



Sistemas Operacionais

CCP/SIF

UNISUL – Tubarão

Cassio Brodbeck Caporal

`cassio{NOSPAM}ostec.com.br`

Agenda

- Revisão sobre tipos de sistemas operacionais;
- Revisão sobre hardware e software básico;
 - Processador, memória (principal, cache, secundária), dispositivos de E/S, BUS, arquiteturas CISC e RISC;
 - Softwares básicos: linkers, compiladores, carregadores (loaders), depuradores, etc.

Agenda

- Estrutura de Sistemas Operacionais
 - Chamadas de sistema;
 - Kernel monolítico;
 - Kernel em camadas;
 - Microkernel;
 - Virtual machine.

Central Processor Unit (CPU)

- Função principal:
 - Controle e execução de instruções localizadas na memória principal;
 - Utilização de registradores para armazenar variáveis e resultados (temporário);
- Componentes: Unidade de Controle (UC), Unidade Lógica e Aritmética (ULA), Registradores.

Central Processor Unit (CPU)

- Unidade de Controle:
 - Responsável por gerenciar todos os recursos (componentes) do computador;
- Unidade Lógica e Aritmética:
 - Responsável por operações lógicas e aritméticas (sério???)

Central Processor Unit (CPU)

- Registradores:
 - Armazenamento de dados temporários;
 - Memória de alta velocidade e pouco armazenamento;
 - Completamente dependente de arquitetura computacional;

Central Processor Unit (CPU)

- Registradores:
 - Program counter (contador de instruções):
 - Armazenamento do endereço da próxima instrução a ser executada.
 - Stack pointer;
 - Aponta para o topo da pilha.
 - Registrador de status (PSW):
 - Armazena informações sobre execução de instruções

Memória Principal

- Local de armazenamento de dados e instruções;
- Células;
 - Endereços de memória referenciam células;
 - Ponteiros em C;
- *Memory Register Address* (MRA ou MAR);
 - Identificação de qual célula será utilizada.

Memória Principal

- *Memory Buffer Register* (MBR);
 - Não confuda com *Master Boot Record*;
 - Registrador responsável por guardar a informação propriamente dita;
- Classificação:
 - Volátil;
 - Não-volátil;

Memória Cache

- Características:
 - Alta velocidade;
 - Baixa capacidade de armazenamento;
 - Caras (R\$);
- Finalidade:
 - Aumentar a velocidade da área de troca entre CPU e memória principal;

Memória Cache

- Mensagens:
 - CACHE_HIT;
 - CACHE_MISS;
- E os registradores?
 - Também são considerados um tipo de memória específica da CPU;
 - Altíssima velocidade.

Memória Secundária

- Memória não-volátil;
- Características:
 - Alta capacidade de armazenamento;
 - Baixa velocidade (??);
 - Baixo custo (R\$);
- Exemplo: *hard drives*.

Dispositivos de E/S

- Propriedades:
 - Controlador;
 - Dispositivo;
- Controlador é um chip que controla fisicamente um dispositivo (interface para o SO);
- Driver: comunicação com controlador.

Dispositivos de E/S

- Interrupções
 - Exemplo com chegada de um pacote de rede;
- *Direct Memory Access*;
 - Controle de fluxo de bits diretamente entre o dispositivo e a memória, sem intervenção da CPU.

Barramentos

- Meio físico de comunicação entre unidades funcionais de um computador;
- Existência de linhas de controle e linhas para transmissão/transferência de dados;
- Com a evolução, surgiu a necessidade de criar diversos outros barramentos: cache, local, memória, PCI, SCSI, USB, IDE, SATA, ISA, ...);

Barramentos

- *Industry Standard Architecture;*
 - IBM PC/AT, 8,33Mhz, transfere 2 bytes de uma vez, vel. Máxima de 16,67MB/s;
- *Peripheral Component Interconnect;*
 - Criado pela Intel como sucessor do ISA;
 - Funciona a 66Mhz e transfere 8 bytes por vez, 528MB/s;
 - PCI-X 64/133/266/533.

