

Sistemas Operacionais

Roberto Willrich
INE- CTC-UFSC
E-Mail: willrich@inf.ufsc.br
URL: <http://www.inf.ufsc.br/~willrich>

Introdução


- Porquê existem os sistemas operacionais?
 - Um computador é composto de:
 - um (ou mais) processadores
 - memória
 - clocks
 - terminais
 - discos
 - interfaces de rede
 - outros dispositivos de E/S.
 - Construir programas que utilizem todos estes componentes é uma tarefa extremamente difícil.
 - Algo precisava ser feito!

Introdução

- No computador
 - Diversas atividades competem os recursos da máquina
 - uma máquina pode ser conectada a diversos terminais atendendo a diversos usuários
 - um PC pode realizar diversas atividades ao mesmo tempo:
 - impressão, edição, anti-virus, ...
 - Competição exige um alto grau de coordenação
 - para evitar que atividades independentes umas as outras
 - garantindo a comunicação eficiente e confiável entre atividades interdependentes

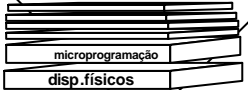
Introdução

- Solução: isolar os programadores dos detalhes de hardware através de uma camada de software
 - máquina virtual



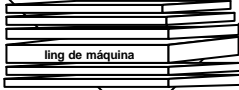
Introdução

- Microprograma
 - nível de software primitivo que controla os dispositivos
 - é um interpretador de instruções tais como ADD,MOV
 - quebra tais instruções em vários passos menores
 - o conjunto de instruções que o microprograma interpreta define a linguagem de máquina
 - não faz parte do hardware, mas usuários pensam que é



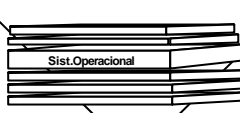
Introdução

- Linguagem de Máquina - Assembly
 - uma linguagem de máquina possui entre 50 e 300 instruções
 - movimentação de dados, comparações, operações aritméticas
 - E/S é controlado através de dispositivos chamados registradores



Introdução

- Sistema Operacional
 - finalidade: esconder a complexidade e proporcionar uma interface mais amigável com o hardware
 - em vez de comandar o cabeçote pode-se ler diretamente um bloco de dados

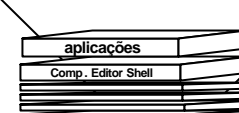


Roberto Willrich - INEAFSC - 09/10/2000

7

Introdução

- Demais Camadas
 - acima do S.O. estão os demais softwares:
 - interpretador de comandos (shell) , compiladores, ligadores, etc;
 - programas aplicativos



Roberto Willrich - INEAFSC - 09/10/2000

8

O que é um sistema operacional?

- Visão top-down:
 - uma camada de software que apresenta para o usuário uma visão modificada do hardware
- Visão bottom-up:
 - um gerenciador de recursos!
 - sob esta ótica deve prover acesso controlado a:
 - processadores, memória, dispositivos de E/S, etc
- O Sistema Operacional tem as seguintes características:
 - Definição de interface com o usuário;
 - Compartilhamento do hardware entre os usuários;
 - Permite aos usuários compartilhar dados entre si;
 - Escalonamento de recursos entre usuários (estabelecer a sequência de quem ira usar);
 - Facilidades de Entrada e Saída;
 - Recuperação de erros.

Roberto Willrich - INEAFSC - 09/10/2000

9

O Processo Evolutivo

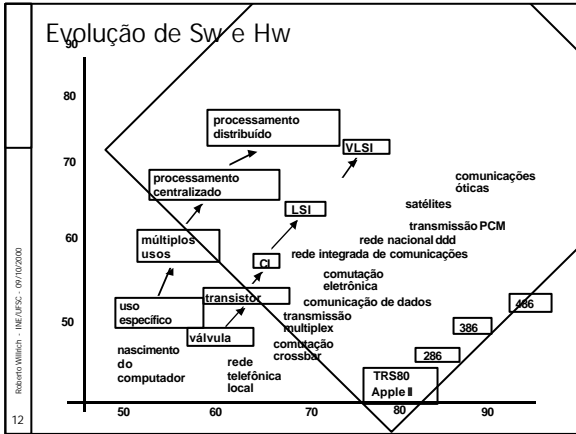
Os sistemas operacionais tem sofrido uma série de mudanças revolucionárias que podem ser caracterizadas como gerações.

