

AULA 00: Introdução aos Sistemas Operacionais

Sumário

1. Apresentação.....	3
1.1. A Banca.....	3
1.2. Metodologia das aulas.....	3
3. O que é um Sistema Operacional ?	4
4. O que é um Sistema Operacional faz ?.....	5
5. O que compõe um Sistema Operacional ?	7
6. Como o sistema operacional é inicializado	9
7. Visão geral do funcionamento de um sistema computacional	14
8. System Call e Interrupções.....	16
9. Cluster Beowulf.....	19
10. Modo Núcleo e Modo Usuário.....	20
11. Arquiteturas de Núcleo.....	20
12. Processos	23
13. Pipes.....	24
14. Questões Comentadas	28
15. Gabarito.....	32

Olá Pessoal!

Eu sou o Prof. Almeida Júnior. Durante muitos anos fui concurseiro como você e sei exatamente o que você está passando. Por isso, quando escrevo um curso, escrevo focado nas reais necessidades de um concurseiro. Meu objeto é um só: aprovar você.

Atualmente sou Auditor Federal de Finanças e Controle na Controladoria Geral da União (CGU). Desde que entrei na CGU tenho atuado em diferentes equipes da TI. Em relação a minha formação, sou engenheiro eletrônico, tenho pós-graduação e mestrado na área de TI. Apesar dessa base acadêmica ser importante para a construção das minhas aulas, considero muito mais importante a minha experiência fazendo provas de concursos. Assim, acredito que o grande valor das minhas aulas é que elas terão foco no que é necessário para ser aprovado. Não estamos aqui somente para aprender a matéria. Nosso objetivo é que você acerte todas as questões da prova e seja aprovado.

Qualquer dúvida você pode entrar em contato comigo pelo email:
professor.almeidajunior@gmail.com

1. Apresentação.

1.1. A Banca.

Vamos utilizar questões de diversas bancas

1.2. Metodologia das aulas.

As aulas serão de teoria e exercícios. A teoria será concisa, ou seja, terá o necessário e somente o necessário para você acertar as questões. O material também terá completude, ou seja, você não precisará recorrer a outro material. Tudo que você precisa estará nesse curso. Por fim, as questões servirão como treino e fixação.

2. O que é um Sistema Operacional ?

A primeira vista parece ser simples definir o que é o Sistema Operacional e que ele faz. Contudo, como veremos, nem sempre é possível delimitar uma fronteira clara. De qualquer forma, não há dúvida que um Sistema Operacional é um programa. O diferencial é que ele é um dos programas mais complexos que rodam na sua máquina. Um sistema operacional é responsável pelo gerenciamento do hardware do seu computador. É ele que vai dizer, por exemplo, qual processo (programa em execução) poderá utilizar o processador e por quanto tempo. Também é papel do SO garantir que um programa não interfira no outro. Ou seja, o SO atua como um grande gerente de recursos.

Podemos ver um sistema operacional com uma camada entre o hardware e as aplicações de usuário. Dessa forma, o propósito de um sistema operacional é criar um ambiente que permita a execução de forma conveniente, ou seja, simples e ao mesmo tempo eficiente.

É importante também notar que existe um vasto espectro de sistemas operacionais. Por exemplos, alguns são responsáveis por gerenciar o hardware de PC's, outros de celulares e ainda existem SO's para Mainframes.

É importante entender de forma clara o que significa uma execução conveniente de programas. Note que se o SO não existisse seria papel do programador lidar diretamente com hardware, além disso, ele deveria prever a existência e competição com outros programas. Ademais, para cada novo sistema (hardware) provavelmente seria necessária uma nova codificação para adaptar o programa ao novo hardware. Dessa forma, podemos dizer que o sistema operacional cria uma abstração para o programador.

Segundo Tanenbaum, um SO pode ser visto como **uma máquina estendida** ou como **um gerenciador de recursos**. Como uma máquina estendida, o usuário não está interessado em saber os detalhes funcionais dos dispositivos. Vamos tomar como exemplo o acesso aos dados de disco rígido. Uma abstração típica seria que o disco contenha um conjunto de nomes de arquivos. Note que no fundo o que existem são 0's e 1's. Um sistema de arquivos é uma abstração criada pelo SO. O usuário não precisa se preocupar com detalhes com em que posição está o cabeçote de leitura do HD. Dessa forma, o SO **esconde do usuário a complexidade do hardware e apresenta uma visão fácil e simples para as operações sobre**

os dispositivos. Essa visão é equivalente a **uma máquina estendida ou máquina virtual**, mais fácil de lidar.

Na outra visão de Tanenbaum, o SO é um gerenciador de recursos. Lembre-se que o sistema computacional é conjunto de recursos: processadores, memórias, discos, mouses, teclados, impressoras, etc. É tarefa do SO gerenciar a alocação desses dispositivos para os diversos processos, de forma que não aconteçam inconsistências.

3. O que é um Sistema Operacional faz ?

Do ponto de vista do usuário, um PC consiste basicamente de teclado, monitor, mouse e um sistema operacional. Para o usuário, o sistema operacional tem o papel de maximizar o desempenho da máquina. Provavelmente, o usuário desejará abrir diversas abas do navegador, ouvir música, e talvez jogar ao mesmo tempo. Dessa forma, do ponto de vista do usuário, o sistema operacional tem como função principal facilitar o uso. Ou seja, abstrair a complexidade do hardware para o usuário e entregar um ambiente fácil de utilizar e útil. Em muitos sistemas, mesmo com apenas uma CPU, o sistema operacional é capaz de criar a ilusão que as diferentes tarefas estão sendo executadas ao mesmo tempo. Note que quando há apenas uma CPU isso é de fato uma ilusão, já que somente um processo (programa em execução) poderá estar de posse da CPU durante um determinado tempo. Essa ilusão é chamada de multitasking ou multitarefa.

