 + a*t$, e apresenta o resultado na tela. Faça um programa que:
 - a. Solicite ao usuário um número inteiro entre 100 e 999 usando a sentença:
"Entre com um valor entre 100 e 999:"
 - b. Caso o valor não esteja entre 100 e 999, **o programa deve apenas apresentar a mensagem:**
"O valor deve estar entre 100 e 999".
 - c. Caso contrário, o programa imprime o número recebido escrito ao contrário.
Ex: recebe 431 e imprime 134.
- 5) Faça um programa que calcule de forma recursiva o valor X^Y , onde x e y são inteiros positivos.
- 6) Escreva uma função CALCULA que:
 - a. Receba como parâmetros duas variáveis inteiras, X e Y;
 - b. Retorne em X a soma de X e Y;
 - c. Retorne em Y a subtração de X e Y.
- 7) Faça um programa que tenha a função *HorarioParaSegundos*, que recebe 3 parametros: *horas*, *minutos* e *segundos* e retorna o numero de segundos total. Na função *main*, use a função *HorarioParaSegundos* para calcular o numero de segundos existentes entre dois horários fornecidos pelo usuário e imprima o resultado.
- 8) Faça um programa que, dados dois vetores unidimensionais $A = (2, 4, 35, 50, 23)$ e $B = (17, 9, 12, 27, 5)$, calcule e imprima:
 - a. a soma vetorial dos elementos destes vetores
 - b. o maior numero entre todos os elementos dos dois vetores