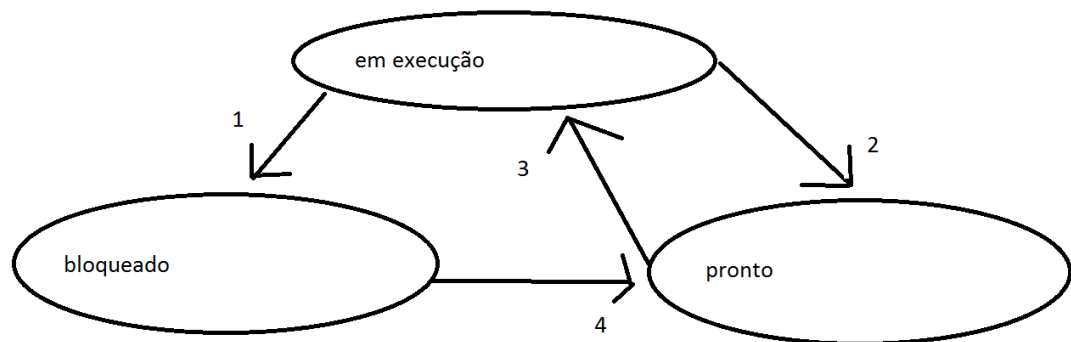


Respostas do exercício de revisão (atividade 5) de sistemas operacionais 1

- 1) As principais diferenças entre os sistemas monotarefa e multitarefa são: a monotarefa executa apenas uma única vez um determinado programa por meio do scanf, já o sistema multitarefa e o que permite o uso de mais de um programa ao mesmo tempo, por meio do time-sharing e é uma variação da multiprogramação, por onde há uma interação maior entre os usuários e agilidade nas respostas citadas.
- 2) Processo e um programa em execução e ao mesmo tempo e uma entidade ativa no sistema com a execução de diversos eventos para o uso como dispositivos de entrada e saída, execução de programas e outros, já que o programa e algo passivo.
- 3) O processo esta no diagrama abaixo:



na primeira execução: o processo bloqueia aguardando uma entrada, por onde o mesmo sai da area de execução e estara sendo bloqueado, já na segunda execução e o escalonador seleciona outro processo por onde este processo vai ficar pronto e de la vai para o passo 3, na terceira execução e realizado outro processo de escalonador e dentro disso e feito a seleção deste mesmo processo, por fim na quarta e ultima execução a entrada se teorma disponivel devido ao fato de sair de bloqueado para pronto.



Já o segundo processo, mostra como e a execução mais detalhada do mesmo, com um nova admissão e depois em pronto e por lá e feito o despacho do agenda dor para

rodar, assim passa a ter dois caminhos, o primeiro que é o de cima e interromper o processo e volta para onde estava e o segundo é a espera por evento de entrada/saída, dentro desta espera é feito a execução do programa e ao sair da espera é encerrada a conclusão do evento de entrada e saída e ambos voltam a se encontrando junto até o rodando, assim eles saem já com o processo encerrado.

