DIREÇÃO CONCURSOS

CURSOS ONLINE DOMINANDO TI

Aula 01 – Infraestrutura de TI

Sistemas Operacionais - Técnico Judiciário TRF3 - Informática

Prof. Rômulo Santos

Sumário

SUMÁRIO. 2

APRESENTAÇÃO. 3

SISTEMAS OPERACIONAIS. 5

O QUE É UM SISTEMA OPERACIONAL? 5
FUNÇÕES DOS SO. 11
CLASSIFICAÇÃO DOS SO. 13
CONCEITOS DE PROCESSOS E THREADS. 19

RESUMO DIRECIONADO. 24

QUESTÕES COMENTADAS PELO PROFESSOR. 30

REFERÊNCIAS. 40

Apresentação

Olá concursando, é com imensa satisfação que anunciamos essa grande parceria entre o Direção Concursos e o Dominando TI. Caso ainda não me conheça, sou o professor Rômulo Santos e leciono disciplinas ligadas à área de Tecnologia da Informação, como ênfase em Microinformática, Sistemas de Computação, Arquitetura e Organização de Computadores, Sistemas Operacionais, Linux, Infraestrutura de TI, *Datacenters*, Bancos de Dados, Redes de Computadores, SGBD Oracle, dentre outras. Também me dedico a um Programa de Estudos de "Alto Rendimento para Concursos" que por meio de sucessivas evoluções, vem alcançando aprovações e resultados extremamente satisfatórios. Meu objetivo principal será o direcionamento da sua preparação em concursos públicos visando a obtenção de resultados ainda mais eficientes.

Minha trajetória acadêmica iniciou com a bacharelado em Análise de Sistemas (UFMS). Logo, em seguida conclui algumas especializações aos quais destaco Criptografia e Segurança em Redes (UFF), Aplicações Complementares às Ciências Militares (EsFCEx), Gestão de Tecnologia da Informação na Administração Pública (JK) e Ciências Militares (EsAO). Tive a grata felicidade de defender o Mestrado em Ciência da Informação (Unb), e atualmente sou doutorando em Gestão de Projetos de TIC (UNINI).

Já na área profissional puder exercer várias funções ligadas à área tecnológica. Atualmente sou gerente de TI no Exército Brasileiro, instituição em que ingressei no ano de 1997. Tenho experiência na área de Ciência da Informação, com ênfase em Arquitetura da Informação, Organização e Recuperação da Informação e na área de Ciência da Computação, com ênfase em Metodologias e Técnicas da Computação, Gestão de Projetos de TI, Gestão de Equipes de Desenvolvimento de *Software*, Infraestrutura de *Datacenters*, Criptografia, Segurança da Informação, Bancos de Dados, *Business Intelligence* e *Data Warehouse*. Participei de projetos envolvendo desenvolvimento de Sistemas Corporativos, Banco de Dados, Certificação Digital, Telefonia VoIP, Segurança da Informação, BI, OLAP, ETL e DW.

Além disso, consegui algumas certificações de mercado, entre as quais, a *Microstrategy Certified Project Designer 9.2.1* (*Microstrategy University*), *Oracle Certified Associate (OCA) - Oracle Database Administrator* (*Oracle University*), *Oracle Certified Professional (OCP) - Oracle Database Administrator* (*Oracle University*), *Oracle Certified Expert (OCE) - Oracle Real Application Clusters Administrator* (*Oracle University*).

Minha experiência em docência iniciou-se em 1998, no Exército Brasileiro, onde pude contribuir por 12 anos na capacitação, preparação e formação técnico-profissional de milhares de jovens. Nesse período pude ser condecorado com as comendas da "Ordem do Mérito Thomaz Coelho" e da "Ordem do Mérito Marechal Trompowsky", ambas do Instituto dos Docentes do Magistério Militar. Desde 2010, leciono em cursos preparatórios para concursos públicos, nas modalidades *online* e presencial. Encontrei nessa atividade uma satisfação profissional jamais desfrutada.

Agora que nos conhecemos podemos passar para o mais interessante, ou seja, falar sobre esse nosso **curso**.

Você está iniciando os estudos em TI para concursos? Se sim, isso não será um problema. O curso o atenderá perfeitamente, uma vez que, será abordada "em detalhes" toda a teoria necessária para sua preparação, assim como, a resolução comentada de inúmeros exercícios para que você possa entender e praticar bastante cada aspecto estudado. E caso você já seja um pouco mais experiente terá uma ótima oportunidade para revisar minuciosamente os assuntos.

Inicialmente faça uma "leitura na diagonal" (*skimming*) desse material. O objetivo dessa fase é captar a "ideia geral" e identificar os assuntos a serem tratados. Em seguida, assista às videoaulas anotando as dúvidas que surgirem ao longo do processo. A terceira fase consiste em "estudar" novamente esse material com maior nível de detalhes e aprofundamento. Nessa etapa, não esqueça de realizar os resumos para futuras revisões. Por fim, faça uma maior quantidade de exercícios possíveis. Durante esse processo, fique à vontade para me procurar no fórum de dúvidas sempre que for necessário.

O curso contempla os seguintes materiais:



Curso completo em videoaulas

teoria e exercícios resolvidos sobre os pontos do edital

Curso completo escrito (pdf)

teoria e mais exercícios resolvidos sobre os pontos do edital

Fórum de dúvidas

para você sanar suas dúvidas sempre que necessário

Como este curso está organizado

Aula	Conteúdo do edital
01	*Princípios de sistemas operacionais – 1ª Parte.*
02	*Princípios de sistemas operacionais – 2ª Parte.*
03	*Princípios de sistemas operacionais – 3ª Parte.* *Aplicações de informática e microinformática.*
04	*RedHat Linux: princípios, conceitos e operação básica – 1ª Parte.*
05	*RedHat Linux: princípios, conceitos e operação básica – 2ª Parte.*
06	*Administração de usuários, grupos, permissões.* *Controle de acesso (LDAP e Active Directory).*

Contatos

romulo@dominandoti.com.br

@profromulosantos

/professorromulosantos

Sistemas Operacionais

Caro concursando,

Seja muito bem-vindo! É um prazer tê-lo conosco.

Este material foi elaborado como parte integrante do curso, para que nele, você possa mergulhar nos assuntos relacionados a "sistemas operacionais".

Estude-o atenciosamente e consulte-o sempre que necessário. Em caso de dúvidas, me procure nossos fóruns da disciplina.

Desejo sinceramente que você aproveite ao máximo esse curso, e que através dele, possa progredir em sua preparação para concursos, bem como, alcançar a sua sonhada nomeação.

Um grande abraço e força nos estudos.

Prof. Rômulo Santos.

1) O que é um Sistema Operacional?

Um sistema computacional (SO) é constituído por uma série de componentes, a saber, um ou mais processadores, memória principal, unidades de discos, impressoras, teclado, *mouse*, monitor, interfaces de rede e uma porção de outros dispositivos de entrada e saída. Dessa forma, podemos classificá-lo como um sistema complexo.

Vamos supor que cada desenvolvedor de aplicações tivesse que entender em detalhes, o funcionamento de todos esses componentes acima mencionados. Acredito que nenhum *software* chegaria a ser concluído. Brincadeiras à parte, é importante que saibamos que gerenciar todos esses componentes e usá-los de maneira otimizada é algo extremamente difícil de ser alcançado. Por essa razão, os computadores atuais possuem um dispositivo de *software* denominado sistema operacional.

Um sistema operacional além de gerenciar todos esses componentes de forma harmônica e otimizada, se destina ainda, a fornecer aos programas de usuário um modelo de computação mais simples e sem a necessidade de entender todas as particularidades do *hardware*.

Quanto aos critérios de localização, um sistema operacional se situa entre o *hardware* e os programas de usuário, conforme ilustrado na figura 1. Consideremos, por exemplo, que para um editor de texto (um dos programas de usuário) ter a possibilidade de gravar um de seus arquivos em disco (*hardware*), é necessário o envolvimento do sistema operacional. Nesse caso, o editor de texto deverá "solicitar" ao sistema operacional serviços relacionados aos módulos de gerenciamento de arquivos e discos. Ou seja, o editor de texto não tem privilégios suficientes para acessar diretamente o disco, sem a participação direta do sistema operacional no processo. Segue abaixo uma síntese das características dos SO

Quadro Resumo 1 - Características dos SO

- Software básico mais importante que é executado em um computador.
- Concede aos programas do usuário a possibilidade de utilizar o equipamento.
- Fornece rotinas básicas para controle do hardware.
- Sem ele o computador e os demais programas existentes seriam praticamente inúteis.

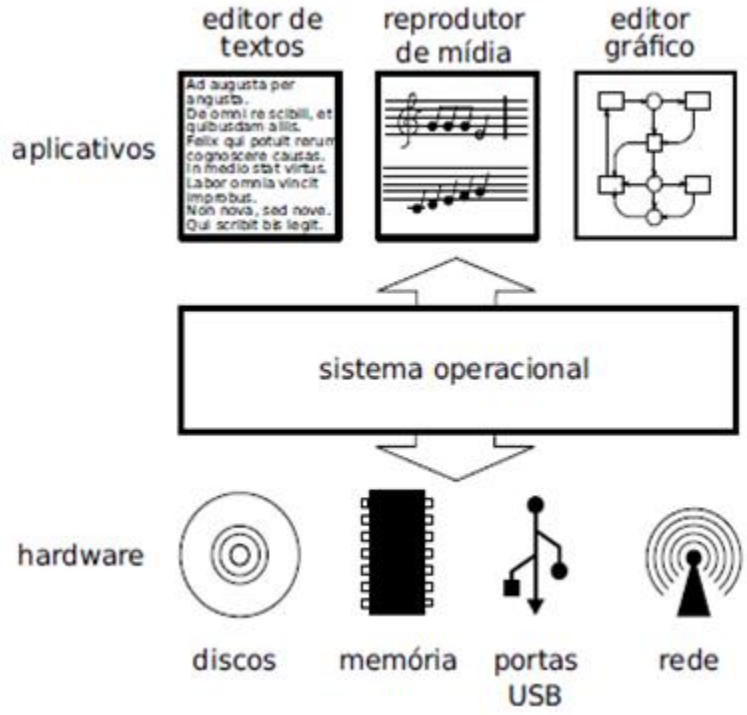


Figura 1: Localização do SO.

Para que um usuário possa utilizar um SO é necessária uma interface. Do exposto, podemos concluir que as "interfaces" são os mecanismos pelos quais os usuários interagem com o SO, seja o Windows, Linux ou qualquer outra plataforma.

As interfaces podem ser implementadas de duas formas:
- Interface de Linha de Comandos (*command-line interface* – CLI): Ex: Shell Linux (interpretador de comandos). Conforme ilustrado na figura 2.

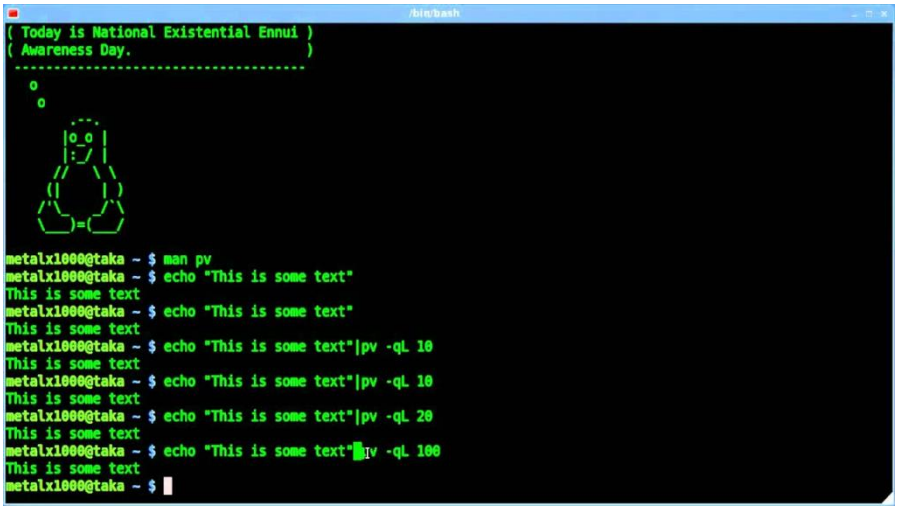


Figura 2: Exemplo de Interface de Linha de Comando do GNU/Linux.