Gerações do computador

- Evolução do hardware
 - primeira geração: válvulas
 - segunda geração: transistores
 - terceira geração: circuitos integrados
 - quarta geração: circuitos LSI e VLSI (very large-scale integrated circuitry)
 - as gerações seguintes tem se caracterizado por:
 - redução de custo, tamanho, emissão de calor e consumo de energia
 - aumento na velocidade e capacidade de armazenamento

Roberto Willrich - INEAFSC - 09/10/2000

11



Evolução da arquitetura de hw: classe 1

- Processador central
 - controla diretamente I/O emitindo seqüências de comandos dentro de períodos de tempo bastante rígidos (timing)

Mem — CPU — Periférico

13

Evolução da arquitetura de hw: classe 2

- Inserido um controlador de I/O
 - tarefa de controlar seqüência de pulsos para o periférico é retirada do processador
 - processador periodicamente examina (poll) o controlador p/ saber quando ele terminou um comando.
 - processador central e o controlador operam em paralelo.

Mem — CPU — controlador — Periférico

14

Evolução da arquitetura de hw: classe 3

- Controlador passa a sinalizar o processador quando do término de um comando através de interrupções.
 - processador não gasta tempo realizando polling

Mem — CPU — controlador — Periférico

15

Evolução da arquitetura de hw: classe 4

- Acesso direto a memória (DMA) é acrescentado ao controlador
 - blocos de dados são movimentados de/para memória sem intervenção do processador.
 - processador é interrompido somente após o final da transferência

Mem — CPU — controlador — Periférico

16

Evolução da arquitetura de hw: classe 5

- Um buffer de instruções é acrescentado ao controlador para permitir ao processador enviar uma seqüência de operações de I/O
 - controlador interrompe somente após seqüência ter sido executada

Mem — CPU — controlador — buffer — Periférico

17

Evolução da arquitetura de hw: classe 6

- O controlador é aperfeiçoado p/conter um conjunto completo de instruções
 - controle, looping e teste.
 - processador central cria um programa de I/O na memória que o controlador pode buscar e executar.
 - ex: processadores de canais dos IBM /360 /370

Mem — CPU — Proc I/O — Periférico

18

Evolução da arquitetura de hw: classe 7

- O processador de I/O tem uma memória local tornando-se um computador e formando uma rede com o processador central
 - ex. CDC6600

19

Evolução da arquitetura de hw: classe 8

- O processador de I/O tem um conjunto de instruções geral e pode evoluir da mesma forma que o processador principal.

20

O ciclo evolutivo do hardware

21

Gerações de Sistemas Operacionais

Os sistemas operacionais do mesmo modo que o hardware passou por diversos estágios que podemos classificar como gerações ...

Primeira Geração: 1945 a 1955

- Características:
 - Válvulas
 - não permitiam confiabilidade do hardware máquinas enormes e muito lentas
 - usados p/solução de cálculos numéricos simples (seno,...)
 - usuários tinham acesso completo ao hardware
 - não havia linguagem de programação
 - programação era feita pela modificação dos circuitos
 - Utilização
 - por um grupo de pessoas que projetava, construía, operava e mantinha o computador.
 - programadores reservavam um período para executar seus programas
 - não existiam sistemas operacionais
 - início da década de 50 surge o cartão perfurado

23

Segunda Geração 1955-65

- Característica: transistores
 - (meados de 50) tornam projetos mais confiáveis
 - atendiam a grandes corporações, agências governamentais e universidades
 - custo muito alto
 - torna-se clara a separação entre projetistas, construtores, operadores, programadores e pessoal de manutenção

24

Segunda Geração 1955-65

- Para executar um programa (job):
 - Programador primeiro escrevia em papel (Fortran ou assembly)
 - Programa é transformado em cartões perfurados
 - Programador levava os cartões para um operador
 - que colocava em uma unidade de leitura
 - Após o job
 - operador levava a saída ao programador

Roberto Willrich - INEAFSC - 09/10/2000

25

Processamento em lote

- muito tempo era gasto enquanto os operadores caminhavam de um lado para outro
 - pegando novos jobs
 - introduzindo-os no computador
 - iniciando o processo
 - recolhendo os resultados
 - entregando-os para o programador
- tempo é dinheiro
 - solução
 - sistemas batch (processamento em lote)