Do ponto de vista do sistema computacional, o SO é grande maestro. Ele é responsável por alocar os recursos da máquina. Perceba que uma máquina tem diversos recursos: memória, tempo de CPU, dispositivos de I/O, etc. É papel do SO dizer qual aplicação vai utilizar qual recursos e por quanto tempo. Vamos pegar como exemplo a impressora. Note que se todos os programas pudessem enviar livremente caracteres para sempre impressos, os textos resultados seriam uma mescla de diferentes arquivos. Para que isso não ocorra, o SO controla o acesso a esse hardware criando um fila.

01. (2014/VUNESP/Analista de Sistemas) Existe um tipo de sistema operacional que fornece ao usuário a ilusão de que o número de processos que são executados simultaneamente no computador é superior ao número de processadores existentes no computador. Esse tipo de sistema operacional é conhecido como

- a) escalonável.
- b) multiexecução.
- c) multitarefa.
- d) preemptivo.
- e) round robin.

Comentários:

Como vimos, essa ilusão criada pelo SO é chamada de multitarefa. Os demais termos serão explicados em aulas posteriores.

4. O que compõe um Sistema Operacional ?

Nem sempre é possível dizer o que compõe um sistema operacional. Não há dúvida que o Kernel (núcleo do sistema operacional) faz parte do SO. Contudo, nem sempre é fácil dizer o que faz parte do SO. Por exemplo, um navegador é parte do SO ou é um aplicativo? Talvez de forma apressada você pense que é aplicativo e ponto final. Contudo, essa questão foi até os tribunais nos Estados Unidos. A Microsoft passou a distribuir o seu navegador de forma gratuita junto com a instalação do sistema operacional. A Netscape foi a justiça alegando que essa prática era monopolista e ilegal. Então o que vem “na caixa” quando compro o Windows, nem tudo é o SO? E os programas de Shell (interpretadores de comandos) são parte do SO ou apenas um outro programa. Talvez você já tenha ouvido falar no Chrome OS. Esse sistema operacional da google, baseado em Linux, usa um navegador web (Google Chrome) como interface principal com o usuário. E agora, essa “interface navegador” faz parte do SO? De qualquer forma, não há dúvida que há um conjunto de atividades que fazem parte do SO, esse conjunto de atividades é realizada pelo que chamamos núcleo do sistema operacional ou simplesmente Kernel. O Kernel é um programa que está rodando o tempo todo no computador. Além do Kernel, temos outros dois tipos de programas: programas de sistema e programas de aplicação. Programas de sistema estão ligados ao sistema operacional, tem uma relação muito próxima, mas não fazem parte necessariamente do Kernel. Um exemplo clássico de programa de sistema é um compilador. Os programas aplicativos são os demais programas. Eles têm uma aplicação prática para o usuário. Por exemplo, o word ou o excel.

02. (2005/ ESAF/ Receita Federal) Em um sistema operacional, o kernel é

- a) um computador central, usando um sistema operacional de rede, que assume o papel de servidor de acesso para os usuários da rede.**
- b) a técnica usada para permitir que um usuário dê instruções para a máquina, usando instruções gráficas.**
- c) o processo de intervenção do sistema operacional durante a execução de um programa. Tem como utilidade desviar o fluxo de execução de um sistema para uma rotina especial de tratamento.**

d) o núcleo do sistema, responsável pela administração dos recursos do computador, dividindo-os entre os vários processos que os requisitam. No caso do Linux, o Kernel é aberto, o que permite sua alteração por parte dos usuários.

e) um pedido de atenção e de serviço feito à CPU.

Comentários:

a) Esse é um servidor de autenticação

b) GUI

c) Tratamento de interrupções

d) Gabarito

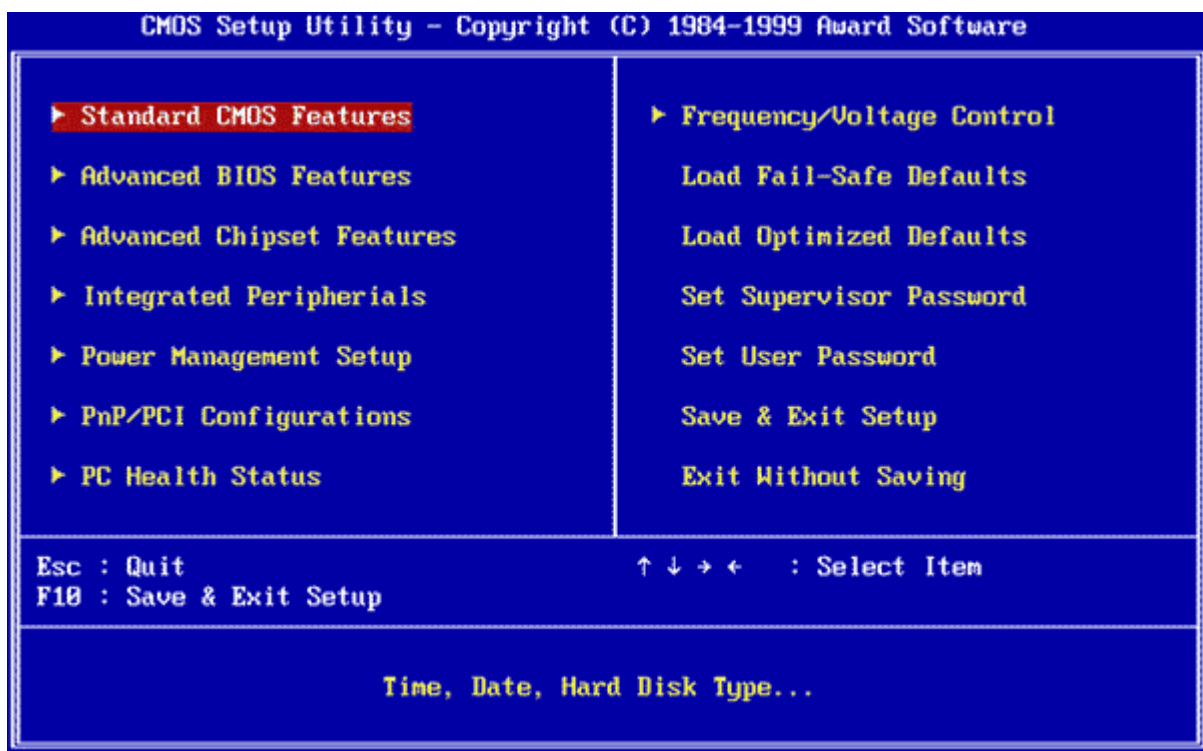
e) Interrupção

5. Como o sistema operacional é inicializado

Já vimos que o principal programa de um sistema operacional é o Kernel, mas como o kernel é carregado na memória. Ora, lembre-se que os programas são arquivos que estão armazenados, geralmente, no HD. Logo, uma questão que deve ser respondida é como o SO é inicializado. Ou seja, quando você liga o computador, como o HW sabe onde está o kernel?

Bem, quando ligamos ou resetamos a nossa máquina uma série de processos é executada antes que ele esteja pronta para rodar o kernel. Esse processo é chamado de **boot**. Na verdade, boot é um nome mais curto para **bootstrap**.

O processo de boot é guiado por um software gravado em um chip de memória. Essa memória pode ser uma memória ROM ou EEPROM. Nos computadores mais modernos esse chip de memória é geralmente flash. O programa gravado nesse chip é chamado de BIOS (Basic Input-Output system). Você pode configurar alguns parâmetros entrando na "BIOS". Para tanto, geralmente, você deve apertar DEL durante o boot (ou outra tecla pode ser utilizada, por exemplo F1) para ter acesso a tela de configuração. A essa tela tem a aparência da figura a seguir.

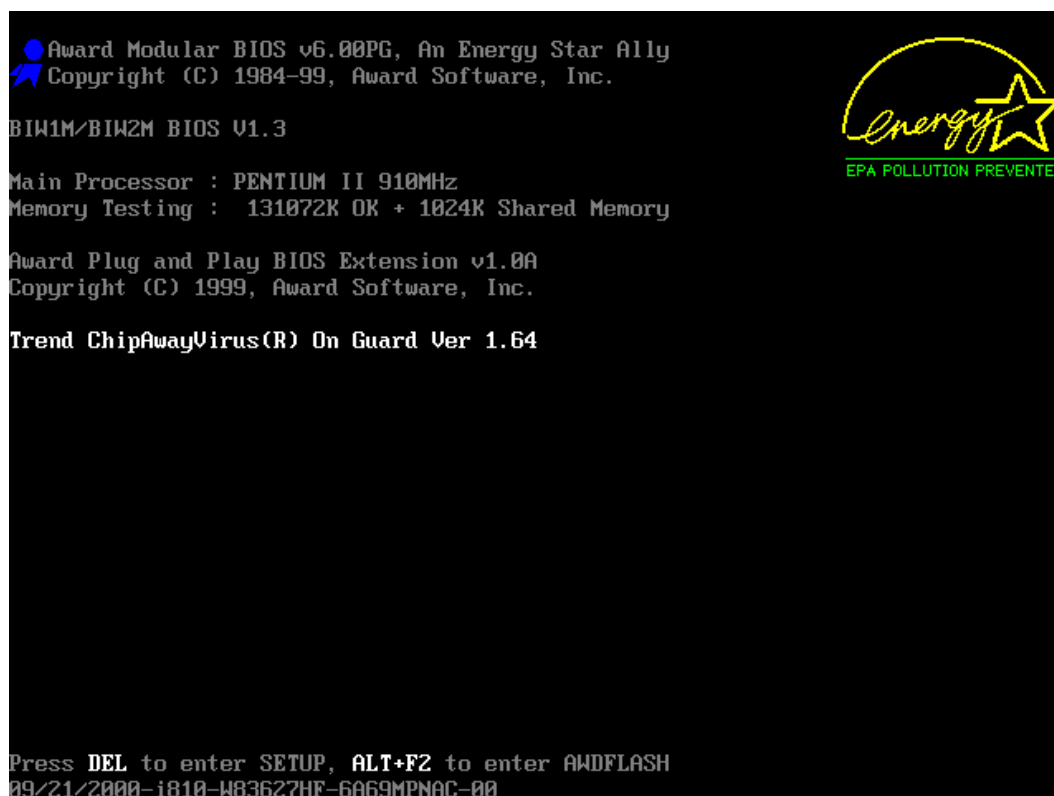


O programa gravado nesse chip é responsável, por exemplo, por identificar quantos HD's sua máquina possui e qual deles será o mestre, o disco de boot, etc. Para ser mais preciso, essa "parte" do programa de configuração da BIOS é chamado de SETUP.

Aqui cabe um alerta. Para ser mais didático, podemos até dizer que existem três programas: BIOS, SETUP e POST. Na verdade, estão todos gravados na mesma memória flash. Tudo no fundo é o "BIOS" ou o "bootstrap". Portanto, não seja rigoroso em questões de prova. Tente saber o que o examinador está querendo como resposta.

Uma vez que o PC é ligado, a primeira etapa do boot é executar o POST (power on self test). O POST é um programa que faz parte da BIOS (ou seja é uma etapa do boot/bootstrap) e é responsável por verificar erros e defeitos no HW. Por exemplo, se houver um conflito na memória ou se ela estiver mal encaixada ou ainda com defeito, provavelmente você ouvirá beeps. O padrão de beeps permite você identificar o tipo de problema que está ocorrendo.

Se não houver nada de errado, tipicamente são mostradas na tela informações do HW. Por exemplo, quantidade de memória RAM, fabricante da BIOS, etc. A figura a seguir ilustra essa etapa.