- **Interfaces Gráfica com o Usuário** (*graphical user interface* – GUI): Ex: Windows Explorer (interpretador gráfico usado para gerenciamento de arquivos). Conforme ilustrado na figura 3.

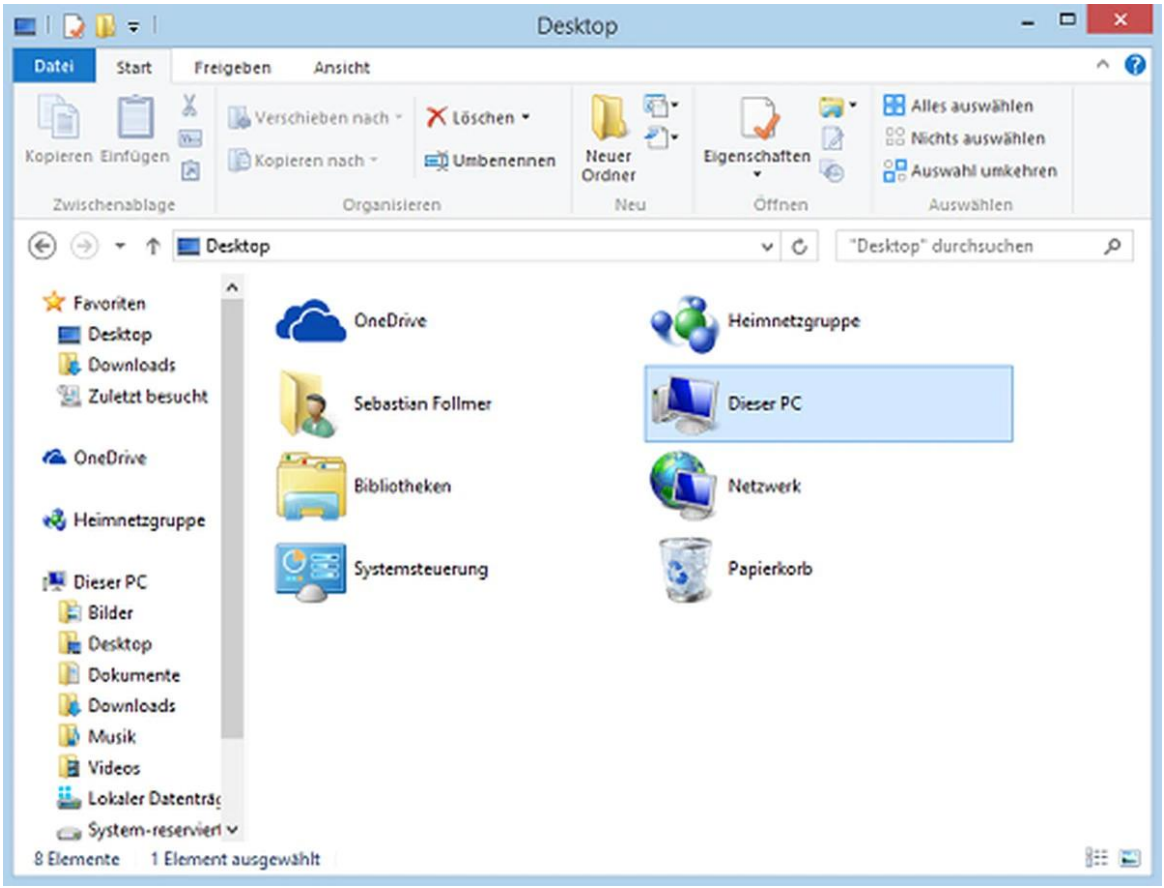


Figura 3: Exemplo de Interface de Linha de Comando do Windows.

Cabe ressaltar que é por intermédio dessas interfaces, seja de linha de comando ou gráfica, que os usuários enviam suas solicitações de serviços (comandos ou cliques de *mouse*) para que sejam utilizados por outras aplicações ou até mesmo serem direcionados para o SO, nas situações em que for necessário o uso de um *hardware*.

Para garantir a execução apropriada do SO, temos que ser capazes de distinguir entre a execução de código do sistema operacional e de um código definido pelo usuário. A abordagem adotada pela maioria dos sistemas de computação é o fornecimento de suporte de *hardware* que permite diferenciar as **modalidades de execução** ou **modos de acesso**.

Precisamos de, pelo menos, dois modos de operação separados: o modo **usuário** e o modo ***kernel*** (também chamada de modo núcleo, supervisor, de sistema ou privilegiado).

O modo usuário **NÃO** possui instruções privilegiadas, logo, precisa do modo *kernel* para intermediar o acesso ao *hardware* da máquina. Por sua vez, o modo núcleo possui instruções privilegiadas e em grande parte dos SO modernos, possui também, instruções não privilegiadas.

Logo, podemos concluir que os programas que funcionam no modo usuário interagem com o modo *kernel*, com o objetivo de que lhes sejam fornecidas instruções privilegiadas que os permitam ter acesso ao *hardware* do computador conforme ilustrado na figura 4.

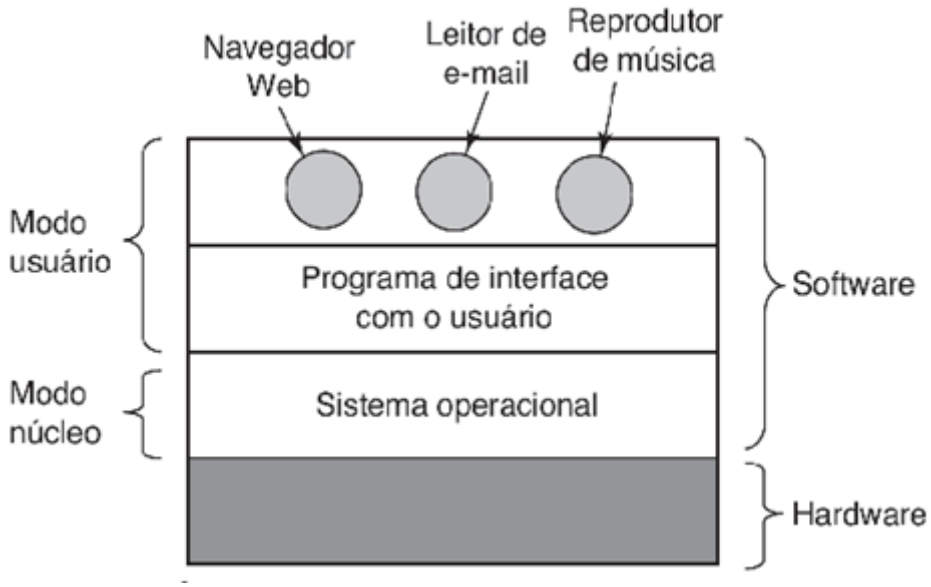


Figura 4: Modos de Execução ou Acesso de um SO.

"DESAFIO GAVIÃO"



***Pergunta**: as interfaces CLI e GUI ficam localizadas no modo usuário ou no modo núcleo?*

***Resposta**: ambas ficam localizadas no modo usuário.*

Agora que entendemos as motivações que levam o modo usuário a ter que interagir com o modo núcleo, precisamos compreender como que se dá essa comunicação. Para responder a esse questionamento é necessário conhecer o conceito de "chamadas de sistemas".

As **chamadas de sistemas** (***system calls***) consistem no mecanismo usado por um programa para **solicitar** um serviço ao SO. Esse conceito é implementado por intermédio de "Interfaces de Programação de Aplicativos" (*Application Programming Interface* - API) tanto no Windows quando no GNU/Linux.

System Calls fornecem o meio para que um programa de usuário solicite ao SO a execução de tarefas reservadas a ele em nome do programa de usuário. Geralmente, uma chamada de sistema assume a forma de uma exceção (*exception*) para uma localização específica no vetor de interrupções. Essa exceção pode ser executada por uma instrução *trap* genérica.

Uma **interrupção de software** (geradas por *exception* ou *traps*) é utilizada para permitir que um programa em modo usuário passe para o modo núcleo e o seu controle para o SO. Diferente da anterior, uma **interrupção por *hardware*** ou simplesmente interrupção consiste em um "evento" onde um dispositivo (*hardware*) solicita a intervenção do processador, e consequentemente, seu tratamento pelo SO.

Para melhor fixar os conhecimentos apresentados "sugiro", entenda como uma forma carinhosa de dizer "eu estou mandando" você resolver as questões de concursos apresentadas a seguir 😂😂😂. A resolução das questões permite que você identifique com clareza como os assuntos que costumam ser cobrados em provas, e principalmente, a forma pela qual (*modus operandi*) a banca examinadora FCC gosta de explorar tais assuntos.

➡ ***Observação**: Em caso de dúvidas na resolução das questões recomendo consultar o item "Questões comentadas pelo professor" no final deste material.*

Questões de Fixação

Questão 01

[FCC – TRT/SE – 2010] Componente central do sistema operacional da maioria dos computadores. Serve de ponte entre aplicativos e o processamento real de dados feito no âmbito do hardware. Essa é a definição de:

a) Sistema.
b) Software
c) Kernel.
d) Microprocessador.
e) Boot.

Questão 02

[FCC – TCM/PA – 2010] A comunicação de uma aplicação com o subsistema de entrada e saída de um sistema operacional é estabelecida por meio de

a) shell.
b) device drivers.
c) system calls.
d) scripting.
e) batch.

Questão 03

[FCC – TRF/3 – 2016] Um Técnico Judiciário de TI do TRF3, ao estudar os princípios dos sistemas operacionais, teve sua atenção voltada ao processo que perfaz a interface do usuário com o sistema operacional. Observou que este processo lê o teclado a espera de comandos, interpreta-os e passa seus parâmetros ao sistema operacional. Entendeu, com isto, que serviços como login/logout, manipulação de arquivos e execução de programas são, portanto, solicitados por meio do interpretador de comandos ou

a) Kernel.
b) System Calls.
c) Shell.
d) Cache.
e) Host.

Questão 04

[FCC – TRF/1 – 2014] Para garantir a execução apropriada do Sistema Operacional (SO), há pelo menos duas modalidades de execução. No momento da inicialização do SO, o hardware começa a operar na modalidade ..I.... . O SO é carregado e dá início às aplicações das pessoas que utilizam o computador na modalidade ..II.... . As lacunas I e II são, correta e respectivamente, preenchidas com:

a) de firmware - de aplicativo
b) prioritária - de programa
c) de alta prioridade - de baixa prioridade
d) prioritária - sem privilégios
e) de kernel - de usuário

Questão 05

[FCC – ARCE – 2012] Sistema Operacional (SO) é uma camada de software colocada sobre o hardware para gerenciar todos os componentes do sistema, apresentando-o ao usuário como uma interface simples de entender e de programar. Considere as afirmativas a seguir sobre Sistemas Operacionais.
I. Os programas de aplicação solicitam serviços ao SO através da execução de chamadas de sistema. Os SOs oferecem Application Program Interfaces (APIs) para que os programadores usem funções para interagir com suas rotinas.
II. O Basic Input/Output System (BIOS) é um dispositivo de hardware que assegura que todos os recursos funcionem em conjunto num computador.
III. Firmware são programas ou instruções gravados no hardware da máquina que permitem a comunicação com outros dispositivos eletrônicos.
IV. A interface entre o SO e os programas de aplicação é definida pelo conjunto de instruções estendidas fornecidas pelo SO. Estas instruções são conhecidas como Dynamic Link Library (DLL).
Está correto o que se afirma em

a) I, II, III e IV.
b) I e II, apenas.
c) II, III e IV, apenas.
d) I e III, apenas.
e) II e IV, apenas.