Pipelining

- Execução de múltiplas instruções paralelamente (mas em estágios/momentos diferentes);
- Fracionamento de instruções (sub-tarefas);
- Unidades funcionais *pipelining*;
- Característica:
 - Aumento de desempenho;

Arquitetura RISC

- *Reduced Instruction Set Computer;*
- Caracterizada por poucas instruções, executadas de forma simples pelo próprio hardware;
- Pouca iteratividade com a memória principal devido a existência de múltiplos registradores;

Arquitetura CISC

- *Complex Instruction Set Computer;*
- Instruções complexas executadas por microprogramas;
- Utilização contínua de memória principal, poucos registradores.

Estrutura de Sistemas Operacionais

- Kernel (o coração):
 - Conjunto de “rotinas” que oferecem tanto serviços para usuários e aplicações, como ao próprio sistema;
- Utilitários *versus* aplicações;
- Hierarquia: aplicações → utilitários → kernel → hardware;

Estrutura de Sistemas Operacionais

- Mas o que o *kernel* efetivamente faz?
 - Gerenciamento de memória, E/S, *file systems*, escalonamento de processos, tratamento de exceções, concorrência, interrupções, etc etc etc.
 - Para isso uma disciplina de Sistemas Operacionais. :-)

Chamadas de Sistema

- Antes de tudo, o que é *kernel* e *user space*?
 - Ou então, instruções privilegiadas e não-privilegiadas?
 - Certas instruções não podem ser feitas por um usuário, é necessário uma camada intermediária de proteção;
 - Aplicações precisam ler informações do disco, mas não podem fazer isso DIRETAMENTE;

Chamadas de Sistema

- *System calls* são implementadas como forma de interfaceamento entre *user space* e *kernel space*;
- Qual objetivo?
 - Não comprometer a integridade do sistema;
- Através das *system calls* podemos ler um arquivo sem comprometer o sistema...

Chamadas de Sistema

- .. não precisamos saber informações específicas do dispositivo (abstração);
- Através das *system calls* a execução de um processo entre em modo privilegiado de forma segura;
- *System call* → kernel space → execução → user space → retorno;

Chamadas de Sistema

- Cada sistema operacional tem seu conjunto de chamadas de sistema;
 - POSIX 9945-1 (Portable Operating System Interface for Unix):
 - Padronização de chamadas de sistema;

Chamadas de Sistema

- Exemplo prático:
 - Um simples “Hello world” / Uma simples substituição ao “cat”.
 - *Análise com system call trace.*

Kernel monolítico

- Presença de vários módulos;
- Compilação separada, *linking* para gerar um único e grande arquivo;
 - Exemplo: *vmlinuz*.
- Desenvolvimento facilitado, no entanto, manutenção crítica;
- MS-DOS, primeiros Unices.

Kernel em camadas

- Divisão em níveis onde cada um sabe exatamente o que oferecer para camada superior;
- Características:
 - Isolamento de funções do sistema operacional;
 - Baixo desempenho através de mudanças de acesso entre camadas (modos);

Microkernel

- Paradigma de serviços como processos;
- Relacionamento de cliente e servidor para processar requisições;
- Características:
 - Fácil manutenção;
 - Flexibilidade;
 - Alta portabilidade.

Microkernel

- Na prática, hmmm, “nem tudo são rosas”..
 - Implementação MUUUUITO complicada;
 - Funções com acesso direto ao hardware, especialmente dispositivos de E/S;
 - Exemplo: Exokernel (MIT), GNU/Hurd.

Máquina virtual

- Substituição de uma camada da arquitetura computacional;
- Pode estar entre sistema operacional e hardware;
- Pode estar entre aplicação e sistema operacional (JVM);
- Problemas? Desempenho :P