Roberto Willrich - INEAFSC - 09/10/2000

26

Processamento em Lote

- sistemas batch
 - coletava-se um conjunto de jobs da sala de entrada
 - transferia-os para uma fita usando um computador mais barato
 - a fita era rebobinada e levada para a sala do computador maior
 - o operador carregava um programa especial (ancestral do S.O.) que lia o primeiro job e executava-o
 - a saída era gravada numa segunda fita ao invés de ser impressa
 - após o término de cada job, o sistema automaticamente lia o próximo job da fita e iniciava sua execução
 - depois de processada a fita de entrada, o operador retirava ambas as fitas e levava a fita de saída para outro computador para impressão off-line

Roberto Willrich - INEAFSC - 09/10/2000

27

Processamento em Lote

- sistemas batch

Roberto Willrich - INEAFSC - 09/10/2000

28

Processamento em Lote

- Sistema Operacional
 - simplifica a instalação de programas e torna suave a transição de um job para outro
 - não há interação com o usuário
 - jobs eram armazenadas em fita aguardavam para serem executados em uma fila de jobs
 - fila FIFO (First In, First-out)
 - maioria tinham esquemas de prioridade

Roberto Willrich - INEAFSC - 09/10/2000

29

Primeiros S.O.s

- Cada job executava até terminar ou parava em uma situação de erro
 - o operador tinha que carregar um programa para executar dump de memória, remover fitas, cartões perfurados e relatórios
 - somente depois o próximo job era carregado para execução
 - enquanto isso o sistema ficava parado
- Características do IBM 7094
 - quando um job parava para aguardar uma fita ou qualquer operação de E/S, a CPU permanecia *idle* até a operação terminar
 - em aplicações científicas esta característica não comprometia muito o resultado final (CPU-bound)
 - em aplicações comerciais, onde o tempo de E/S atinge 80% a 90% do tempo total, o resultado final era sofrível

Roberto Willrich - INEAFSC - 09/10/2000

30

Final da década de 50

- características dos S.O's fornecidos na época
 - processamento em batch enfileirado
 - rotinas padrão de entrada/saída p/ isolar detalhes de mais baixo nível
 - facilidades de transição entre jobs para reduzir o overhead envolvido na troca de jobs
 - técnicas de recuperação de erros que automaticamente "faziam a limpeza" após um job terminar anormalmente e permitiam o próximo job inicial com um mínimo de intervenção do operador
 - jcl (job control language) permitiam aos usuários especificar em detalhes seus jobs e recursos necessários
 - impresso antes da execução do job

31

Terceira Geração: 1965-1980

- no início dos anos 60 a maioria dos fabricantes tinham 2 linhas de produtos diferentes e totalmente incompatíveis:
 - **linha científica**
 - computadores científicos de larga escala
 - usados para cálculos numéricos em ciência e engenharia
 - ex: 7094 IBM
 - **linha comercial**
 - computadores comerciais
 - usados para classificar e imprimir relatórios
 - ex 1401

32

O conceito de família

- havia o problema:
 - muitos clientes necessitavam inicialmente de computadores menores
 - mais tarde necessitavam de computadores maiores que:
 - fossem mais rápidos
 - executassem os programas da instalação sem alteração
- IBM propõe a família System/360
 - série de máquinas que mantinham compatibilidade de software
 - máquinas diferiam somente em preço e performance
 - capacidade de memória
 - velocidade do processador
 - número de dispositivos de E/S permitidos
 - como as máquinas tinham a mesma arquitetura e conjunto de instruções:
 - ao menos em teoria, programas escritos p/ uma máquina poderiam rodar em todas as demais

33

Características Principais

- caracterizaram-se por serem sistemas multimodo
 - processamento batch
 - processamento time-sharing
 - processamento real-time
 - multiprocessamento
- eram sistemas operacionais grandes e caros
- sistemas introduziram grande complexidade com a qual os usuários não estavam acostumados
 - sistemas interpuseram uma camada de software entre o usuário e o hardware
 - usuários tinham que acostumar-se aos JCL (job control languages)

34

Conceito de Família: pontos fracos

- o grande apelo da ideia de uma família de compatíveis foi também o ponto fraco.
 - a ideia era que todo o software, incluindo o S.O. deveria rodar em todos os modelos
 - deveria rodar em máquinas pequenas que substituíam os 1401 (cópia de fitas, cartões, impressão) e ao mesmo tempo em máquinas grandes que substituíam os 7094 (processamento pesado)
 - deveria rodar com poucos periféricos e com muitos
 - deveria atender a ambientes comerciais e científicos
 - deveria ser eficiente em todos os ambientes
- SO deveria atender a todos estes requisitos contraditórios
 - um S.O. extremamente complexo e grande
 - milhões de linha de linguagem assembly escrita por vários programadores e com milhões de bugs...