Questão 06

[FCC – TRF/2 – 2012] Em uma CPU, se um processo estiver executando um programa de um usuário em modo usuário e precisar de um serviço do sistema, como ler dados de um arquivo, terá de executar uma instrução de chamada ao sistema ou uma instrução

a) TSL.
b) Mov.
c) Call.
d) Disable.
e) Trap.

GABARITO - Questões de Fixação

1. C
2. C
3. C
4. E
5. D
6. E

2) Funções dos SO:

Para que um *software* ser considerado um SO é necessário que realize duas funções, a saber: atuar como uma **máquina estendida** e **gerenciar recursos** de *hardware*.

Uma **máquina estendida** fornece aos programas aplicativos um conjunto de recursos claros ao invés de recursos confusos de *hardware*. Podemos identificar, conforme ilustrado na figura 5, o SO apresentando ao programa usuário (aplicativos) uma máquina mais simples (chamadas de sistemas) com as mesmas funções da máquina real (hardware complexo).

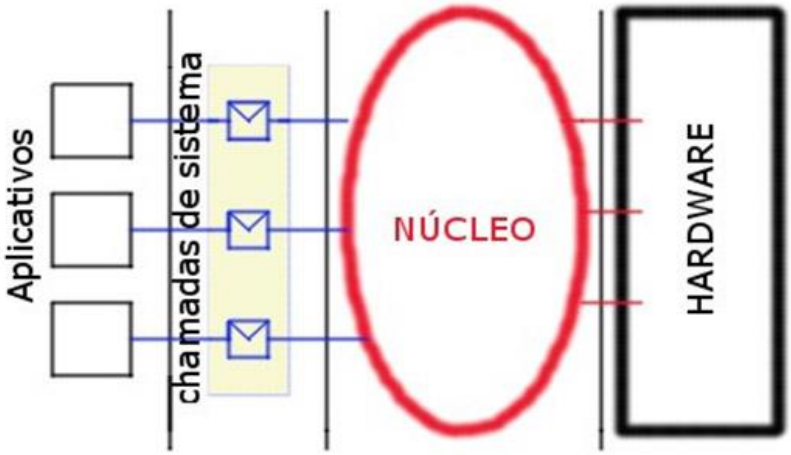


Figura 5: SO como máquina estendida.

É importante entendermos que processadores, memórias e dispositivos de E/S são de complexo entendimento e apresentam interfaces difíceis, desajeitadas, incompatíveis e incoerentes para os programadores que precisam escrever *softwares* que as utilizem. Na visão de Tanenbaum (2010), os sistemas operacionais transformam o *hardware* feio em abstrações bonitas conforme podemos visualizar na figura 6.

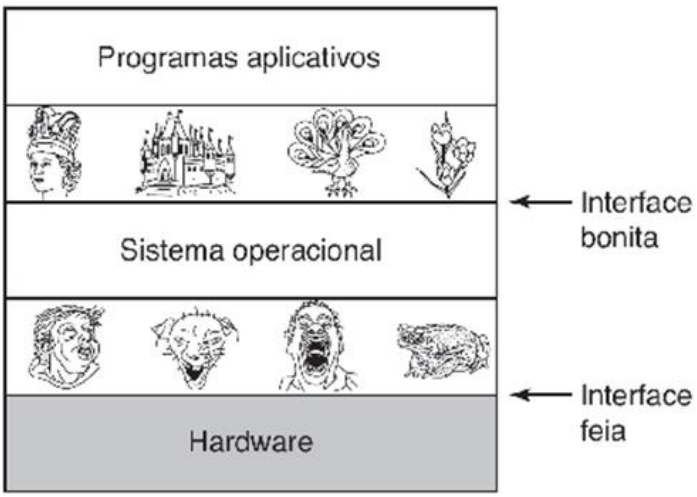


Figura 6: SO transformando o *hardware* feio em abstrações bonitas.

Como **gerenciador de recursos** cabe ao SO gerenciar todas as partes de um sistema complexo de computação. Ou seja, deve fornecer uma alocação ordenada e controlada de processadores, memórias e dispositivos de E/S entre vários programas que competem por eles. A figura 7, ilustra os diferentes módulos de gerenciamento de um SO.

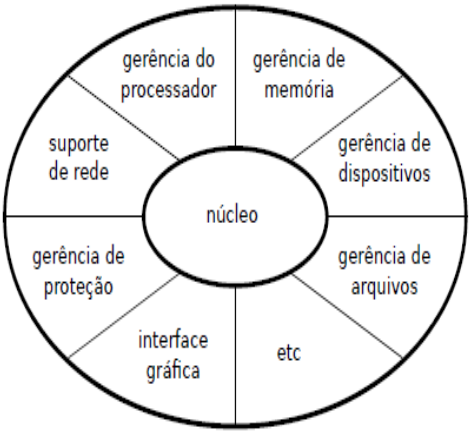


Figura 7: SO como gerenciador de recursos.

Detalhando um pouco mais sobre o gerenciamento de recursos, é possível acrescentar que o SO deve manter o controle sobre quem está usando qual recurso (contabilidade) e atender as requisições de recursos, além de gerenciar e proteger a memória, os dispositivos de entrada e saída e outros recursos.

Por fim, é sua responsabilidade controlar a multiplexação, também conhecida por partilha ou compartilhamento de recursos de duas maneiras diferentes:
- No tempo: diferentes programas ou usuários aguardam sua vez de usá-lo.
- **No espaço**: cada programa ou usuário ocupa uma parte do recurso.

Quadro Resumo 2 – Máquina estendida
- Oculta os detalhes complicados que devem ser executados pelo hardware.
- Apresenta ao usuário uma máquina mais fácil de ser utilizada pela máquina real.

Quadro Resumo 3 – Gerenciador de recurso
- Controla o tempo em que um programa fica com o recurso.
- Controla o espaço compartilhado pelo recurso para cada programa.

Agora que você está familiarizado com as "funções de um SO" resolva as questões apresentadas a seguir, para melhor fixação e entendimento sobre o assunto.

Questões de Fixação

Questão 07

[FCC – TJ/PA – 2009] NÃO é função do Kernel de um sistema operacional:

a) Lançamentos na contabilidade da empresa.
b) Contabilização de uso do sistema.
c) Suporte a redes locais e distribuídas.
d) Gerência do sistema de arquivo.
e) Gerência dos dispositivos de E/S.

Questão 08

[FCC – TRT/PI – 2010] O sistema operacional é responsável por uma ou mais das seguintes atividades relacionadas ao gerenciamento de disco:
(I) Gerenciamento do espaço livre.
(II) Alocação do armazenamento.
(III) Interpretação de comandos.
(IV) Escalonamento do disco.
Está correto o que se afirma em

a) I, II e III, apenas.
b) I, III e IV, apenas.
c) I, II e IV, apenas.
d) II, III e IV, apenas.
e) I, II, III e IV.

Questão 09

[FCC – DPE/SP – 2010] NÃO é uma função do sistema operacional:

a) Permitir aos programas armazenar e obter informações.
b) Controlar o fluxo de dados entre os componentes do computador.
c) Responder a erros e a pedidos do usuário.
d) Impor escalonamento entre programas que solicitam recursos.
e) Gerenciar apenas a base de dados.

GABARITO - Questões de Fixação

7. A 8. C 9. E

3) Classificação dos SO:

Um SO pode possuir diferentes **classificações** considerando, para isso, os variados critérios adotados para a sua correta classificação. Alguns critérios podem ser utilizados sendo os mais comuns: pela quantidade de usuários, quantidade de tarefas, quantidade de processadores, forma que as aplicações são escalonadas, forma de funcionamento em redes de computadores e ambiente de operação.

Considerando a "quantidade de usuários" temos os sistemas operacionais **monousuário** e **multiusuário**. Os sistemas monousuário se caracterizam pelo fato de permitir somente um usuário logado (operação de *login*) por vez. Isso significa que se o usuário "professor" estiver logado o usuário "aluno" terá que esperar o primeiro deslogar (operação de *logout*), para ter condições de utilizar o sistema. Já em sistemas multiusuário, a exemplo dos atuais sistemas operacionais Windows e GNU/Linux, essa atividade pode ser simultânea, ou seja, os usuários "professor" e "aluno" podem estar logados ao mesmo tempo no SO. A figura 8, ilustra um sistema operacional multiusuário.

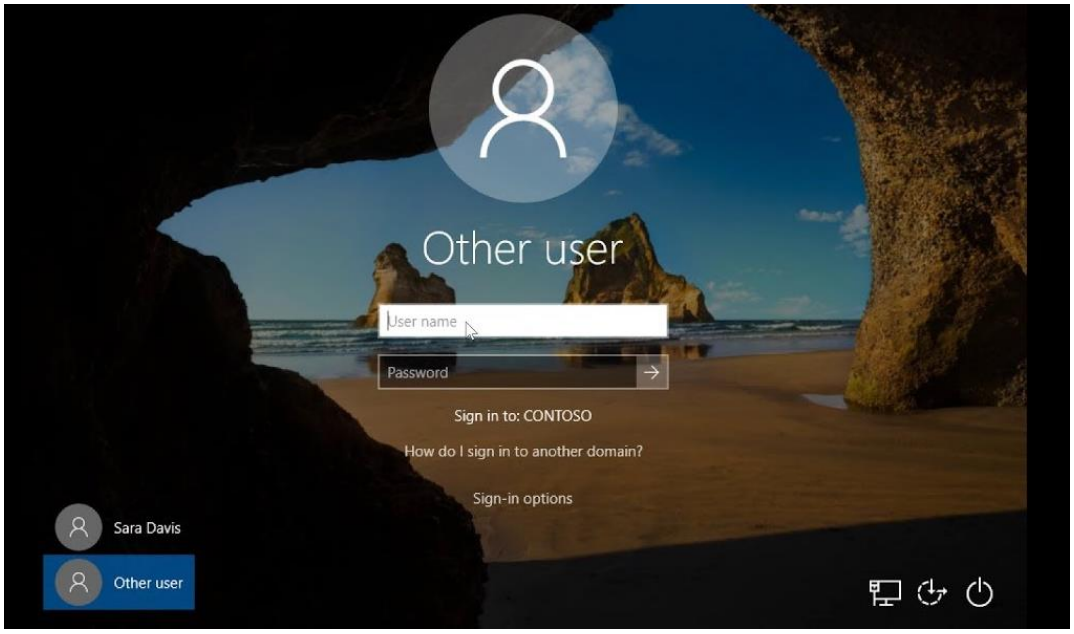


Figura 8: Exemplo de um SO Windows multiusuário.

Quanto ao critério de "quantidade de tarefas" um SO pode ser monotafefa (monoprogramação) ou multitarfa (multiprogramação). Em concursos públicos favor considerar "monotarefa e monoprogamação", assim como, "multitarefa e multiprogramação" como conceitos equivalentes.

Sistemas operacionais monotarefa caracterizam-se pelo fato de terem a capacidade limitada de carregar um único "programa de usuário", para a memória principal, por vez. Considerando que a memória principal possui apenas um programa o processador pode executar apenas uma única tarefa.