Após o POST, ocorre o acesso ao primeiro setor do disco (MBR - Master Boot Sector) que foi definido como boot disk (essa definição pode ser alterada no SETUP). Neste setor, espera-se encontrar um **bootstrap loader**, **bootstrap** ou ainda **boot loader** específico para cada sistema operacional. Por exemplo, no windows 2000, o boot loader é chamado de NTLDR.

É importante entender que o bootstrap é o programa responsável lançar o sistema operacional. Depois de carregado na memória, o SO passa a controlar o hardware do computador. Nos sistemas modernos existem os chamados boot loader de segundo estágio. Por exemplo, o GRUB. Outro exemplo no Linux seria o LILO. Esses programas podem ser capazes de fornecer ao usuário múltiplos boots. Ou seja, o usuário pode carregar mais de um SO a partir deste ponto (geralmente selecionando a partir de uma lista). Veja a figura a seguir.



Lembre-se também que o processo de boot é um termo curto para bootstrap. Assim, não se assuste se um autor associar todo o processo descrito ao bootstrap (ou ao bootstrap loader). Outro ponto importante para prova é que podemos afirmar que o bootstrap (boot em sentido amplo) não faz parte do SO.

03. (FCC/2007) NÃO é um componente dos sistemas operacionais:

- a) bootstrap.
- b) scheduler.
- c) kernel.
- d) shell.
- e) GUI.

Comentários:

a) O bootstrap não faz parte do SO. Ele é responsável por carregar o SO na memória, uma vez carregado, o SO passa a controlar o hardware da máquina. Portanto, é nosso gabarito.

b) O scheduler é parte do SO responsável por agendar os processos. Ou seja, qual processo (programa em execução) terá posse da CPU e por quanto tempo.

c) O kernel é a parte que do SO que está em constante execução. Tem o as principais funções do sistema operacional

d) Shell é um interpretador de comandos. Apesar de não fazer parte do kernel, o shell está intimamente ligado ao SO.

e) Gui faz referência a interface gráfica fornecida pelo sistema operacional,

04. (FCC/2010) Algumas Regras Gerais de Uso do Computador

I. Sempre feche todas as aplicações abertas antes de instalar qualquer software. Para um cuidado extra, faça uma nova inicialização (boot) e feche todas as aplicações que são inicializadas automaticamente antes de você instalar algum programa. Sempre desligue seu computador antes de plugar ou desplugar qualquer tipo de equipamento. [...]

No item I, a recomendação para fazer uma nova inicialização (boot), significa que o computador executará uma série de tarefas, acionando primeiramente

- a) o SO (sistema operacional).
- b) a memória RAM.
- c) o disco rígido (HD).

d) a BIOS (Sistema Básico de Entrada e Saída).

e) o programa SETUP.

Comentários:

O primeiro programa que acesso no seu computador é o bootstrap. Esse programa está em chip flash. De forma mais abrangente, é o que chamamos de BIOS. Portanto, gabarito D.

05. (CESPE/2011/PC-ES) O carregamento (boot) do sistema operacional Linux pode ser gerenciado pelo programa LILO

Comentários:

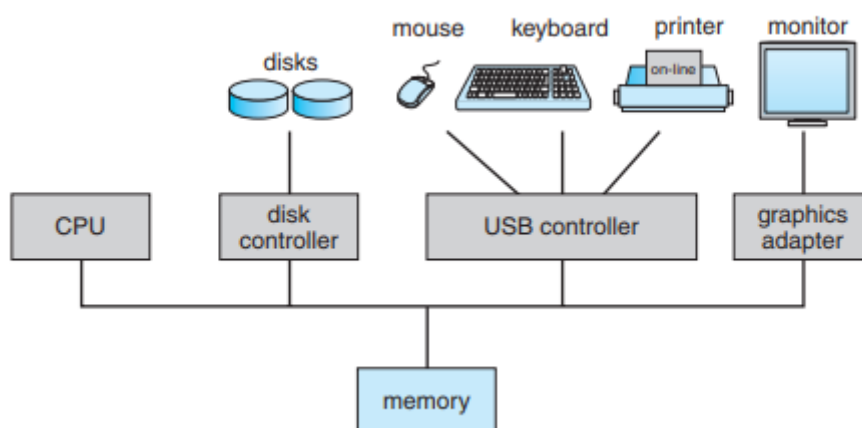
LILO é um acrônimo para a expressão inglesa Linux LOader que, em português significa carregador de linux. Ele permite configurar o boot de múltiplos sistemas operacionais na mesma máquina.



6. Visão geral do funcionamento de um sistema computacional

Um sistema computacional é tipicamente formado por mais de uma CPU (ou núcleos), memória principal e vários dispositivos de entrada e saída. Para cada um desses dispositivos, existe uma controladora. A CPU e as controladoras podem executar em paralelo. Logo, elas vão competir por ciclos de memória. Esse fato impõe a necessidade do gerenciamento do acesso a memória compartilhada. Para controlar os dispositivos de I/O os fabricantes fornecem código específicos para um determinado sistema operacional. Esses códigos são chamados de “drivers”. Assim, por exemplo, quando instalamos uma impressora, além da instalação física, é necessário instalar o respectivo “driver”.

A figura seguir mostra de forma esquematiza um sistema computacional moderno típico.



06. (2012/VUNESP/TJ-SP/Analista de Sistemas) Analise as afirmações sobre os componentes de um sistema operacional típico.

I. O núcleo é componente responsável pela gerência dos recursos de hardware usados pelas aplicações de um computador.

II. Os módulos de código específicos para acessar os dispositivos físicos de um computador são conhecidos como drivers.

III. Os programas utilitários facilitam o uso do sistema computacional, fornecendo funcionalidades complementares ao núcleo, como a formatação de discos.

Sobre as afirmações, está correto o contido em

a) I, apenas.

b) II, apenas.

c) I e II, apenas.

d) I e III, apenas.

e) I, II e III.

e) I, II e III.

