

1. c) O Sistema Operacional funciona como uma interface entre o usuário e computador.
2. a) (F) A CPU ou UCP é um dispositivo lógico que controla o Sistema Operacional.
b) (V) Desfragmentador é um tipo de software utilitário.
c) (V) Dispositivos I/O são dispositivos de entrada e saída.
d) (V) A memória principal é uma unidade funcional do tipo volátil.
3. a) (F) Um Sistema Operacional não permite as pessoas utilizarem o hardware do computador.
b) (V) Sistemas Operacionais Monotarefas executam um aplicativo de cada vez.
c) (F) Nos Sistemas Operacionais Multitarefa o gerenciamento de memória é simples.
d) (V) Os sistemas com múltiplos processadores caracteriza.
4. Um computador sem um sistema operacional pode rodar programas desde que o programa esteja executando o *boot*. Um computador conhecido que não possuía sistema operacional é o ENIAC, onde o programa era feito através de 3.000 *switches* com conexões feitas por centenas de cabos. Então um sistema operacional torna o computador mais fácil de usar, pois suas duas principais funções são de controle de recursos e abstração de partes complexas. O controle de recursos controla o tempo e espaço que cada programa irá ter. E a abstração da complexidade é tornar o sistema mais fácil de ser programável. Assim o acesso aos recursos do sistema fica mais fácil e o compartilhamento dos recursos se mantém organizado.
5. A vantagem de sistemas que operam em rede é que há a possibilidade de compartilhamento. Por exemplo, cada sistema pode compartilhar suas impressoras e seus dados com os outros sistemas da rede. Também os dados

podem ser replicados por essa rede, gerando maior segurança dos casos caso algum nó da rede falhe. Mas se não houver replicação de dados e um nó falhar, esta pode ser uma desvantagem da rede. Computadores *stand-alone* são computadores sem qualquer acesso a rede local ou internet, de forma que todos os recursos estão nele. A desvantagem dele é a falta de segurança dos dados, não há replicação ou compartilhamento.

6. Máquina virtual é um ambiente, geralmente um *software* ou sistema operacional que não existe fisicamente, mas é criado por um outro ambiente. Neste contexto a VM (*Virtual Machine*, em inglês) é chamada de *guest* (convidado, em inglês) enquanto o ambiente que a criou é chamado de *host*. Geralmente as VMs são criadas para executar um conjunto de instruções diferente do que a do ambiente do *host*. O *host* controla os recursos para cada máquina virtual criada, geralmente atribuindo dinamicamente para cada uma. Um exemplo de VM é a JVM (*Java Virtual Machine*) que interpreta comandos específicos do Java. Ela executa um código chamado de *bytecode* abstraindo os recursos para este *bytecode*. A JVM não depende de conjunto de instruções ou das APIs específicas de cada sistema operacional. Ela virtualiza todos os recursos para o *bytecode*. O usuário que interage com um servidor virtualizado o vê como uma máquina física, no sentido de que ele consegue acessar os recursos da máquina como discos rígidos, memória RAM, processadores e conexões Ethernet. Na verdade tudo isso é virtual. Por exemplo, ao invés de acessar o disco rígido real, o usuário acessa um disco criado pelo ambiente *host*. Este disco que irá acessar o disco rígido real. A vantagem disto é que as VMs podem ser ambientes diferentes do *host*, facilitando testes, aumentando a segurança, melhor uso do hardware e também gera economia. Com é possível definir um próprio hardware para cada máquina virtual, não é necessário gastar dinheiro em uma máquina física para cada sistema.

7. • **Sistema operacional de tempo-real** Sistema usado para controle de

máquinas, instrumentos científicos e sistemas industriais. Possui uma interface com o usuário bem pequena, pois não é um sistema flexível. A parte mais importante é a de gerenciamento de recursos, todas as operações devem ser executadas em um tempo pré-definido sempre que forem executadas. Se a operação for executada mais rápida ou mais lenta, será caracterizado como uma falha.