Ao contrário, os sistemas operacionais multitarefa podem carregar mais de uma tarefa para a memória simultaneamente. Como existe um único processador é possível alternar o uso do processador para os vários programas já carregados na memória. Temos o conceito de "concorrência". A figura 9, ilustra um SO multitarefa.



Figura 9: Exemplo de um SO GNU/Linux multitarefa.

Um sistema operacional também pode ser classificado conforme a "quantidade de processadores" ou núcleo de processamento utilizado. Nesse caso temos os sistemas operacionais monoprocessado e multiprocessado. Um sistema operacional multiprocessado tem um algoritmo que permite não apenas a identificação de duas ou mais CPUs, em caso da sua existência, mais também a possibilidade de alocar para cada CPU existente a uma tarefa diferente a ser processada. Tal ação permite que sejam implementados não apenas o conceito de "concorrência", mas também o de "simultaneidade", ou seja, várias tarefas sendo realizadas ao mesmo tempo. A figura 9, ilustra uma CPU *multicore* de um *smartphone*.

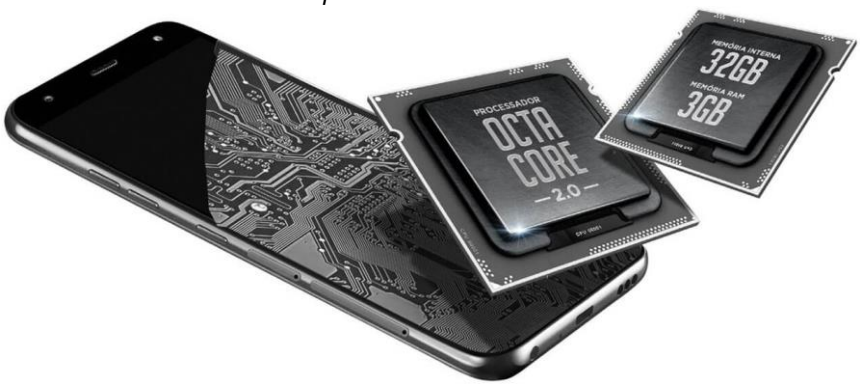


Figura 9: Exemplo de um CPU *multicore*.

É possível subdividir os sistemas multitarefas (multiprogramados) em categorias considerando a forma como as aplicações que rodam nesses sistemas são **escalonadas**. Nessa situação teremos os sistemas operacionais em lote (*batch*), de tempo compartilhado (*time-sharing*) e em tempo real (*real-time*).

Os sistemas *batch* foram os primeiros tipos de sistemas operacionais multiprogramáveis. Os *jobs* (tarefas novas), eram submetidos para execução e aguardavam em uma fila para ser processados. Caracterizam-se por permitir um sequenciamento automático de tarefas (lote de tarefas), ou seja, funcionam como uma "linha de produção" de uma fábrica onde, normalmente, são submetidos vários produtos em série em um processo de fabricação. A figura 10, ilustra as fases de um antigo processamento em lote.

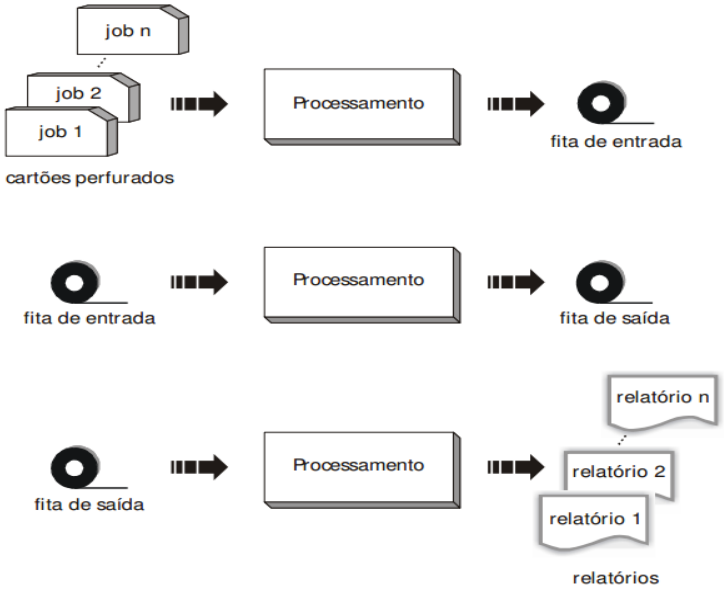


Figura 10: Fases de um processamento em lote.

O processamento em lote não permite a interação do usuário com a aplicação, ou seja, os processos rodam em segundo plano (*background*). Quando bem projetados os SO em lote são eficientes, em razão da melhor utilização do processador, no entanto, oferecerem tempos de resposta longos (somente é possível saber o resultado da primeira tarefa quando todo o lote for concluído).

Os sistemas de tempo compartilhado (*time-sharing*) permitem a execução de vários programas, a partir da divisão do tempo do processador, em pequenos intervalos denominados fatia de tempo (*time-slice*). Caso a fatia de tempo seja insuficiente para a conclusão do programa, ele será interrompido pelo SO (preempção) e substituído (escalonado) por um outro, enquanto aguarda na fila de "prontos" por uma nova fatia de tempo.

Permitem a interação dos usuários com o sistema, ou seja, os processos rodam em "primeiro plano" (*foreground*). Normalmente, respondem em poucos segundos à maioria das solicitações de usuários. A maioria das aplicações comerciais atualmente é processada em sistemas *time-sharing.*

Temos ainda, os SO em tempo real (*real-time*) cujo processamento exige "requisitos rígidos" de tempo e que devem ser obedecidos, caso contrário, poderão ocorrer problemas irreparáveis.

Nesses sistemas não existe a ideia de fatia de tempo, logo, um programa utiliza o processador o tempo que for necessário ou até que apareça outro mais prioritário. A importância ou prioridade de execução de um programa é definida pela própria aplicação e não pelo SO. A figura 11, ilustra um SO em tempo real que controla o funcionamento do sonar (sistema de navegação e determinação de distância por intermédio do som) utilizado em uma fragata (tipo de navio utilizado em conflitos navais).



Figura 11: Processamento em tempo real utilizado para controlar um sonar de uma fragata.

Podemos classificar os SO considerando a forma com utilizam as "redes de computadores" em:

Sistemas operacionais de rede: os usuários sabem da existência de vários computadores, ou seja, dos recursos que são compartilhados na rede. Cada *host* na rede pode executar um SO diferente. São empregados massivamente para compartilhar recursos na rede.

Sistemas operacionais distribuídos: os usuários desconhecem onde os programas são executados e onde os arquivos são armazenados. Normalmente é empregado um único SO em cada computador, estratégia adotada na Computação em *Cluster* (*Cluster Computing*) ou vários SO diferente, comum em Computação em Grade (*Grid Computing*). Nesse último caso necessitam de um sistema intermediário (*middleware*). O *middleware* é o *software* que se encontra entre o sistema operacional e os aplicativos nele executados. A figura 12, ilustra um sistema distribuído organizado com *middleware*.

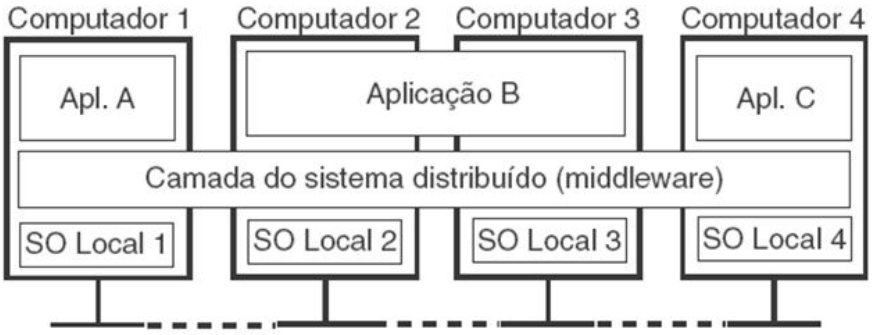


Figura 12: Sistema distribuído organizado com *middleware*.

Por fim, ainda é possível classificar os SO quanto ao "ambiente de operação" em:

Sistemas operacionais interativos: os usuários interagem com o sistema e esperam que suas necessidades sejam atendidas no melhor tempo.

Sistemas operacionais embarcados: operam em dispositivos, normalmente, diferentes do conceito tradicional de computador, como por exemplo, micro-ondas, geladeiras, mp3 *players*, etc.

Computadores pessoais: destinados a tarefas domésticas, como por exemplo, processamento de texto, planilhas e navegação na *web*, etc.

Computadores servidores e *mainframes*: se caracterizam pela maior robustez, capacidade de processamento e disponibilidade.

Mostre que compreendeu bem as classificações dos sistemas operacionais resolvendo as questões de concursos apresentadas a seguir.

Questões de Fixação

Questão 10

[FCC – TCE/GO – 2009] Considere as seguintes características dos sistemas operacionais:
I. ser multiusuário e multitarefa.
II. ser multiprogramável e multiusuário.
III. ser monousuário e multitarefa.
IV. realizar processamento em modo batch.
Está correto o que consta em

a) I, II e III, apenas.
b) I, II, III e IV.
c) I, II e IV, apenas.
d) II e III, apenas.
e) II, III e IV, apenas.

Questão 11

[FCC – TRF/1 – 2014] Um dos aspectos mais importantes dos sistemas operacionais é a capacidade de realizar multiprogramação. Sobre este assunto, é INCORRETO afirmar:

a) A multiprogramação aumenta a utilização da CPU organizando os jobs (código e dados) prontos para serem executados, de modo que a CPU tenha sempre um deles para executar, não ficando ociosa.
b) Em sistemas de tempo compartilhado, a CPU executa múltiplos jobs alternando-se entre eles, mas as mudanças ocorrem com tanta frequência que os usuários ficam impedidos de interagir com os programas enquanto estão sendo executados.
c) O SO mantém vários jobs na memória simultaneamente; como a memória costuma ser pequena para acomodar todos os jobs, estes são mantidos inicialmente em disco na fila de jobs, que é composta por jobs que aguardam alocação na memória principal.
d) A multiprogramação pode ser comparada ao trabalho de um advogado: ele trabalha para vários clientes; enquanto um caso está aguardando julgamento ou esperando documentos, ele pode trabalhar em outro caso.
e) O tempo compartilhado (ou multitarefa) é uma extensão lógica da multiprogramação. Apenas um pequeno tempo de CPU é dado a cada usuário, de forma que ele tem a impressão de que todo o sistema de computação está dedicado exclusivamente ao seu programa.