Comentários:

I. Correto. O kernel é software que está sempre rodando. Ele é responsável pelas funções principais do sistema operacional. Dentre elas ser um grande maestro da utilização dos recursos de hardware.

II. Correto. Um driver é um código, geralmente escrito pelo fabricante do I/O, que permite que o SO possa controlar e gerenciar o I/O.

III. Correto. Além do Kernel, temos outros dois tipos de programas: programas de sistema e programas de aplicação. Programas de sistema estão ligados ao sistema operacional, tem uma relação muito próxima, mas não fazem parte necessariamente do Kernel. Os programas aplicativos são os demais programas. A rigor um programa que faz formatação de disco deveria ser considera um programa de sistema, mas o examinador não foi tão criterioso. Chamou simplesmente de programa de aplicação, os programas que não fazem parte do Kernel. Talvez se alguém tivesse entrado com recurso o gabarito poderia ter sido alterado.

Gabarito E

7. System Call e Interrupções

Como vimos, o processo de boot é responsável por encontrar o código do SO (Kernel) no HD e colocar na memória RAM. Uma vez que o SO está carregado, o controle passa a ser do sistema operacional. O SO é um programa capaz de fornecer serviços. Parte desses serviços não são fornecidos diretamente pelos Kernel, mas por programas de sistema. Tais programas também são chamados de daemons. Os daemons rodam durante todo o tempo que o kernel está rodando. No UNIX, o primeiro processo que é iniciado é o “init”. Ele é responsável por inicializar outros daemons.

Uma vez que o kernel está carregado e também os daemons, o SO está pronto para “atender” as requisições dos seus usuários. Nesse momento, o SO aguarda por um evento. Para ter conhecimento do evento é utilizada uma interrupção. As interrupções podem ser geradas diretamente pelo hardware ou via software. As interrupções de hardware são sinais enviados, por meio do barramento, para a CPU. Já as interrupções de software são geradas por uma “System Call” (chamada de sistema).

Uma system call (chama de sistema) também é conhecida como “monitor call”. As chamadas de sistemas (system calls) **constituem a interface entre um programa do usuário e o Sistema Operacional**. Note que uma system call é um meio de comunicação entre um programa e os serviços oferecidos pelo sistema operacional. Por exemplo, se um programa deseja escrever um arquivo no disco, ela fará isso por meio de uma system call. Dito de outro forma, uma chamada de sistema é uma porta de entrada para acesso ao núcleo do sistema, que contém suas funções. Sempre que o usuário necessitar de algum serviço, o solicita através de uma chamada de sistema definida e específica. Cada serviço disponível por um determinado Sistema Operacional possui uma chamada de sistema associada e cada SO possui seu próprio conjunto de chamadas. As chamadas de sistemas, de fato, servem também para proteger o núcleo do Sistema Operacional, tentando impedir que um determinado programa realize alguma operação que altere a integridade do sistema.

Quando uma interrupção é gerada, a CPU passa a executar um código específico que trata aquela interrupção. Ou seja, o fluxo de execução é direcionado para um endereço que possui um rotina para tratar a interrupção. Para saber o local (endereço) onde está o código que trata cada interrupção é utilizado um conjunto de ponteiros que chamamos de **vetor de**

interrupções. Ou seja, o vetor de interrupções é uma tabela de endereços de memória que apontam para as rotinas de tratamento de interrupções. Geralmente, esse vetor ocupa as posições mais baixas do espaço de memória (ou seja, encontram-se nos primeiros bytes).

07. (2009/ UFF/ Analista de Tecnologia da Informação) No tocante aos sistemas operacionais, quando o usuário deseja solicitar algum serviço do sistema, ele realiza uma chamada a uma de suas rotinas (ou serviços) por meio de "artifícios" que são a porta de entrada para o acesso ao núcleo do sistema operacional. Esse artifício ou acesso é conhecido como:

- a) bibliotecas;**
- b) sub-rotinas;**
- c) system calls;**
- d) kernel;**
- e) buffer.**

Comentários:

a) Errado. Bibliotecas é um termo genérico. Aqui o examinador está se referindo a códigos que pode ser reutilizados por diferentes programas.

b) Errado. De forma geral sub-rotina nada mais é quem uma parte do código que está encapsulado em um código maior e pode ser chamada. É uma espécie de micro-programa dentro do programa principal.

c) Correto. As chamadas de sistema são interrupções via software que permite o acesso aos serviços do sistema operacional. Por exemplo, abrir um arquivo em disco.

d) Errado. Kernel é o núcleo do SO, ou seja, o programa que roda o tempo todo na memória e provê as principais funcionalidades de um sistema operacional

e) Errado. Buffer é termo genérico para uma memória temporária. Geralmente o buffer serve para acumular dados prévios.

08. (2011/VUNESP/ Administrador de Banco de Dados) Na estrutura organizacional do sistema operacional Linux, a função do System Call é

- a) descobrir as aplicações em execução no sistema.
- b) prover a interface entre a aplicação e o Kernel.
- c) realizar a chamada de aplicações de rede.
- d) realizar a comunicação entre os dispositivos e o Kernel.
- e) sincronizar as aplicações multitarefa.

Comentários:

No Linux, ou em qualquer outro sistema operacional, uma chamada de sistema permite ao programador acessar os serviços do sistema operacional. Assim, ela é interface entre o programa (aplicação) e o núcleo do SO que fornece efetivamente o serviço. Portanto, gabarito B.

8. Cluster Beowulf



Um cluster Beowulf é um conjunto de dois ou mais computadores independentes, interligados via rede, que trabalham em conjunto trocando informações entre si em torno de uma única tarefa. Normalmente são utilizando computadores pessoais conectados por meio de uma rede local. Também é comum que o sistema operacional seja o Linux. Nesse cluster existe um computador principal. Essa máquina é responsável por controlar todo o cluster e pela distribuição de tarefas.