35

Requisitos conflitantes

- como haviam requisitos conflitantes
 - o resultado foi um enorme e complexo sistema operacional
 - milhares de linhas em assembly
 - escrito por milhares de programadores
 - contendo centenas de "bugs" que necessitavam um contínuo fluxo de novas releases
 - cada release corrigia alguns problemas e introduzia novos
 - provavelmente o número de bugs permanecia constante no tempo
- independentemente dos problemas, esta geração satisfazia a maioria dos usuários razoavelmente bem

36

Requisitos Conflitantes: a solução da IBM

- pequenos usuários não necessitavam todo o poderio do OS/360
- instalações pequenas sofriam severa degradação de performance
- Solução da IBM
 - DOS/360 para pequenos usuários 360
 - OS/MFT para médios e grandes usuários 360
 - OS/MVT para grandes usuários 360
 - CP-67/CMS para o poderoso 360/67 que era timesharing e possuía memória virtual

37

Multiprogramação

- Solução para o problema de tempo de resposta enquanto jobs esperavam por E/S
 - particionar a memória
 - instalar um job em cada partição
 - enquanto um job esperava por uma operação de E/S, outro job poderia usar a CPU
 - se fosse possível manter na memória todos os jobs de uma só vez, a CPU poderia ter uma taxa de utilização de 100%
 - múltiplos jobs necessitam de proteção de memória
 - 3ª geração implementava em hardware

38

Spool

- característica importante
 - sempre que um job terminava, o S.O. podia carregar o próximo do disco para a partição disponível e executá-lo
 - técnica denominada SPOOLing (Simultaneous Peripheral Operation On Line)
 - SPOOLing também foi utilizado para saídas
 - com esta técnica os 1401 não eram mais necessários e o transporte de fitas de um lado para outro diminuiu sensivelmente

39

Time-sharing (compartilhamento do tempo)

- apesar de bem aceitos para aplicações científicas e comerciais, constituíam-se em sistemas batch
 - tempo de resposta comprometido
 - ex. procedimento de compilação
 - procedimento de submeter um programa para compilação, receber o resultado, corrigir algum erro, resubmetê-lo novamente poderia consumir várias horas
- necessidade de melhores tempos de resposta conduziram a sistemas time-sharing
 - variação de multiprogramação onde cada usuário tem um terminal on-line

40

Processamento em batch e interativo

- Processamento em batch
 - tem falta de interatividade com os usuários

- Processo interativo, timing sharing

41

Compartilhamento do tempo

- Sistema operacional
 - deve ser capaz de revezar a execução dos vários jobs através do compartilhamento do tempo (time-sharing)
 - refere à técnica de dividir o tempo em intervalos ou fatias (time-slices), e restringir a execução, dentro de cada uma dessas frações de tempo, a um job de cada vez
 - no término do intervalo, o job corrente é retirado do processamento e um outro é acionado
 - revezamento de jobs cria a impressão de paralelismo
 - requer troca de contexto (salvamento do estado do programa na troca)
 - antigos sistemas operacionais suportavam até cerca de 30 usuários com uma aceitável resposta em tempo-real.
- Atualmente
 - time-sharing é usada também em sistemas mono-usuários (multitasking)
 - mais de uma tarefa pode ser executada ao mesmo tempo
 - aumenta o desempenho global do sistema

42

43

Roberto Willrich - INEAFSC - 09/10/2000

Terminais

- foram desenvolvidos sistemas onde o usuário podia interagir diretamente com o computador através de terminais typewriterlike
 - sistemas time-sharing operavam em modo interativo
 - usuário podia enviar requisições ao computador teclando comandos
 - computador processava os comandos
 - respostas eram escritas no terminal do usuário
 - este tipo de interação revolucionou o processo de desenvolvimento de programas
 - usuário podia rapidamente localizar e corrigir erros em segundos ou minutos