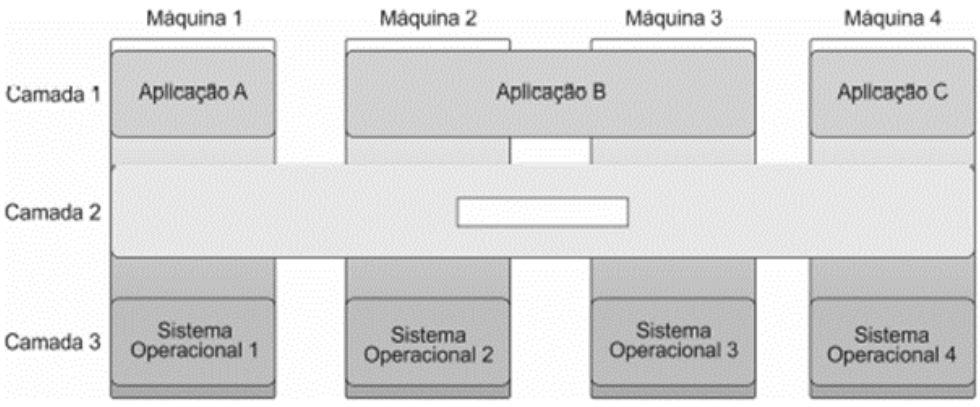
Questão 12

[FCC – DPR/RS – 2013] Para melhor utilizar a CPU, os sistemas operacionais utilizam ... I... , que permite que vários jobs fiquem na memória ao mesmo empo, assegurando que a CPU sempre tenha um job para executar. Os sistemas ....II.... são uma extensão deste mecanismo, em que algoritmos ...III... atuam alternando rapidamente entre os jobs, dando a impressão que cada um deles está sendo executado concorrentemente.
As lacunas I, II e III são, correta e respectivamente, preenchidas por

a) o multiprocessamento - multiprocessados - de escalonamento
b) o multiprocessamento - de tempo real - multithreads
c) o escalonamento - multicore - de tempo compartilhado
d) a multiprogramação - multiprocessados - multicore
e) a multiprogramação - de tempo compartilhado - de escalonamento

Questão 13

[FCC – TRF/5 – 2017] Considere a figura abaixo.



De acordo com a figura, um usuário acessa uma aplicação na Camada 1 achando que está acessando um único sistema em um único local, porém pode estar acessando múltiplos sistemas, que podem estar em máquinas físicas diferentes e com Sistemas Operacionais diferentes. Na verdade, a aplicação não interage com o Sistema Operacional, mas com um software que fica na Camada 2 e este, então, interage com os Sistemas Operacionais, na Camada 3, que podem ser de diferentes plataformas. O software da Camada 2 permite, por exemplo, que o usuário execute solicitações como enviar formulários em um navegador da web ou permitir que o servidor web apresente páginas dinâmicas da web com base no perfil de um usuário. O software da Camada 2 é denominado

a) Firmware.
b) Kernel.
c) Middleware.
d) BIOS.
e) Bootstrap.

Questão 14

[FCC – TJ/PA – 2009] Quando um sistema operacional permite a comunicação direta do usuário com o processo durante o seu processamento está executando um processo

a) Backend.
b) Front-End.
c) Foreground.
d) Background.
e) I/O Bound.

GABARITO - Questões de Fixação

10. B
11. B
12. E
13. C
14. C

4) Conceitos de processos e *threads*:

Basicamente um processo consistem em um programa em execução. Associado a cada processo está o seu **espaço de endereçamento,** ou seja, uma lista de posições de memória em que este processo pode ler e escrever. Há também um conjunto de registradores (**contexto de *hardware***), que inclui o contador de programa, o ponteiro para a pilha de execução e demais informações necessárias para executar um programa. Por fim, o SO registra uma série de características do processo incluídas na execução de um programa consideradas como **contexto de *software*** (identificação, cotas, privilégios, etc). A figura 13, ilustra o contexto de um processo.

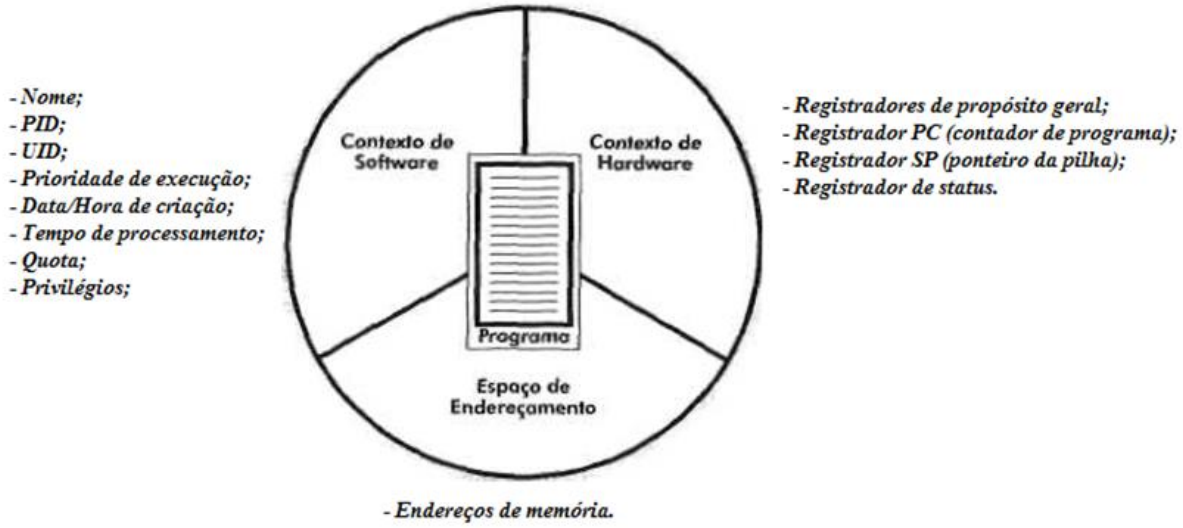


Figura 13: Contexto de um processo.

Um processo pode transitar entre vários estados ao longo da sua execução. Estes estados são:
- **Novo**: o processo ainda não foi admitido, ou seja, aguarda a vez de ser carregado para a memória. O SO, naturalmente, não reconhece esse estado. Pode, por exemplo, ser um arquivo executável que está armazenado no HD.
- **Pronto** (*ready*): o processo está carregado na memória e esperando para ser executado pela CPU.
- **Execução** (*running*): o processo está sendo executado pela CPU.
- **Bloqueado** (*wait*): o processo está esperando algum evento externo ou por algum recurso de I/O para poder prosseguir seu processamento.
- **Terminado**: neste estado temos a finalização do processo. Aqui há a devolução para o SO das estruturas de controles utilizadas (contexto de *software*, contexto de *hardware* e espaço de endereçamento de memória).

A maior parte das literaturas consagradas consideram apenas os estados de **pronto, em execução** e **bloqueado** como aqueles que o SO controla de fato. A figura 14, ilustra os estados de um processo.

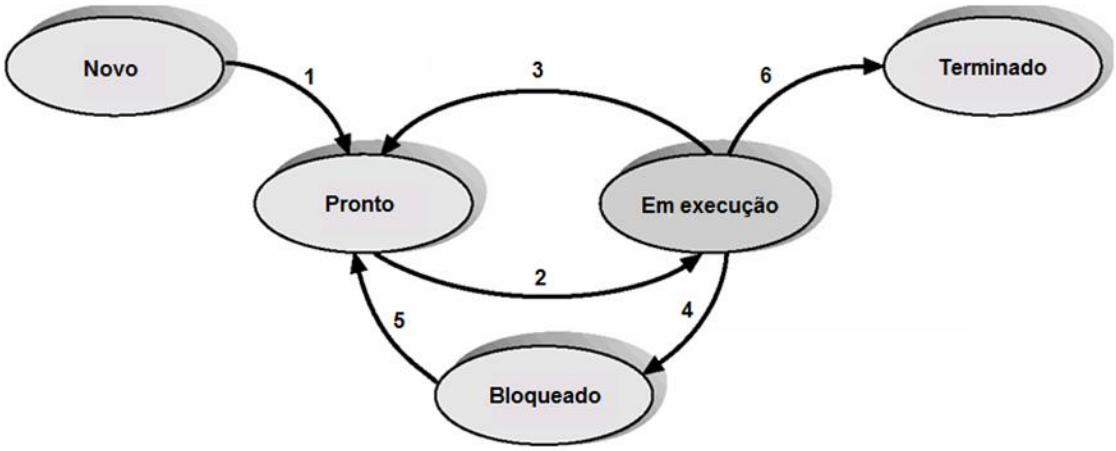


Figura 14: Estados de um processo.

Em sistemas operacionais tradicionais, cada processo tem um espaço de endereçamento e um único *thread* (fluxo) de controle. Na verdade, isso é quase uma definição de processo. Contudo, frequentemente há situações em que é desejável ter múltiplos *threads* de controle no mesmo espaço de endereçamento executando quase em paralelo como se fossem dois processos separados.

Os *threads* são utilizados para que múltiplas execuções ocorram no mesmo ambiente do processo com um grande grau de independência uma da outra. São fáceis de criar e destruir, não possuem recursos associados (recurso é alocado ao processo).

Quadro Resumo 4 – privativos de cada *thread*
- Conjunto de registradores e informações de controle.
- Espaço da pilha na memória.

Quadro Resumo 5 – compartilhados entre as *threads*
- Recursos do SO.
- Espaço de endereçamento (especialmente, a seção de códigos e a seção de dados).

A figura 15, ilustra um processo *multithread*.

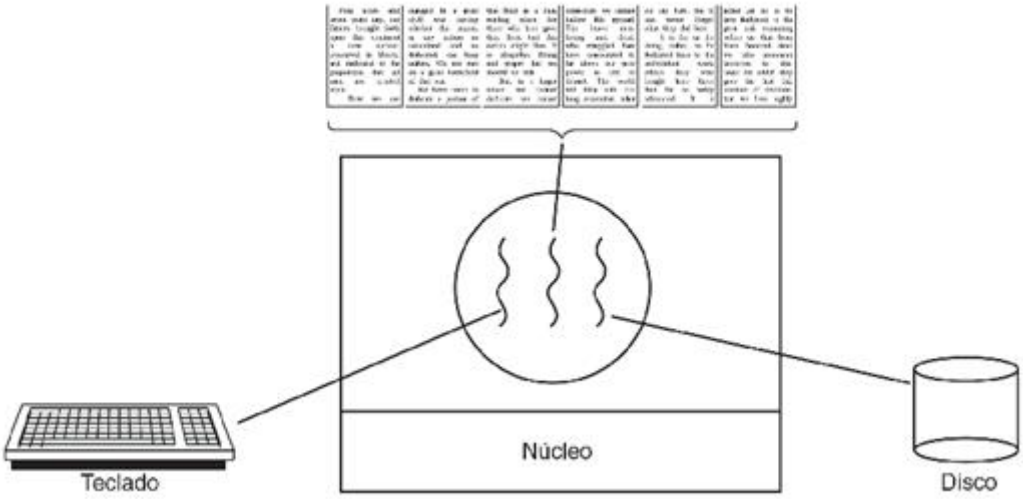


Figura 15: Processo *multithread*.

Para não perder o bom hábito faremos algumas questões sobre o tema anteriormente elucidado.

Questões de Fixação

Questão 15

[FCC – TRF/4 – 2010] No contexto de sistemas operacionais, sobre processos é correto afirmar:

a) Um computador multiprocessador pode ter tantos processos em execução quanto são os processadores.
b) Os termos processo e programa são sinônimos.
c) Todo processo utiliza o mesmo espaço de endereço.
d) A qualquer dado instante somente um processo pode executar instruções em um computador com qualquer tipo de arquitetura.
e) Um processo não pode ter um número zero de processos-pai.

Questão 16

[FCC – DPE/SP – 2013] Em sistemas com multiprogramação, inicialmente um novo processo é inserido na fila de prontos. Este processo aguarda até ser selecionado para execução ou ser despachado. Uma vez que o

processo seja alocado à CPU, eventos podem ocorrer. Analise as afirmativas sobre o processo.
I. pode emitir uma solicitação de I/O e então ser inserido em uma fila de I/O.
II. pode criar um novo subprocesso e esperá-lo terminar.
III. pode ser removido forçosamente da CPU, como resultado de uma interrupção, e ser devolvido à fila de prontos.
Está correto o que se afirma em

a) I, apenas.
b) II, apenas.
c) III, apenas.
d) I e II, apenas.
e) I, II e III.

Questão 17

[FCC – TJ/PA – 2009] Um processo em um sistema operacional pode mudar para o estado de pronto a partir do estado de

a) criação, apenas.
b) execução, apenas.
c) criação ou término, apenas.
d) execução ou término, apenas.
e) criação, espera ou execução.

