João Victor Barbosa de Andrade – 20151y6-rc0223

1) As system calls são funções usadas pelas aplicações para requisitar a execução de algum serviço do kernel do SO, porisso elas tem nível de privilégio maior que as instruções normais dos aplicativos. Com essas chamadas de sistema, pode acessar serviços como: Manipular arquivos do disco, alocar memória, etc, além de que são elas que permitem a criação e o encerramento dos processos.

2) Ele gerencia a criação e eliminação dos processsos, a comunicação entre processos, também gerencia a escala dos processos para que a UCP execute. Ele também faz o controle de qual processo vai executar na UCP naquele momento, para dar a impressão de que eles tão em paralelo.

3) Ele faz a alocação e a desalocação de memória: Quando a aplicação pede memória ao sistema o sistema aloca, quando a aplicação deixa de utilizar a memoria alocada, o SO libera.

4) Ele faz a manipulação de arquivos e diretórios, sendo Abrir ou Fechar arquivos/diretórios, também criar esses arquivos e salvá-los no disco, Localizar arquivos/diretórios, etc.

5) O interpretador de comandos é o responável por interpretar os comandos entrados para outras aplicações ou também diretamente em chamadas de sistema. O shell fica fora do kernel pois o kernel é a abstração do hardware para que os processos utilizem os recursos de hardware mais facilmente, o shell é a parte que o usuário tem contato, para que sejam feitas as requisições para o kernel entrar em contato com o hardware.

6) A arquitetura de camadas, favorece a modificação do código de uma camada, também oferece mais robustez ao SO e também mais segurança, pois a arquitetura de camadas diz que : Cada camada usa os serviços da camada precedente a ela. Já as desvantagens, é que o processador precisa de uma arquitetura mais sofisticada para rodar esse SO e também precisam de algumas funções mais especiais para acessar camadas superiores.

7) Memória compartilhada – Quando uma mesma parte da memória se encontra no espaço de endereçamento de dois ou mais processos diferentes. Se um processo faz uma modificação nessa região, ela é vista por todos os outros processos que compartilhar essa parte da memória.
Vantagens: Eficiencia – É a maneira mais rápida para que dois processos efetuem uma troca de dados, ou seja, os dados não precisam passar para o kernel para que ele repasse os dados para os outros processos.
Desvantagens: Não Existe mecanismo automatico de sincronização. As vezes exige o uso de mecanismos (Locks, etc...) por parte dos processos, para que ocorra a sincronização.

Troca de Mensagens (PIPE) – Os pipes são um canal de comunicação, de uma única direção, entre processos. Quando um pipe é estabelecido entre processos, um deles pode enviar “mensagens” para o outro.
Vantagem : Comunicação simples entre processos – O envio e recebimento de “mensagens” é feito com os serviços normais de leitura e escrita de arquivos : o read() e o write()
Desvantagem : O pipe é unidirecional, ou seja, para que a troca de mensagens seja mútua, precisa ser estabelecido dois pipes entre os processos.

8) O Sistema operacional host abriga e executa o SO convidado por meio de virtualização, por exemplo, vmware. A virtualização emula em um SO host recursos de hardware para um SO convidado, e esse atua como um SO independente, ou seja, qualquer ação feita nele não afeta o SO host. Para que um SO convidado seja abrigado, o SO host precisa suportar virtualização de SO.

9) O microkernel, semelhantemente ao modelo em camadas, é modularizado, ou seja, dividido em partes que executam tarefas distintas, e que se comunicam. Porém, o microkernel, pode se comunicar entre os módulos sem a restrição de ser só com o anterior a ela.

10) Pois uma das funções do compilador JIT (Just-in-time) é a compilação em tempo de execução, ou seja, o código é compilado em enquanto é executado, assim permitindo ser inserido mais trechos de código enquanto o código já estava em execução.

11) O microkernel modulariza os gerentes do hardware, permitindo uma troca de gerentes sem precisar reiniciar a maquina, assim a expansão de sistemas distribuídos ocorre mais fácil. Os recursos são acessados por meio de um protocolo cliente/servidor, sendo o cliente os serviços do SO e o servidor os gerentes do microkernel. O desempenho dele é menor que o do kernel monolítico, pois essa modularização faz com eles tenham que se comunicar, e isso pode gerar uma perda.

12) O kernel monolítico é uma estrutura de kernel que roda inteiramente em modo protegido. Ela tem um bom desempenho em questão de comunicação de troca de mensagens, porem como é inteiramente em modo protegido, apresenta muitas desvantagens, por exemplo: alterações nele demandam um esforço grande e também desperdício de recursos, pois os drivers ficam sempre carregados, mesmo que os dispositivos não estejam em uso. Ele é a estrutura de kernel mais usada, pelo seu desempenho e simples estrutura, mesmo que outros SO atualmente já implementem o microkernel.

REFERENCIAS:

http://tecnologia.hsw.uol.com.br/sistemas-operacionais5.htm

http://www.uniriotec.br/~morganna/guia/chamadas.html

http://siep.ifpe.edu.br/anderson/arquivos/so/servicos_so.pdf

http://siep.ifpe.edu.br/anderson/arquivos/so/so1_aula3_organizacao_so.pdf

https://www.oficinadanet.com.br/artigo/1253/interpretador_de_comandos

http://pcleon.if.ufrgs.br/~leon/Livro_3_ed/node32.html

https://www.vivaolinux.com.br/artigo/Sistemas-Operacionais-Kernel-e-Shell?pagina=3

http://www.inf.ufes.br/~pdcosta/ensino/2008-1-sistemas-operacionais/Slides/Aula16-4slides.pdf

http://jeiks.net/wp-content/uploads/2010/09/Apostila_Pipes.pdf

https://pt.wikipedia.org/wiki/JIT

https://www.oficinadanet.com.br/post/10330-kernel-dos-sistemas-operacionais