- **Single-user, single task** Sistema desenvolvido para que haja apenas um usuário executando apenas uma tarefa por vez. Um exemplo é o Palm OS, sistema para computadores embarcados.
- **Single-user, multi-tasking** É o tipo de sistema mais comum para usuários com laptops ou desktops. Microsoft Windows, Mac OS são exemplos de plataformas que permitem um usuário executar mais de uma tarefa simultaneamente.
- **Multi-user** Um sistema multi-usuário permite que vários usuários tomem vantagem dos recursos do computador simultaneamente. O sistema deve se certificar que a divisão de recursos para cada usuário esteja balanceada. Unix, VMs e sistemas de mainframe são exemplos de sistemas operacionais de multi-usuário.

8. Ele tem a característica de não ter interação do usuário com o programa. A entrada e saída de dados do programa é implementada para utilizar dados em uma memória secundária, como arquivos em um disco rígido. Exemplos de programas que fazem processamento batch são programas de cálculo numérico, compiladores, programas de ordenação e backups. Nenhum destes exemplos exige a interação do usuário.
9. Sistemas de tempo compartilhado dão a liberdade de um programa executar por intervalos pequenos logo deixando outro programa executar por um intervalo pequeno. Este método de executar um pouco de cada programa dá a sensação ao usuário que o ambiente está executando todos os processos simultaneamente. Enquanto que no sistema de tempo compartilhado quanto

mais rápido executar melhor será a experiência do usuário, em sistemas de tempo real o que importa é a pontualidade. Aplicações industriais e bélicas são indicadas para sistemas de tempo real, como sistema de controle de produção e robôs enviados para outros planetas.

10. São sistemas que possuem processadores interligados trabalhando em conjunto. A vantagem é poder dividir o mesmo programa entre os processadores ou dividir cada processador para um programa.
11. Em sistemas fortemente acoplados os recursos são compartilhados entre todos. Nos fracamente acoplados não há compartilhamento e cada sistema precisa fazer o controle independentemente dos outros sistemas.
12. São sistemas interligados através de linhas de comunicação com cada um fazendo de forma independente o gerenciamento de recursos como processadores, memória e dispositivos de entrada e saída.
13. Em sistemas operacionais de rede cada nó pode compartilhar seus recursos, como impressores e arquivos com os outros nós da rede. Nos sistemas distribuídos, o sistema irá esconder detalhes dos nós individuais, fazendo com que seja tratado como um conjunto único e um sistema fortemente acoplado.
14. Processador, memória principal e dispositivos de entrada e saída.
15. O processador é formado de uma Unidade de Controle (UC), uma Unidade Lógica e Aritmética (ULA) e de Registradores. A UC gerencia as atividades dos componentes do computador, como leitura e escrita de dados. A ULA realiza operações lógicas e aritméticas (comparações, somas e subtrações).
16. A memória principal é em geral um dispositivo de armazenamento volátil, que guarda dados e instruções utilizados pelo processador na execução de programas. A memória secundária não é volátil, possui maior capacidade de armazenamento mas é mais lenta para acessar os dados armazenados.

17. Dispositivos de entrada e saída são dispositivos usados para entrada de dados para um programa processar e saída de dados com os resultados do processamento.
18. RISC possui um conjunto de instruções pequeno que dá facilidade para ser processado, pois o objetivo é que cada instrução seja executado em um ciclo de clock. CISC tem um conjunto de instruções maior com a finalidade de facilitar a vida do programador, com instruções que fazem o trabalho mais difícil. Essas instruções tomam mais ciclos de clock para serem executados. Os programas para CISC são em geral bem menores, mas não mais rápidos que os RISC. Atualmente os processadores domésticos são desenvolvidos com o melhor de cada arquitetura, criando uma arquitetura mista de RISC e CISC.