Questão 18

[FCC – DPE/AM – 2018] Um sistema operacional moderno prevê uma maneira de um processo dividir a si mesmo em outras tarefas, as quais podem ser executadas concorrentemente. Essas novas tarefas são denominadas

a) interrupções.
b) Kernel.
c) máquina virtual.
d) Servlets.
e) Threads.

Questão 19

[FCC – Prefeitura de Teresina/PI – 2016] Considere uma aplicação com 4 threads. Em um sistema

a) multicore concorrência significa que a execução dos threads será intercalada com o passar do tempo.
b) com vários núcleos concorrência significa que os threads podem ser executados em paralelo.
c) em que cada thread só pode ser executado por vez, o sistema operacional é multicore.
d) multicore os threads podem ser atribuídos aos núcleos simultaneamente, somente se houver 4 núcleos, um para cada thread.
e) com um único núcleo à aplicação multithread provocará deadlock ou starvation.

Questão 20

[FCC – TRE/SP – 2012] Nos conceitos de ciência da computação, um processo é um módulo executável que pode conter threads. Um conceito importante sobre threads que estão contidas no mesmo processo é que

a) podem compartilhar a memória do processo.
b) evitam que processos concorrentes acessem recursos do processo principal.
c) acessam a memória estática ao contrário de processos convencionais.
d) controlam o acesso da memória dinâmica (heap) dos processos externos.
e) possuem proteção contra problemas comuns de processos, como vazamento e acessos inválidos da memória.

Questão 21

[FCC – TJ/PE – 2012] As threads de usuário são admitidas acima do kernel e gerenciadas sem o suporte do kernel, enquanto as threads de kernel são admitidas e gerenciadas

a) exclusivamente pela memória RAM.
b) diretamente pela unidade central de processamento.
c) diretamente pelo sistema operacional.
d) exclusivamente por uma deadlock específica.
e) exclusivamente pelo IPL ou boot.

GABARITO - Questões de Fixação

15. A	16. E	17. E
18. E	19. B	20. A
21. C		

5) Resumo Direcionado:

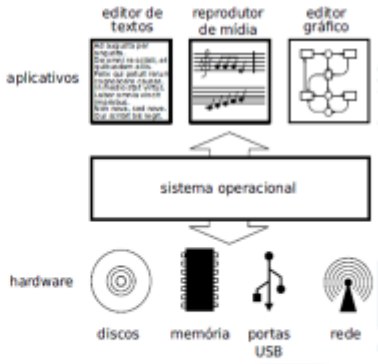
Infraestrutura de TI

- O que é um Sistema Operacional?

Um computador é composto por:
- Um ou mais processadores.
- Memória principal.
- Discos.
- Outros dispositivos de E/S.

Para gerenciar todos esses componentes é necessária uma camada de *software*, ou seja, o sistema operacional (SO).





DIREÇÃO CONCURSOS — CURSOS ONLINE DOMINANDO TI

Infraestrutura de TI

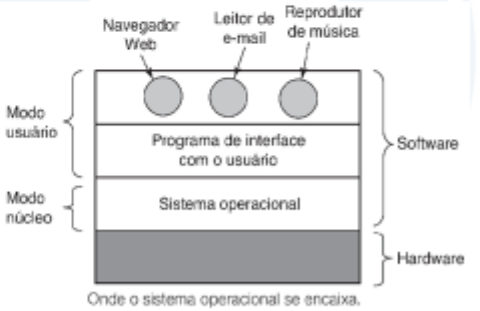
- O que é um Sistema Operacional?

O usuário **interage** como o SO por meio de **interfaces**:
- *CLI* (*command-line interface*). Ex: Shell (interpretador de comandos).
- *GUI* (*graphical user interface*). Ex: Windows Explorer (interpretador gráfico usado para gerenciamento de arquivos).

Há **modos** de **acesso** aplicados aos *softwares*.

- *Não tem privilégios ao hardware.*
- *Interage com o modo núcleo (por meio de chamadas de sistema).*

- *Tem privilégios ao hardware.*
- *Interage com o hardware (registradores, MP, discos, etc).*



Onde o sistema operacional se encaixa.

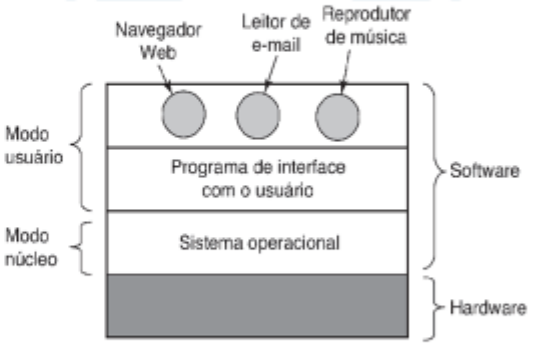
DIREÇÃO CONCURSOS — CURSOS ONLINE DOMINANDO TI

Infraestrutura de TI

- O que é um Sistema Operacional?

Comunicação dos modos de acesso entre si e com o *hardware*.

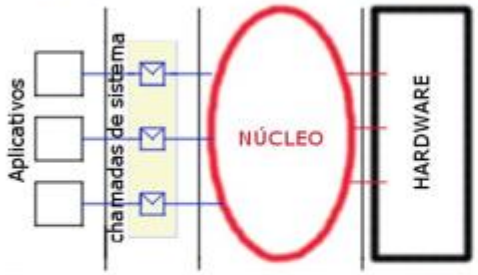
- ***Chamada de sistema*** *(system call): mecanismo usado por um programa para "solicitar um serviço" do SO. Ex: Windows/Linux API (Application Programming Interface).*
- ***Interrupção de software****: utilizada para permitir que um programa em modo usuário passe para o modo núcleo e o seu controle para o SO. Conhecidas também por* ***traps*** *ou* ***exceptions****.*
- ***Interrupção (hardware)****: consiste em um "evento" onde um dispositivo (hardware) solicita a intervenção do processador e o tratamento pelo SO.*



Onde o sistema operacional se encaixa.

Infraestrutura de TI

- Funções dos SO:



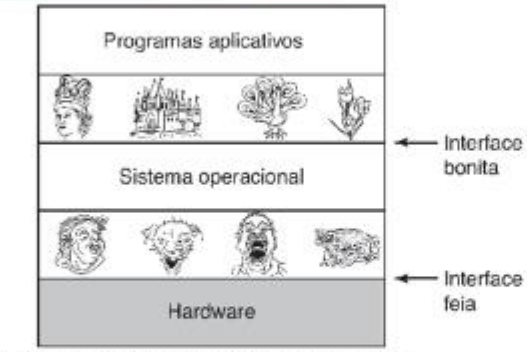
SO como **máquina estendida**.

SO como **gerenciador de recursos**.



Infraestrutura de TI

- SO como máquina estendida:



Sistemas operacionais transformam hardware feio em abstrações bonitas.

O SO deve ocultar o *hardware* oferecendo aos aplicativos **abstrações** claras e simples.

DIREÇÃO CONCURSOS — CURSOS ONLINE DOMINANDO TI

Infraestrutura de TI

- SO como gerenciador de recursos:
  - Deve **manter o controle** sobre quem está usando qual recurso (contabilidade) e atender as requisições de recursos.
  - Gerenciar e proteger a memória, os dispositivos de entrada e saída e outros recursos.
  - Controlar a multiplexação (partilha ou compartilhamento) de recursos de duas maneiras diferentes:
    - No tempo.
    - No espaço.



DIREÇÃO CONCURSOS — CURSOS ONLINE DOMINANDO TI

Infraestrutura de TI

- Classificação dos SO:

1) Quantidade de **usuários**:
- Monousuário e Multiusuário.

2) Quantidade de **tarefas**:
- Monotarefa (monoprogramação): é executada apenas uma tarefa por vez.
- Multitarefa (multiprogramação): são executadas concorrentemente mais de uma tarefa por vez.



3) Quantidade de **processadores** (ou núcleos):
- Monoprocessado.
- Multiprocessado: são executadas concorrentemente e/ou simultaneamente (mais de uma tarefa por vez).

DIREÇÃO CONCURSOS — CURSOS ONLINE DOMINANDO TI

Infraestrutura de TI

- Classificação dos SO:

4) Forma que as aplicações são **escalonadas**:



a. Em **lote** (*batch*): primeiros sistemas multiprogramados que permitiam sequenciamento automático de tarefas:
- não é realizada interação do usuário com a aplicação.
- oferece tempos de respostas longos.



b. Em **tempo compartilhado** (*time-sharing*):
- variante da multiprogramação (trabalha com fatias de tempo).

c. Em **tempo real** (*real-time*): existem requisitos rígidos de tempo (normalmente ambientes críticos onde existe alta disponibilidade).



DIREÇÃO CONCURSOS — CURSOS ONLINE DOMINANDO TI

Infraestrutura de TI

- Classificação dos SO:

5) Baseado em **redes** de computadores:



a. Sistemas operacionais de **rede**:
- usuários sabem da existência de vários computadores (ou recursos).
- cada computador pode executar um SO diferente.
- permite compartilhar recursos na rede.

b. Sistemas operacionais **distribuídos**:
- usuários desconhecem onde os programas são executados e onde os arquivos são armazenados.
- existência de um único SO em cada computador (comum em *Cluster Computing*) ou mais de um SO (comum em *Grid Computing* - usam para isso o ***Middleware***).
- a mesma aplicação pode ser executada em vários computadores.

DIREÇÃO CONCURSOS — CURSOS ONLINE DOMINANDO TI

Infraestrutura de TI

- Classificação dos SO:

6) Quanto ao **ambiente** de **operação**:



a. Sistemas operacionais **interativos**:
- usuários interagem com o sistema e esperam que suas necessidades sejam atendidas no melhor tempo.

b. Sistemas operacionais **embarcados**:
- operam em outros dispositivos (diferentes do conceito tradicional de computador, como microondas, geladeiras, mp3 players, etc) e possui funções específicas e limitadas (em geral não permite modificações e atualizações no sistema).

c. Computadores **pessoais**: destinados à tarefas domésticas.

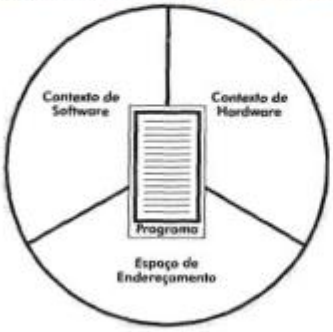
d. **Servidores** e ***mainframes***: possuem maior robustez.

DIREÇÃO CONCURSOS — CURSOS ONLINE DOMINANDO TI

Infraestrutura de TI

- Características de um processo:
  - **Processo**: abstração da execução do programa (sequência de instruções). Se um programa estiver executando duas vezes, há dois processos. Sua **estrutura** envolve: contexto de *hardware*, contexto de *software* e espaço de endereçamento.

