Tarefa Avaliativa de Sistemas Operacionais 2º Bimestre 01
Aluno: Pedro Beethoven da Costa
Matrícula: 20181014040015

1- Uma vez detectado um impasse, as abordagens para resolver são as seguintes:
Eliminar tarefas: uma ou mais tarefas envolvidas no impasse são eliminadas, liberando seus recursos para que as demais tarefas possam prosseguir. A escolha das tarefas a eliminar deve levar em conta vários fatores, como o tempo de vida de cada uma, a quantidade de recursos que cada tarefa detém, o prejuízo para os usuários que dependem dessas tarefas, etc.
Retroceder tarefas: uma ou mais tarefas envolvidas no impasse têm sua execução parcialmente desfeita (uma técnica chamada rollback), de forma a fazer o sistema retornar a um estado seguro anterior ao impasse. Para retroceder a execução de uma tarefa, é necessário salvar periodicamente seu estado, de forma a poder recuperar um estado anterior quando necessário.

2- a) t1 ---> r2 -> t2 ----> r1 -> t1
b) t1 ---> r2 -> t2 ---> r3 -> t3 ---> r1 -> t1 (Considerando que t4 não libera uma das instâncias de r3 pra t2, consequentemente gerando um impasse. Caso contrário, não ocorrerá impasse.).

3 - Exclusão mútua: Os carros em sentidos diferentes não consegue seguir seu curso ao mesmo tempo.
Posse e espera: Os carros solicitam acesso ao cruzamento sem liberar o seu próprio cruzamento ocupado.
Não-preempção: Os carros só liberam o espaço do seu cruzamento se assim decidir.
Espera circular:O ciclo está definido, pois a liberação dos carros azuis, dependem da liberação do cruzamento dos carros verdes, que por sua vez depende da liberação dos amarelos, que dependem dos vermelhos, que esperam pela liberação dos azuis, fechando o ciclo de espera.
Regra de resolução: definir semáforos em cada um dos sentidos, possibilitando o prosseguimento das tarefas em um sentido por vez.

4 - Codificação: Nessa etapa o programador que escolhe em qual endereço será armazenado as variáveis e o código do programa. Normalmente, isso ocorre quando é feito em uma linguagem de mais baixo nível, como Assembly.
Compilação: O compilador realiza no processo de compilação (tradução) do código-fonte, a escolha da posição das variáveis na memória.
Ligação: Durante a etapa anterior, é gerado um arquivo de saída ***object file*** que contém o código binário e uma tabela de símbolos descrevendo as variáveis e funções usadas, seus tipos, onde estão definidas e onde são usadas. Em seguida, o ***linker*** pega esses arquivos e suas tabelas que foram geradas, define os endereços de memória dos símbolos e gera um arquivo executável.
Carga: Nessa etapa, o carregador (**loader**) é o responsável por carregar o código do processo na memória e definir os endereços de memória que devem ser utilizados.
Execução: Os endereços que o processador gerou são analisados e convertidos nos endereços efetivos que serão acessados na memória real.

5- TEXT: contém o código binário a ser executado pelo processo, gerado durante a compilação e a ligação com as bibliotecas e armazenado no arquivo executável.
DATA: esta seção contém as variáveis estáticas inicializadas, ou seja, variáveis que estão definidas do início ao fim da execução do processo e cujo valor inicial é declarado no código-fonte do programa.
BSS: historicamente chamada de Block Started by Symbol, esta seção contém as variáveis estáticas não-inicializadas.
HEAP: esta seção é usada para armazenar variáveis alocadas dinamicamente, usando operadores como malloc(), new() e similares.
STACK: esta seção é usada para manter a pilha de execução do processo, ou seja, a estrutura responsável por gerenciar o fluxo de execução nas chamadas de função e também para armazenar os parâmetros, variáveis locais e o valor de retorno das funções.

6 - Endereços físicos: São os endereços dos bytes de memória física do computador.
Endereços lógicos: São os endereços de memória usados pelos processos e pelo sistema operacional.
O uso dos endereços lógicos são para simplificar os procedimentos de alocação de memória, pois durante a execução de processos, o processador gera endereços lógicos que são utilizados para facilitar o acesso a memória, que logo em seguida serão traduzidos para os endereços físicos. Além de que a sua utilização implica na proteção dos processos, já que é realizado um processo de tradução dos endereços (lógico para físico), tornando-o mais seguro e estável.

7 -
0:45 - 89
1:100 - 300
2:90 - 90
3:1.900 - Violação de segmento (estouro do limite)
4:200 - 1.400

8-

Página	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Quadro	3 - 4.000	12	6	-	9	741	2 - 1.995 - 6.633	-	0	5	-	1995	-	7	414	1

9 - Resposta: B

10- Falta de página é uma interrupção (ou exceção) disparada pelo hardware quando um programa acessa uma página mapeada no espaço de memória virtual, mas que não foi carregada na memória física do computador; As causas possíveis são: a página correspondente ao endereço requisitado não está carregada na memória; a página correspondente ao endereço de memória acessado está carregada, mas o seu estado

corrente não foi atualizado no hardware; O sistema operacional deve tratar da seguinte maneira: verifica se a página é válida, usando os flags de controle da tabela de páginas. Caso a página seja inválida, o processo tenta acessar um endereço inválido e deve ser abortado. Por outro lado, caso a página solicitada seja válida, o processo deve ser suspenso enquanto o sistema transfere a página de volta para a memória RAM e faz os ajustes necessários na tabela de páginas. Uma vez a página carregada em memória, o processo pode continuar sua execução.