09. (2004/ESAF/CGU) Computadores de baixo custo podem ser agrupados em clusters, onde exercem funções que exigem alto desempenho computacional como se fossem uma única máquina de grande porte. É correto afirmar que um cluster Beowulf com o sistema operacional Linux é constituído de vários nós escravos,

- a) sendo que cada nó exerce o controle sobre todos os demais nós.
- b) sendo que cada nó exerce o controle sobre seu nó vizinho e o último nó exercerá o controle sobre o primeiro nó.
- c) sem a necessidade de controle.
- d) sendo que cada nó exerce seu próprio controle.
- e) controlados por um computador principal.

Comentários:

Existe um nó na rede que gerencia o cluster. Portanto, gabarito E.

9. Modo Núcleo e Modo Usuário

Já sabemos que o sistema operacional é formado por um conjunto de rotinas (kernel) que oferece serviços aos usuários e suas aplicações. Por razões de segurança e estabilidade, os sistemas operacionais restringem as operações executadas pelas aplicações. Por exemplo, restrições de acesso a dispositivos de hardware (HD, memória, etc). Para tanto, existem dois modos de acesso. O modo de acesso é um mecanismo presente no hardware dos processadores. No modo usuário, uma aplicação só pode executar instruções **não privilegiadas**, ou seja, instruções que não oferecem riscos ao sistema. Já no modo Kernel (também chamado de modo supervisor), uma aplicação pode executar também instruções privilegiadas, ou seja, instruções que oferecem risco ao sistema, tais como instruções que acessam dados no disco. Perceba que as chamadas de sistemas possuem no seu código instruções privilegiadas, logo o processador deve estar em modo kernel para executá-las.

10. Arquiteturas de Núcleo

O projeto de um sistema operacional depende muito do hardware a ser utilizado e do tipo de SO que se deseja construir (tempo compartilhado, tempo real, etc). A maneira como o código do sistema é organizado e o inter-relacionamento entre os seus diversos componentes pode variar conforme a concepção do projeto. As principais arquiteturas dos são:

- Arquitetura monolítica
- Arquitetura em camadas
- Gerência de Máquinas virtuais
- Arquitetura microkernel

Na arquitetura monolítica, os vários módulos que são compilados separadamente e depois linkados **formando um único programa executável**. A desvantagem é que a manutenção é bastante difícil. Por outro lado, o projeto é mais simples e permite um bom desempenho.

Na arquitetura em camadas, o sistema é dividido em níveis sobrepostos, onde cada camada oferece um conjunto de funções que podem ser utilizadas apenas pelas camadas superiores. As camadas mais internas são mais privilegiadas que as externas. Tem como vantagem criar

uma hierarquia de níveis de modos de acesso. Também facilita a depuração e manutenção. Tem como desvantagem o desempenho.

Na gerência de máquinas virtuais, cria-se um nível intermediário entre o hardware e o sistema operacional. São criadas máquinas virtuais independentes, cada uma oferece uma cópia virtual do hardware. Como cada VM é independente, é possível que cada VM possua seu próprio sistema operacional e que os usuários executem aplicações como se estivesse dedicado a cada um deles. A vantagem dessa abordagem é o isolamento total entre cada VM, oferecendo grande segurança para cada uma delas. A desvantagem é necessidade de uma gerencia dessas máquinas virtuais, já que vão compartilhar os mesmos recursos do hardware reais.

Na arquitetura de microkernel, a ideia é tornar o núcleo do SO o mais simples possível. Os serviços do sistema são disponibilizados através de processos, responsáveis por oferecer um conjunto específico de funções (gerência de arquivos, processos, de memória e escalonamento). Sempre que uma aplicação deseja algum serviço, deve solicitar ao processo responsável. A aplicação que solicita serviço é chamada de cliente e o processo que responde é denominado de servidor. A principal função do núcleo é realizar a troca de mensagens entre cliente e o servidor. Tem como vantagem a manutenção, escalabilidade, flexibilidade e portabilidade. A desvantagem é que sua implementação é difícil.

10. (2012/ ESAF/Analista de Sistemas) Na arquitetura Microkernel

- a) permite-se isolar as funções do sistema operacional por diversos processos servidores pequenos e dedicados a serviços específicos.**
- b) o sistema operacional passa a ser de difícil manutenção.**
- c) nunca há necessidade de acesso direto ao hardware, como operações de entrada/saída.**
- d) permite-se agrupar as funções do sistema operacional por diversos processos clientes pequenos e dedicados a servidores espontâneos.**
- e) confere-se escalabilidade reduzida ao sistema operacional.**

Comentários:

- a) Correto. No microkernel, utiliza-se o conceitos de processo servidor e cliente.

- b) Errado. A manutenção é facilitada
- c) Errado. A arquitetura que blindaria o SO do acesso ao hardware real seria a de máquinas virtuais.
- d) Errado. Quem fornece os serviços (funções) do sistema operacional são os processos servidores
- e) Errado. A arquitetura é escalável.

11. Processos

Um processo é um programa em execução. Ou seja, é um código que está carregado e memória e está sendo executado pelo processador. Esse termo é útil para diferenciar do código armazenado no HD. Quando compilamos um código fonte, um executável é criado. Esse arquivo que fica no HD chamamos de programa. Note que um mesmo programa pode ser carregado em diferentes instâncias na memória. Imagine que exista um programa que é uma calculadora é possível “abrir” dois desses programas. Note que na memória existiram duas cópias do mesmo código (programa), essas entidades que existem na memória é que chamamos de processos. Note que cada processo tem um espaço de endereçamento na memória, ou seja, ele ocupa determinadas posições na memória. Um processo pode dar origem (criar) novos processos. O processo que cria novos processos é chamado de pai e os criados são chamados de filhos. Claro que esse processo pode ser recursivo, ou seja, os processos filhos também podem criar seus próprios filhos.