44

Roberto Willrich - INEAFSC - 09/10/2000

CTSS – *Compatible Time-Sharing System*

- time-sharing
 - se o sistema tem 20 usuários e 17 estão conversando, ou pensando, a CPU pode ser alocada para os 3 jobs restantes
 - permite melhores tempos de resposta para usuários interativos
- CTSS
 - primeiro sistema time-sharing
 - desenvolvido no MIT em uma versão modificada do 7094 (segunda geração)
 - não tornou-se muito popular até o surgimento de mecanismos de proteção por hardware (terceira geração)
 - rodou de 61 até final da década de 60
 - suportava grande número de usuários interativos

45

Roberto Willrich - INEAFSC - 09/10/2000

Computador utilitário

- após o sucesso do CTSS
 - MIT, Bell Labs e General Electric (maiores fabricantes da época) decidiram iniciar o desenvolvimento de um computador utilitário
 - máquina que poderia suportar centenas de usuários time-sharing simultâneos
 - ideia do sistema de distribuição de eletricidade: "quando você necessita basta plugar na tomada e a energia necessária estará lá"
 - MULTICS - MULTIpLexed Information and Computing Service)
 - S.O de grande escala
 - apesar das idéias inovadoras, construí-lo foi mais difícil que se podia imaginar
 - Bell cancelou o projeto
 - General Electric saiu do ramo de computadores
 - Multics influenciou em muito os sistemas subsequentes

46

Roberto Willrich - INEAFSC - 09/10/2000

Thompson & Kernighan

- Ken Thompson
 - um dos projetistas do Multics
 - achou um PDP-7 que não estava sendo usado e iniciou o desenvolvimento de uma versão "enxuta" e monousuário do Multics
- Brian Kernighan
 - brincando com K.Thompson apelidou o novo sistema de UNICS (UNIpLexed Information and Computing Service)
 - a pronúncia do UNICS originou o nome de batismo atual UNIX
 - Mais tarde o sistema foi portado para um PDP11/20 e posteriormente para um PDP11/45

47

Roberto Willrich - INEAFSC - 09/10/2000

UNIX

- Dennis Ritchie
 - outro cientista da Bell Labs
 - projetou e implementou a linguagem C
 - juntamente com K.Thompson reescreveu o sistema em C
- Bell Labs
 - licenciou o sistema para universidades praticamente sem custo
 - em pouco tempo centenas delas estavam utilizando o novo sistema
- O sistema UNIX foi portado para Interdata 7/32, VAX, Motorola 68000 e vários outros computadores

48

Roberto Willrich - INEAFSC - 09/10/2000

Quarta Geração (1980-hoje)

- Característica : PC's e estações de trabalho
 - desenvolvimento de chips em LSI (Large Scale Integration) permitiu o desenvolvimento de computadores menores
 - conectados via redes de computadores
- Inicialmente 2 sistemas operacionais eram dominantes
 - MS-DOS - Microsoft
 - arquitetura Intel (IBM PC)
 - sucessor é o Windows'9x
 - windows 3.x eram apenas um shell do sistema MSDOS
 - UNIX
 - arquitetura motorola 68000, arquitetura Intel e RISC's
 - variantes: LINUX, Solaris, AIX

NOS & DOS

- a partir de meados de 1980, iniciou o crescimento dos chamados
 - **sistemas operacionais de rede**
 - usuários tem conhecimento dos demais computadores
 - podem se logar em máquinas remotas e trocar arquivos
 - cada máquina roda seu próprio sistema operacional local e tem seu(s) próprio(s) usuário(s)
 - S.O. tradicional com uma interface de rede e softwares para permitir conexão e acesso a arquivos remotos
 - mudam a estrutura do S.O.
 - **sistemas operacionais distribuídos**
 - o sistema comporta-se como se fosse mono-processador
 - os usuários não sabem onde seus programas estão sendo executados nem onde seus arquivos estão sendo armazenados
 - distribuição é transparente para o usuário

Roberto Willrich - INEAFSC - 09/10/2000
49

Sistemas Multiprocessados

- Computadores com vários processadores
 - sistema operacional não apenas coordenará a competição entre as várias tarefas que são executadas de fato simultaneamente também controlará a alocação de tarefas aos diversos processados
 - envolve problemas de balanceamento de carga
 - garantir que os processadores sejam utilizados de modo eficiente
 - envolve também escalção
 - divisão das tarefas em várias sub-tarefas de número compatível com o número de processadores disponíveis