- *Nome;*
- *PID;*
- *UID;*
- *Prioridade de execução;*
- *Data/Hora de criação;*
- *Tempo de processamento;*
- *Quota;*
- *Privilégios;*



- *Registradores de propósito geral;*
- *Registrador PC (contador de programa);*
- *Registrador SP (ponteiro da pilha);*
- *Registrador de status.*

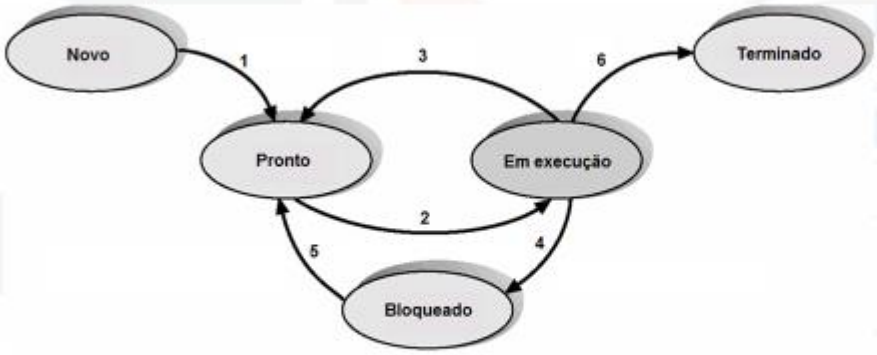
- *Endereços de memória.*

DIREÇÃO CONCURSOS

CURSOS ONLINE DOMINANDO TI

Infraestrutura de TI

- Estados dos processos:



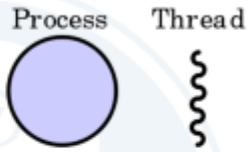
1. O processo é admitido (novo para pronto).
2. O escalonador seleciona o processo (pronto para em execução).
3. O escalonador seleciona outro processo (em execução para pronto).
4. O processo é bloqueado aguardando uma entrada (em execução para bloqueado).
5. A entrada torna-se disponível (bloqueado para pronto).
6. O processo é terminado (em execução para terminado).

DIREÇÃO CONCURSOS

CURSOS ONLINE DOMINANDO TI

Infraestrutura de TI

- *Thread*:
  - É um processo **leve** (mini processo), ou seja, um "tipo de processo" dentro de um outro processo, contém **privativamente**:
    - Conjunto de registradores e espaço de pilha.
  - **Compartilham** com outras *threads* vinculadas (no mesmo processo):
    - Recursos do SO.
    - Seção de código e de dados (espaço de endereçamento).

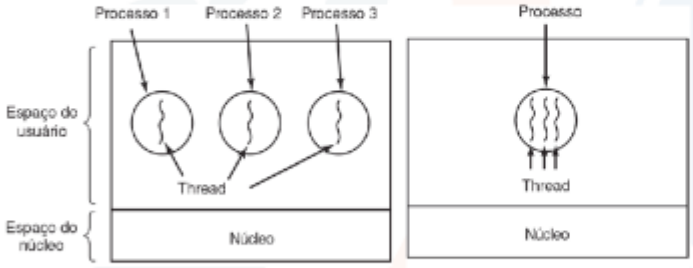


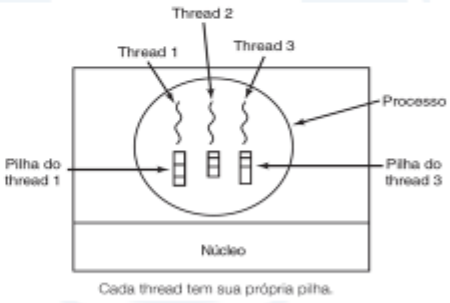


Um processador de textos com três threads.

Infraestrutura de TI

- *Thread*:
  - **Vantagens**:
    - São fáceis de criar e destruir, não possuem recursos associados (recurso é alocado ao processo).





Cada thread tem sua própria pilha.

6) Questões comentadas pelo professor:

Questão 01

[FCC – TRT/SE – 2010] Componente central do sistema operacional da maioria dos computadores. Serve de ponte entre aplicativos e o processamento real de dados feito no âmbito do hardware. Essa é a definição de:

a) Sistema.
b) Software
c) Kernel.
d) Microprocessador.
e) Boot.

_____
Comentários:
Duas informações relevantes são apresentadas no comando da questão e permitem claramente identificar a resposta da correta.
***Primeira**: "Componente central do sistema operacional ...";*
***Segunda**: "... Serve de ponte entre aplicativos e ... hardware...".*
(Resposta C)
Observação:
***Boot**: é um termo utilizado para fazer referência ao processo de inicialização de um computador.*

Questão 02

[FCC – TCM/PA – 2010] A comunicação de uma aplicação com o subsistema de entrada e saída de um sistema operacional é estabelecida por meio de

a) shell.
b) device drivers.
c) system calls.
d) scripting.
e) batch.

_____
Comentários:
Uma aplicação se comunica com o sistema operacional por intermédio de "chamadas de sistemas".
(Resposta C)

Questão 03

[FCC – TRF/3 – 2016] Um Técnico Judiciário de TI do TRF3, ao estudar os princípios dos sistemas operacionais, teve sua atenção voltada ao processo que perfaz a interface do usuário com o sistema operacional. Observou que este processo lê o teclado a espera de comandos, interpreta-os e passa seus parâmetros ao sistema operacional. Entendeu, com isto, que serviços como login/logout, manipulação de arquivos e execução de programas são, portanto, solicitados por meio do interpretador de comandos ou

a) Kernel.
b) System Calls.
c) Shell.
d) Cache.

e) Host.

_____

Comentários:

Uma interface de linha de comandos é consubstanciada por um interpretador de comandos, também conhecido por shell.

(Resposta C)

Questão 04

[FCC – TRF/1 – 2014] Para garantir a execução apropriada do Sistema Operacional (SO), há pelo menos duas modalidades de execução. No momento da inicialização do SO, o hardware começa a operar na modalidade ..I.... . O SO é carregado e dá início às aplicações das pessoas que utilizam o computador na modalidade ..II.... . As lacunas I e II são, correta e respectivamente, preenchidas com:

a) de firmware - de aplicativo
b) prioritária - de programa
c) de alta prioridade - de baixa prioridade
d) prioritária - sem privilégios
e) de kernel - de usuário

_____

Comentários:

O hardware inicia a sua operação no modo kernel (núcleo ou supervisor) já as aplicações operam na modalidade de execução de usuário.

(Resposta E)

Questão 05

[FCC – ARCE – 2012] Sistema Operacional (SO) é uma camada de software colocada sobre o hardware para gerenciar todos os componentes do sistema, apresentando-o ao usuário como uma interface simples de entender e de programar. Considere as afirmativas a seguir sobre Sistemas Operacionais.

I. Os programas de aplicação solicitam serviços ao SO através da execução de chamadas de sistema. Os SOs oferecem Application Program Interfaces (APIs) para que os programadores usem funções para interagir com suas rotinas.
II. O Basic Input/Output System (BIOS) é um dispositivo de hardware que assegura que todos os recursos funcionem em conjunto num computador.
III. Firmware são programas ou instruções gravados no hardware da máquina que permitem a comunicação com outros dispositivos eletrônicos.
IV. A interface entre o SO e os programas de aplicação é definida pelo conjunto de instruções estendidas fornecidas pelo SO. Estas instruções são conhecidas como Dynamic Link Library (DLL).
Está correto o que se afirma em

a) I, II, III e IV.
b) I e II, apenas.
c) II, III e IV, apenas.
d) I e III, apenas.
e) II e IV, apenas.

Comentários:
O BIOS é um firmware para placas-mãe de computadores e um acrônimo para Sistema Básico de Entrada e Saída. A interface entre o SO e os programas de aplicação é definida pelo conjunto de instruções estendidas fornecidas pelo SO. Estas instruções são conhecidas como System Calls.

(Resposta D)

Questão 06

[FCC – TRF/2 – 2012] Em uma CPU, se um processo estiver executando um programa de um usuário em modo usuário e precisar de um serviço do sistema, como ler dados de um arquivo, terá de executar uma instrução de chamada ao sistema ou uma instrução

a) TSL.
b) Mov.
c) Call.
d) Disable.
e) Trap.

Comentários:
Trap é uma interrupção de software causada por uma "system call" e que gera uma mudança de modo.

(Resposta E)

Questão 07

[FCC – TJ/PA – 2009] NÃO é função do Kernel de um sistema operacional:

a) Lançamentos na contabilidade da empresa.
b) Contabilização de uso do sistema.
c) Suporte a redes locais e distribuídas.
d) Gerência do sistema de arquivo.
e) Gerência dos dispositivos de E/S.

Comentários:
A atividade de lançamentos na "contabilidade da empresa" pode ser automatizada por um programa em modo usuário, não sendo uma atividade obrigatória do SO.

(Resposta A)

Questão 08

[FCC – TRT/PI – 2010] O sistema operacional é responsável por uma ou mais das seguintes atividades relacionadas ao gerenciamento de disco:
(I) Gerenciamento do espaço livre.
(II) Alocação do armazenamento.
(III) Interpretação de comandos.
(IV) Escalonamento do disco.
Está correto o que se afirma em

a) I, II e III, apenas.
b) I, III e IV, apenas.
c) I, II e IV, apenas.
d) II, III e IV, apenas.
e) I, II, III e IV.

_____
Comentários:
O interpretador de comando ou shell é implementado por programa em modo usuário.
(Resposta C)

Questão 09

[FCC – DPE/SP – 2010] NÃO é uma função do sistema operacional:

a) Permitir aos programas armazenar e obter informações.
b) Controlar o fluxo de dados entre os componentes do computador.
c) Responder a erros e a pedidos do usuário.
d) Impor escalonamento entre programas que solicitam recursos.
e) Gerenciar apenas a base de dados.

_____
Comentários:
Um Sistema Gerenciador de Banco de Dados (SGBD) é um programa de usuário responsável pela gerência das bases de dados.
(Resposta E)

Questão 10

[FCC – TCE/GO – 2009] Considere as seguintes características dos sistemas operacionais:
I. ser multiusuário e multitarefa.
II. ser multiprogramável e multiusuário.
III. ser monousuário e multitarefa.
IV. realizar processamento em modo batch.
Está correto o que consta em

a) I, II e III, apenas.
b) I, II, III e IV.
c) I, II e IV, apenas.
d) II e III, apenas.
e) II, III e IV, apenas.

_____
Comentários:
Um SO é classificado quantos seguintes critérios:
1) Quantidade de usuários: monousuário e multiusuário.
2) Quantidade de tarefas: monotarefa (monoprogramação) e multitarefa (multiprogramação).
3) Quantidade de processadores: monoprocessado e multiprocessado.
4) Forma que as aplicações são escalonadas: em lote, em tempo compartilhado e em tempo real.
5) Baseado em redes de computadores: SO de redes e distribuídos.
6) Quanto ao ambiente de operação: SO interativos, embarcados, de desktops e mobiles e de servidores e

mainframes.

(Resposta B)

Questão 11

[FCC – TRF/1 – 2014] Um dos aspectos mais importantes dos sistemas operacionais é a capacidade de realizar multiprogramação. Sobre este assunto, é INCORRETO afirmar:

a) A multiprogramação aumenta a utilização da CPU organizando os jobs (código e dados) prontos para serem executados, de modo que a CPU tenha sempre um deles para executar, não ficando ociosa.
b) Em sistemas de tempo compartilhado, a CPU executa múltiplos jobs alternando-se entre eles, mas as mudanças ocorrem com tanta frequência que os usuários ficam impedidos de interagir com os programas enquanto estão sendo executados.
c) O SO mantém vários jobs na memória simultaneamente; como a memória costuma ser pequena para acomodar todos os jobs, estes são mantidos inicialmente em disco na fila de jobs, que é composta por jobs que aguardam alocação na memória principal.
d) A multiprogramação pode ser comparada ao trabalho de um advogado: ele trabalha para vários clientes; enquanto um caso está aguardando julgamento ou esperando documentos, ele pode trabalhar em outro caso.
e) O tempo compartilhado (ou multitarefa) é uma extensão lógica da multiprogramação. Apenas um pequeno tempo de CPU é dado a cada usuário, de forma que ele tem a impressão de que todo o sistema de computação está dedicado exclusivamente ao seu programa.