12. Pipes

Um pipe é um pseudoarquivo utilizado para conectar dois processos. Assim, um pipe pode ser utilizado como um meio de comunicação entre dois processos. Um processo escreve no pipe como se ele fosse um arquivo de saída. Do mesmo modo, o outro processo envolvido lê os dados do pipe como se fosse um arquivo de entrada. Note que isso é extremamente útil, principalmente para quem lida com longos comandos no shell. Se o pipe não fosse possível, seria necessário gerar efetivamente um arquivo para que o outro processo pudesse então ler desse arquivo.

11. (2013/ FCC/ TRT - 5ª Região) É um tipo de pseudoarquivo que pode ser usado para efetuar comunicação entre dois processos. Se um processo A pretende enviar dados para o processo B, o processo A escreve em um lado (do mesmo modo que estivesse escrevendo em um arquivo) e o processo B poderá ler os dados como se estivesse lendo de um arquivo de entrada. A este pseudoarquivo dá-se o nome de

- a) channel.
- b) pipe.
- c) queue.
- d) thread.
- e) stdout.

Comentários:

Como vimos, um pipe facilita a comunicação entre dois processos. Ele se comporta com um arquivo que pode ser escrito por um processo e lido por outro.

12. (2009/ UFF/ Analista de Tecnologia da Informação) No Linux, os processos se comunicam por meio de um mecanismo de troca de mensagens utilizando canais de comunicação. Esse mecanismo é conhecido como:

- a) shell;
- b) pipes;
- c) PID;

d) i-node;

e) cluster

Comentários:

Como vimos, uma das formas dos processos se comunicarem é por meio de pipes. Vamos comentar os demais itens.

a) É o interpretador de comandos

b) Gabarito

c) É um identificador de um processo

d) É um estrutura de dados que descreve um objeto no sistema de arquivos Unix/Linux

e) Conjunto de computadores fortemente ligados que aparentam ser um único sistema.

13. Multiprogramação e Tempo Compartilhado (multitarefa)

Multiprogramação e tempo compartilhado. Esses conceitos estão intrinsecamente ligados ao conceito de sistemas operacionais modernos, de tal forma que podem dizer que fazem parte de sua estrutura. Tipicamente, um programa realiza operações de CPU e I/O. Se um único programa tomasse posse do sistema até o final da sua execução, existiria uma grande desperdício de recursos. Por exemplo, quando um programa (mais precisamente um processo) realiza uma operação de I/O (ex: aguarda a entrada de um dado), a CPU fica ociosa. Assim, cabe ao SO executar outro processo na CPU, enquanto o anterior aguarda ser atendido pelo I/O. Essa estratégia de otimização do uso da CPU é chamada de **multiprogramação**.

Em sistemas multiprogramados, vários processos são mantidos na memória. Aqueles que não podem ser armazenados na memória principal (por limitação de espaço), permanecem em disco (processo de swap/memória virtual). Assim que for possível, o processo sairá do disco e será carregado na memória principal.

A diferença entre sistemas multiprogramados e de tempo compartilhado é sutil, pois os **sistemas de multitarefa** também são multiprogramados (mantêm vários programas na memória e otimizam a utilização da CPU) . Contudo, nos sistemas multitarefa (ou de tempo compartilhado), a CPU alterna entre os processos com uma frequência tão alta que permite a **interação do usuário com programas enquanto eles são executados**. Ou seja, sistemas de tempo compartilhado permitem a comunicação constante entre o usuário e o programa. Eles criam um a ilusão que existe uma máquina está sempre disponível para qualquer processo. Podemos concluir, que **os sistemas de tempo compartilhado nada mais são que uma extensão lógica da multiprogramação**.

13. (ESAF/CVM/2010) São tipos de sistemas operacionais:

a) Sistemas Monusuários/Monopointer, Sistemas Multiusuários/Multipointer, Sistemas com múltiplas entradas.

b) Sistemas Monoprogramáveis/Monotarefa, Sistemas Multiprogramáveis/Multitarefa, Sistemas com múltiplos processadores.

- c) Sistemas Monostakeholder/Monoinstrução, Sistemas Multistakeholder/Multi-instrução, Sistemas com múltiplos processadores.**
- d) Sistemas Monocompiláveis/Monomonitoramento, Sistemas Multicompiláveis/Multimonitoramento, Sistemas com múltiplos usuários.**
- e) Sistemas Monoplanejáveis/Monodesign, Sistemas Multiplanejáveis/Multidesign, Sistemas com processadores de segmentação.**

Comentários

Sistemas monoprogramáveis/monotarefa permitiam que apenas um programa/processo ocupasse o sistema computacional de cada vez. Vimos anteriormente o conceito de sistemas Multiprogramáveis/Multitarefa. Por fim, existem sistemas operacionais que são desenvolvidos para aproveitar computadores com múltiplos processadores. Neste caso, diversos programas podem rodar ao mesmo tempo (um em cada processador) ou partes de um mesmo programa pode ser executados em diferentes processadores.

Gabarito: B

14. Questões Comentadas

01. (2014/VUNESP/Analista de Sistemas) Existe um tipo de sistema operacional que fornece ao usuário a ilusão de que o número de processos que são executados simultaneamente no computador é superior ao número de processadores existentes no computador. Esse tipo de sistema operacional é conhecido como

- a) escalonável.
- b) multiexecução.
- c) multitarefa.
- d) preemptivo.
- e) round robin.

02. (2005/ ESAF/ Receita Federal) Em um sistema operacional, o kernel é

- a) um computador central, usando um sistema operacional de rede, que assume o papel de servidor de acesso para os usuários da rede.
- b) a técnica usada para permitir que um usuário dê instruções para a máquina, usando instruções gráficas.
- c) o processo de intervenção do sistema operacional durante a execução de um programa. Tem como utilidade desviar o fluxo de execução de um sistema para uma rotina especial de tratamento.
- d) o núcleo do sistema, responsável pela administração dos recursos do computador, dividindo-os entre os vários processos que os requisitam. No caso do Linux, o Kernel é aberto, o que permite sua alteração por parte dos usuários.
- e) um pedido de atenção e de serviço feito à CPU.