Roberto Willrich - INEAFSC - 09/10/2000
50

Arquitetura dos Sistemas Operacionais

Arquitetura de um Sistema Operacional

- Basicamente dividido em
 - **shell**
 - é a interface entre o usuário e o sistema operacional
 - é um interpretador de comandos
 - possui embutido uma linguagem de programação
 - **núcleo (kernel)**
 - funções básicas necessárias à utilização do computador

Roberto Willrich - INEAFSC - 09/10/2000
52

Shell

- Define uma interface entre o sistema operacional e seus usuários
 - fornece uma comunicação natural com os usuários do computador
 - shells antigos se comunicavam com os usuários, via teclado e monitor, por mensagens textuais
 - shells modernos fornecem uma interface gráfica com o usuário
 - GUI - Graphical User Interface
 - na qual objetos a serem manipulados (arquivos e programas) são representados graficamente através de ícones na tela
 - permite ao usuário executar um comando do sistema apontando e deslocando ícones

Roberto Willrich - INEAFSC - 09/10/2000
53

Shell

Roberto Willrich - INEAFSC - 09/10/2000
54

Núcleo (kernel)

- Gerenciador de Arquivo (Sistema de Arquivos)
 - A nível de shell o Sistema de Arquivos oferece chamadas de sistema, com interface única para dispositivos
 - Operações mais comuns:
 - Criar; excluir (create; delete)
 - Ler; gravar (read; write)
 - Abrir; fechar (open; close)
 - Alterar nome (rename)
 - Ler e escrever atributos (get/set attributes)

Roberto Willrich - INEAFSC - 09/10/2000

61

Núcleo (kernel)

- Acionadores de dispositivos (device drivers)
 - são os módulos de software que executam a comunicação com os controladores
 - ou as vezes diretamente com os dispositivos
 - solicitando a realização de tarefas aos periféricos
 - um driver é projetado para um dado tipo de controlador ou dispositivo
 - impressora, unidade de disco, unidade de fita, monitor, ...
 - drivers convertem solicitações de alto nível em comandos mais elementares
 - diretamente reconhecíveis pelos controladores ou dispositivos associados àquele driver
 - detalhes técnicos associados aos dispositivos são confinados aos drivers
 - ficando transparentes aos demais módulos de software

Roberto Willrich - INEAFSC - 09/10/2000

62

Núcleo (kernel)

- Acionadores de dispositivos (device drivers)
 - Exemplo: gravação em disco
 - driver para uma unidade de disco é capaz de converter um pedido de gravação de um trecho de arquivo em disco para uma sequência de passos
 - referenciando trilhas e setores, e transferir toda essa informação para o controlador apropriado
 - baseado nas informações extraídas do descritor de arquivo
 - controlador se responsabiliza pelo posicionamento do cabeçote de leitura e gravação e pelo acompanhamento do processo de gravação
 - Exemplo: impressão
 - acionador para um impressora efetua a conversão de um pedido e impressão de um trecho de arquivo para um conjunto de operações básicas
 - envolvendo transferência de caracteres, tipos de letras e contornos de impressão
 - variam de impressora para impressora

Roberto Willrich - INEAFSC - 09/10/2000

63

Núcleo (kernel)


- Gerenciador de Memória
 - Parte do SO que gerencia a utilização da memória principal
 - Em computadores monoprocessados é trivial
 - programa a executar é colocado na memória principal
 - após a execução ele é substituído pela tarefa seguinte
 - Em ambiente multi-usuários ou multi-tarefas é complexo
 - máquina se encarrega de executar diversas tarefas ao mesmo tempo
 - muitos programas e blocos de dados coexistem na memória principal
 - cada qual em sua área de memória própria determinada pelo gerenciador de memória
 - na medida das necessidades das diferentes atividades o gerenciador vai providenciando as áreas necessárias
 - mantendo um mapa das regiões de memória não utilizadas

Roberto Willrich - INEAFSC - 09/10/2000

64

Núcleo (kernel)

- Gerenciador de Memória
 - Sua tarefa se torna mais complexa quando a área total de memória principal solicitada excede espaço realmente disponível na máquina
 - gerenciador pode criar a ilusão de espaço adicional alterando os programas e dados entre a memória principal e o disco
 - espaço ilusório é chamado de **memória virtual**



Roberto Willrich - INEAFSC - 09/10/2000

65

Núcleo (kernel)