_____

Comentários:

Os sistemas em tempo compartilhado caracterizam-se pela interação constante de usuários com o sistema, ou seja, os processos funcionam em primeiro plano (foreground).

(Resposta B)

Questão 12

[FCC – DPR/RS – 2013] Para melhor utilizar a CPU, os sistemas operacionais utilizam ... I... , que permite que vários jobs fiquem na memória ao mesmo empo, assegurando que a CPU sempre tenha um job para executar. Os sistemas ....II.... são uma extensão deste mecanismo, em que algoritmos ...III... atuam alternando rapidamente entre os jobs, dando a impressão que cada um deles está sendo executado concorrentemente.
As lacunas I, II e III são, correta e respectivamente, preenchidas por

a) o multiprocessamento - multiprocessados - de escalonamento
b) o multiprocessamento - de tempo real - multithreads
c) o escalonamento - multicore - de tempo compartilhado
d) a multiprogramação - multiprocessados - multicore
e) a multiprogramação - de tempo compartilhado - de escalonamento

_____

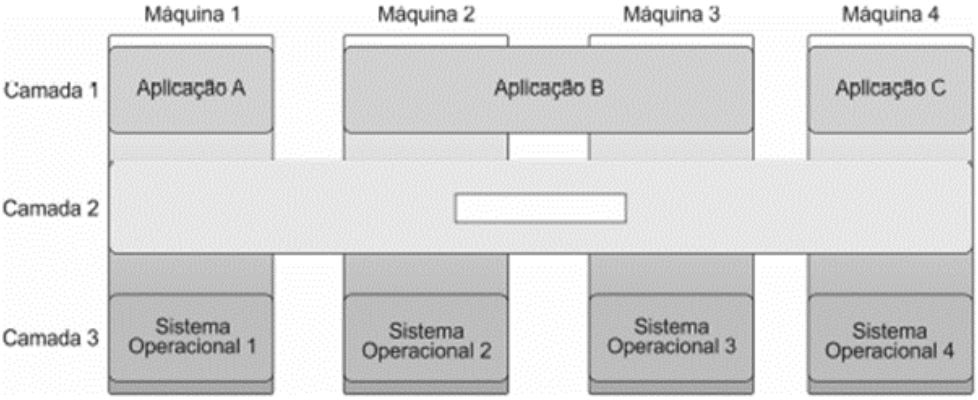
Comentários:

A multitarefa é o recurso que permite que várias tarefas ocupem um espaço de endereço "na memória" ao mesmo tempo. Os sistemas em time em tempo compartilhado caracterizam-se pela interação constante de usuários com o sistema, ou seja, os processos funcionam em primeiro plano. Por fim, utilizam como política de escalonamento a "alternância de tarefas" considerando para isso a fatia de tempo ou time slice.

(Resposta E)

Questão 13

[FCC – TRF/5 – 2017] Considere a figura abaixo.



De acordo com a figura, um usuário acessa uma aplicação na Camada 1 achando que está acessando um único sistema em um único local, porém pode estar acessando múltiplos sistemas, que podem estar em máquinas físicas diferentes e com Sistemas Operacionais diferentes. Na verdade, a aplicação não interage com o Sistema Operacional, mas com um software que fica na Camada 2 e este, então, interage com os Sistemas Operacionais, na Camada 3, que podem ser de diferentes plataformas. O software da Camada 2 permite, por exemplo, que o usuário execute solicitações como enviar formulários em um navegador da web ou permitir que o servidor web apresente páginas dinâmicas da web com base no perfil de um usuário. O software da Camada 2 é denominado

a) Firmware.
b) Kernel.
c) Middleware.
d) BIOS.
e) Bootstrap.

_____

Comentários:

O middleware é o software que se encontra entre o sistema operacional e os aplicativos nele executados. Conhecido como "camada do meio", uma vez que ele conecta duas camadas de software.

(Resposta C)

Questão 14

[FCC – TJ/PA – 2009] Quando um sistema operacional permite a comunicação direta do usuário com o processo durante o seu processamento está executando um processo

a) Backend.
b) Front-End.
c) Foreground.
d) Background.
e) I/O Bound.

_____

Comentários:

Sistemas interativos caracterizam-se por permitir a comunicação direta do usuário com o SO, ou seja, os processos rodam em primeiro plano (foreground).

(Resposta C)

Questão 15

[FCC – TRF/4 – 2010] No contexto de sistemas operacionais, sobre processos é correto afirmar:

a) Um computador multiprocessador pode ter tantos processos em execução quanto são os processadores.
b) Os termos processo e programa são sinônimos.
c) Todo processo utiliza o mesmo espaço de endereço.
d) A qualquer dado instante somente um processo pode executar instruções em um computador com qualquer tipo de arquitetura.
e) Um processo não pode ter um número zero de processos-pai.

Comentários:
Em um sistema multiprocessado é possível a execução simultânea de uma tarefa por processador.
(Resposta A)

Questão 16

[FCC – DPE/SP – 2013] Em sistemas com multiprogramação, inicialmente um novo processo é inserido na fila de prontos. Este processo aguarda até ser selecionado para execução ou ser despachado. Uma vez que o processo seja alocado à CPU, eventos podem ocorrer. Analise as afirmativas sobre o processo.
I. pode emitir uma solicitação de I/O e então ser inserido em uma fila de I/O.
II. pode criar um novo subprocesso e esperá-lo terminar.
III. pode ser removido forçosamente da CPU, como resultado de uma interrupção, e ser devolvido à fila de prontos.
Está correto o que se afirma em

a) I, apenas.
b) II, apenas.
c) III, apenas.
d) I e II, apenas.
e) I, II e III.

Comentários:
Bloqueado (wait): o processo está esperando algum evento externo ou por algum recurso de I/O para poder prosseguir seu processamento. Obs: Essa situação ocorre nas assertivas I e II da questão.
Preempção é o mecanismo usado pelo SO para remover "forçosamente" um processo em execução com a finalidade de ceder a CPU a um outro processo da fila.
(Resposta E)

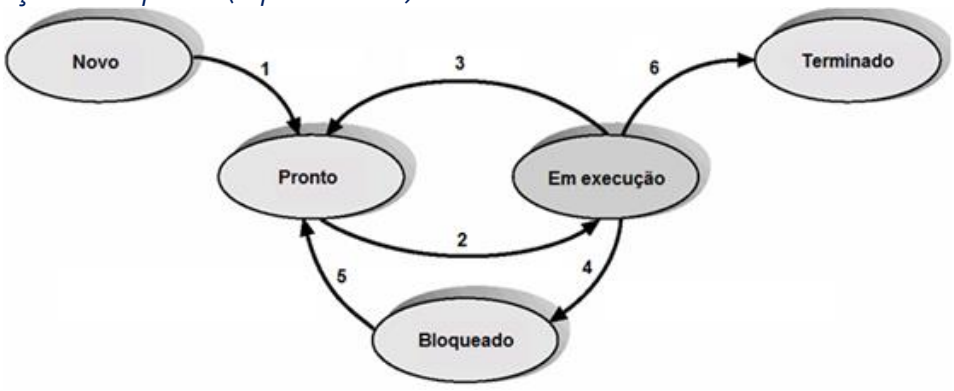
Questão 17

[FCC – TJ/PA – 2009] Um processo em um sistema operacional pode mudar para o estado de pronto a partir do estado de

a) criação, apenas.
b) execução, apenas.
c) criação ou término, apenas.
d) execução ou término, apenas.
e) criação, espera ou execução.

Comentários:

*A figura ilustra que um processo pode ir para o estado de **pronto** quando se encontra nos estados de novo (durante a criação), em execução ou bloqueado (espera ou wait).*



(Resposta E)

Questão 18

[FCC – DPE/AM – 2018] Um sistema operacional moderno prevê uma maneira de um processo dividir a si mesmo em outras tarefas, as quais podem ser executadas concorrentemente. Essas novas tarefas são denominadas

a) interrupções.
b) Kernel.
c) máquina virtual.
d) Servlets.
e) Threads.

Comentários:

Threads são considerados processos leves (mini processos), ou seja, um "tipo de processo" dentro de um outro processo. Podem ser executados concorrentemente ou simultaneamente, a depender do tipo de SO e da quantidade de CPUs disponíveis no host.

(Resposta E)

Questão 19

[FCC – Prefeitura de Teresina/PI – 2016] Considere uma aplicação com 4 threads. Em um sistema

a) multicore concorrência significa que a execução dos threads será intercalada com o passar do tempo.
b) com vários núcleos concorrência significa que os threads podem ser executados em paralelo.
c) em que cada thread só pode ser executado por vez, o sistema operacional é multicore.
d) multicore os threads podem ser atribuídos aos núcleos simultaneamente, somente se houver 4 núcleos, um para cada thread.
e) com um único núcleo à aplicação multithread provocará deadlock ou starvation.

Comentários:

A questão apresenta uma pequena inconsistência, porém, com paciência e análise é possível chegar a resposta do gabarito. Em hardwares equipados com uma única CPU, cada thread é processado de forma "aparentemente

simultâneo", pois a mudança entre um thread e outro é feita de forma tão rápida que para o utilizador isso está acontecendo paralelamente. Em hardwares com múltiplas CPUs ou multicores, os threads são realizados realmente de forma ***simultânea****.*

(Resposta B)

Questão 20

[FCC – TRE/SP – 2012] Nos conceitos de ciência da computação, um processo é um módulo executável que pode conter threads. Um conceito importante sobre threads que estão contidas no mesmo processo é que

a) podem compartilhar a memória do processo.
b) evitam que processos concorrentes acessem recursos do processo principal.
c) acessam a memória estática ao contrário de processos convencionais.
d) controlam o acesso da memória dinâmica (heap) dos processos externos.
e) possuem proteção contra problemas comuns de processos, como vazamento e acessos inválidos da memória.

Comentários:
Esse é um tipo de questão muito recorrente em concursos públicos. Os quadros resumos abaixo permitem com clareza identificar a resposta correta.

Quadro Resumo 4 – privativos de cada *thread*
- Conjunto de registradores e informações de controle.
- Espaço da pilha na memória.

Quadro Resumo 5 – compartilhados entre as *threads*
- Recursos do SO.
- Espaço de endereçamento (especialmente, a seção de códigos e a seção de dados).

(Resposta A)

Questão 21

[FCC – TJ/PE – 2012] As threads de usuário são admitidas acima do kernel e gerenciadas sem o suporte do kernel, enquanto as threads de kernel são admitidas e gerenciadas

a) exclusivamente pela memória RAM.
b) diretamente pela unidade central de processamento.
c) diretamente pelo sistema operacional.
d) exclusivamente por uma deadlock específica.
e) exclusivamente pelo IPL ou boot.

Comentários:
As threads podem ser classificadas em: threads de usuário e de núcleo.
Nas threads de ***usuário*** *a abstração é criada por um conjunto de rotinas de biblioteca utilizada pelo próprio processo. Já nas threads de núcleo a abstração é criada pelo núcleo do sistema operacional.*

(Resposta C)

REFERÊNCIAS:

MACHADO, Francis B.; MAIA, Luiz P. **Arquitetura de Sistemas Operacionais**. 5.ed. [S.l.]: LTC, 2013.

SILBERSCHATZ, A., GALVIN, P.B. e GAGNE, G. **Fundamentos de Sistemas Operacionais** – 9. ed. – Rio de Janeiro: LTC, 2015

TANENBAUM, Andrews S. **Sistemas Operacionais Modernos**. 3. ed. São Paulo: Pearson, 2010.

TANENBAUM, Andrew S.; WOODHULL, A. S. **Sistemas Operacionais: projeto e implementação**. 3. ed. Porto Alegre: Bookman, 2008.