03. (FCC/2007) NÃO é um componente dos sistemas operacionais:

- a) bootstrap.
- b) scheduler.
- c) kernel.
- d) shell.
- e) GUI.

04. (FCC/2010) Algumas Regras Gerais de Uso do Computador

I. Sempre feche todas as aplicações abertas antes de instalar qualquer software. Para um cuidado extra, faça uma nova inicialização (boot) e feche todas as aplicações que são inicializadas automaticamente antes de você instalar algum programa. Sempre desligue seu computador antes de plugar ou desplugar qualquer tipo de equipamento. [...]

No item I, a recomendação para fazer uma nova inicialização (boot), significa que o computador executará uma série de tarefas, acionando primeiramente

- a) o SO (sistema operacional).
- b) a memória RAM.
- c) o disco rígido (HD).
- d) a BIOS (Sistema Básico de Entrada e Saída).
- e) o programa SETUP.

05. (CESPE/2011/PC-ES) O carregamento (boot) do sistema operacional Linux pode ser gerenciado pelo programa LILO

06. (2012/VUNESP/TJ-SP/Analista de Sistemas) Analise as afirmações sobre os componentes de um sistema operacional típico.

I. O núcleo é componente responsável pela gerência dos recursos de hardware usados pelas aplicações de um computador.

II. Os módulos de código específicos para acessar os dispositivos físicos de um computador são conhecidos como drivers.

III. Os programas utilitários facilitam o uso do sistema computacional, fornecendo funcionalidades complementares ao núcleo, como a formatação de discos.

Sobre as afirmações, está correto o contido em

a) I, apenas.

b) II, apenas.

c) I e II, apenas.

d) I e III, apenas.

e) I, II e III.

e) I, II e III.

07. (2009/ UFF/ Analista de Tecnologia da Informação) No tocante aos sistemas operacionais, quando o usuário deseja solicitar algum serviço do sistema, ele realiza uma chamada a uma de suas rotinas (ou serviços) por meio de "artifícios" que são a porta de entrada para o acesso ao núcleo do sistema operacional. Esse artifício ou acesso é conhecido como:

a) bibliotecas;

b) sub-rotinas;

c) system calls;

d) kernel;

e) buffer.

08. (2011/VUNESP/ Administrador de Banco de Dados) Na estrutura organizacional do sistema operacional Linux, a função do System Call é

a) descobrir as aplicações em execução no sistema.

b) prover a interface entre a aplicação e o Kernel.

c) realizar a chamada de aplicações de rede.

d) realizar a comunicação entre os dispositivos e o Kernel.

e) sincronizar as aplicações multitarefa.

09. (2004/ESAF/CGU) Computadores de baixo custo podem ser agrupados em clusters, onde exercem funções que exigem alto desempenho computacional como se fossem uma única máquina de grande porte. É correto afirmar que um cluster Beowulf com o sistema operacional Linux é constituído de vários nós escravos,

- a) sendo que cada nó exerce o controle sobre todos os demais nós.
- b) sendo que cada nó exerce o controle sobre seu nó vizinho e o último nó exercerá o controle sobre o primeiro nó.
- c) sem a necessidade de controle.
- d) sendo que cada nó exerce seu próprio controle.
- e) controlados por um computador principal.

10. (2012/ ESAF/Analista de Sistemas) Na arquitetura Microkernel

- a) permite-se isolar as funções do sistema operacional por diversos processos servidores pequenos e dedicados a serviços específicos.
- b) o sistema operacional passa a ser de difícil manutenção.
- c) nunca há necessidade de acesso direto ao hardware, como operações de entrada/saída.
- d) permite-se agrupar as funções do sistema operacional por diversos processos clientes pequenos e dedicados a servidores espontâneos.
- e) confere-se escalabilidade reduzida ao sistema operacional.

11. (2013/ FCC/ TRT - 5ª Região) É um tipo de pseudoarquivo que pode ser usado para efetuar comunicação entre dois processos. Se um processo A pretende enviar dados para o processo B, o processo A escreve em um lado (do mesmo modo que estivesse escrevendo em um arquivo) e o processo B poderá ler os dados como se estivesse lendo de um arquivo de entrada. A este pseudoarquivo dá-se o nome de

- a) channel.
- b) pipe.
- c) queue.
- d) thread.

e) stdout.

12. (2009/ UFF/ Analista de Tecnologia da Informação) No Linux, os processos se comunicam por meio de um mecanismo de troca de mensagens utilizando canais de comunicação. Esse mecanismo é conhecido como:

a) shell;

b) pipes;

c) PID;

d) i-node;

e) cluster

13. (ESAF/CVM/2010) São tipos de sistemas operacionais:

a) **Sistemas Monousuários/Monopointer, Sistemas Multiusuários/Multipointer, Sistemas com múltiplas entradas.**

b) **Sistemas Monoprogramáveis/Monotarefa, Sistemas Multiprogramáveis/Multitarefa, Sistemas com múltiplos processadores.**

c) **Sistemas Monostakeholder/Monoinstrução, Sistemas Multistakeholder/Multi-instrução, Sistemas com múltiplos processadores.**

d) **Sistemas Monocompiláveis/Monomonitoramento, Sistemas Multicompiláveis/Multimonitoramento, Sistemas com múltiplos usuários.**

e) **Sistemas Monoplanejáveis/Monodesign, Sistemas Multiplanejáveis/Multidesign, Sistemas com processadores de segmentação.**

15. Gabarito.

- 01. C
- 02. D
- 03. A
- 04. D
- 05. C
- 06. E

- 07. C
- 08. B
- 09. E
- 10. A
- 11. B
- 12. B
- 13. B