- 4) PCB ou process control block, que na tradução livre é bloco de controle de processos e por onde é gerada as informações sobre gerenciamento de memória (registrar dados referentes a base e o limite, bem como as tabelas de página ou seguimento), compatibilidade (a quantidade de CPU e o tempo de leitura utilizado, limites de tempo e o número da tarefa) e também sobre o status do sistema como dispositivos de entrada e saída por meio de uma lista, lista de arquivos abertos e outros.
- 5) A troca do contexto é quando salva as variáveis atuais do processo e a restauração de variáveis do processo diferente, e é representado pela PCB no processo além do overhead, já o tempo de troca causado pelo overhead é transcrito pelo fato de o sistema não realizar trabalho útil enquanto realiza a troca e depende da CPU como o hardware, barramentos e a memória.
- 6) Os escalonadores são processos que se migram em diferentes filas e neste caso são divididos em três situações, as de curto, meio termo e longo prazo. As de curto prazo são as que pegam um processo e as executam em seguida e depois disso são alocados na CPU num prazo muito mais rápido, já as de longo prazo são as que são remanejados na fila de pontos e depois possuem a execução por baixa frequência, além disso os mesmos possuem a taxa média de criação igual a taxa de saída média dos processos que deixam o sistema e possuem mais tempo para decidir e neste caso há dois processos o I/O-bound e o CPU-bound, primeiro diz que os escalonadores são de curto prazo e a fila fica vazia, já o segundo é a fila de entrada e saída vazia e o uso de poucos dispositivos. Já os de meio termo são as que reduzem o grau de multiprogramação com os seguintes argumentos: de ser mais vantajoso remover processos de memória e da disputa pela CPU, que poderão ser reintroduzidos depois e é chamado de swapping.
- 7) A comunicação entre os processos é realizada da seguinte forma, por meio de um sistema que combine o cliente com o servidor e por conta disso, pode ser realizado como canal de comunicação para a execução de determinadas tarefas conjuntas.
- 8) Thread é uma unidade básica de utilização da CPU para a execução correta de programas numa subdivisão, e apelidado de processos leves, e uma ID de Thread com um contador de programa, um conjunto de registradores e uma pilha.
- 9) As vantagens dos threads nos processos são: possuem as mesmas aparências de um software, compartilha o mesmo espaço de memória, mas com o contexto do hardware diferente e os principais ganhos são a responsividade no caso de houver travamento em código, compartilhamento de recursos com várias execuções diferentes dentro do mesmo espaço de endereços, economia com realocação de recursos e utilização de arquiteturas de MP com mais de um processador.
- 10) Seção crítica é aquela que o código acessando recursos compartilhados, ou seja é a entrada para uma área por onde há compartilhamento de dados e lá onde ficam os mesmos, seção restante é aquela que código acessando a recursos exclusivos, em outras palavras é um termo que o mesmo possui acesso a uma área restrita do sistema, seção de entrada é onde o código que permite a entrada na seção crítica,

sendo assim a seção de entrada e uma porta de embarque para a seção crítica e lá há os dados compartilhados e por fim seção de saída e onde o código executado após a saída da seção crítica relacionando ao acesso da seção crítica, que em outras palavras é a porta de desembarque realizando assim o término do acesso a seção crítica.

- 11) O processo dentro da seção crítica é constituído de duas formas, a primeira é quando não há execução de processos em sua seção crítica e há o processo de seleção dentro da sua seleção e assim não pode adiar indefinidamente, já o segundo item condiz que a participação ocorre quando não estão numa seção restante.
- 12) Estes dois códigos (em C) possuem muitas semelhanças entre si, mas também diferenças, entre elas a principal é a troca de valores binários (0 e 1) em ambos os códigos, pois no primeiro a expressão `while (turn != 0);` e em `turn = 1;`, já no código ao lado e ao contrário com expressão `while (turn != 1);` e em `turn = 0;` dando a impressão que os valores estejam trocados, que na prática é um efeito comboio (vários dados em sequência).
- 13) Neste código é feita a execução de um processo em C para o Linux, e dentro disso a função `pid_t pid;` em destaque no código é uma variável apenas no nome, mas complementa todo o código para a sua execução, fora disso, a saída do mesmo resultará em uma série de eventos realizados no computador (ou máquina virtual) no exato momento da execução deste programa. E fora disso, são considerados processos pai e filho devido ao uso de elementos como `exec()` e `fork()`, que de forma invertida serve para substituir um determinado espaço na memória, e o pai pode criar mais filhos neste processo, após o término do primeiro, e feita a exclusão, mas em determinadas situações pode haver o aborto (pai encerrar o processo filho), término em cascata (pai primeiro e o filho depois) e outras situações.
- 14) Sim, pois no diagrama todos os ciclos em azul recebem instruções dos retângulos verticais cinzas (menos o de número quatro, pois o mesmo não recebe) e com isso há um deadlock no diagrama.