- Gerenciador de Memória
 - supõe que seja solicitada uma área de 64MB e somente 32MB estejam disponíveis
 - para criar a ilusão de um espaço maior de memória, o gerenciador de memória divide a área solicitada em partes chamadas **páginas** e armazena em disco o conteúdo destas páginas
 - um tamanho típico de página não é superior a 4KB
 - na medida que estas páginas forem sendo solicitadas
 - gerenciador de memória pode armazená-las na memória física em substituição a outras que já não sejam mais necessárias
 - » arquivo de swap
 - de modo que as demais partes do software possam ser executadas

Roberto Willrich - INEAFSC - 09/10/2000

66

Núcleo (kernel)

- Escalonador (scheduler) e despachante (dispatcher)
 - Necessário em um sistema de tempo compartilhado
- Escalonador
 - determina quais atividades serão executadas
- Despachante
 - controla a distribuição de fatias de tempo para tais atividades

Roberto Willrich - INEAFSC - 09/10/2000
67

Coordenação das atividades da máquina

- Objetivo
 - Introduzir o conceito de processo
 - Mostrar como o SO coordena a execução dos softwares de aplicação, utilitários e do próprio SO
- Processo
 - Um dos conceitos mais fundamentais dos SOs modernos
 - é uma distinção entre um programa e a atividade de executá-lo
 - programa
 - um conjunto estático de instruções
 - execução do programa
 - uma atividade dinâmica
 - propriedades mudam à medida que o tempo avança
 - atividade conhecida como **processo**

Roberto Willrich - INEAFSC - 09/10/2000
68

Coordenação das atividades da máquina

- Processo
 - Leva em conta a situação corrente da atividade
 - conhecida com o **estado do processo**
 - Estado do processo, inclui
 - posição do programa que está sendo correntemente executada (contador de instruções)
 - valores contidos nos registradores do processador
 - posições associadas de memória

Roberto Willrich - INEAFSC - 09/10/2000
69

Coordenação das atividades da máquina

- Um único programa pode ser associado a mais de um processo
 - em sistemas multi-usuários de tempo compartilhado
 - dois usuários podem editar textos ao mesmo tempo usando um mesmo programa
 - cada qual caracteriza um processo separado
 - com seus próprios conjunto de dados e sua própria taxa de progresso
 - SO pode manter na memória principal uma só cópia do programa editor e permitir que cada processo o utilize à sua maneira (durante a fatia de tempo que lhe couber)

Roberto Willrich - INEAFSC - 09/10/2000
70

Coordenação das atividades da máquina

- É natural que processos compitam por fatias de tempo
 - estes processos pode ser associados a aplicativos, utilitários e partes do próprio SO
 - SO deve coordenar todos estes processos
 - atividade de coordenação inclui
 - garantir que cada processo tenha acesso aos recursos de que necessita
 - dispositivos, periféricos, área na memória principal, acesso a dados e acesso ao processador)
 - que processos independentes não interfiram uns com os outros
 - que processos que se intercomunicam tenham a possibilidade de trocar informações entre si (comunicação interprocessos)

Roberto Willrich - INEAFSC - 09/10/2000
71

Coordenação das atividades da máquina

- Administração de Processos
 - Escalonador (scheduler) e Despachante (dispatcher)
 - realizam as atividades de coordenação de processos
 - partes integrantes do núcleo do SO
- Escalonador
 - mantém um registro dos processos presentes no sistema computacional
 - inclui novos processos neste conjunto
 - remove processos que já completaram sua missão

Roberto Willrich - INEAFSC - 09/10/2000
72

Coordenação das atividades da máquina

- Escalonador mantém uma **tabela de processos**
 - região na memória principal utilizada para coordenar os processos
 - mantém vários indicadores para cada processo:
 - área de memória designada para o processo (obtida pelo gerenciador de memória)
 - prioridade do processo
 - indicador se o processo está pronto para ser executado ou à espera de um evento
 - pronto: estado em que o processo está pronto para prosseguir
 - estado de espera: aguardando um evento (acesso a disco ou recebimento de uma mensagem)

73

Coordenação das atividades da máquina

- Escalonador mantém uma **tabela de processos**
 - É mantida atualizada à medida que o processo vai progredindo
 - Na medida que os processos evoluem
 - irá alternar de estado pronto e estado de espera
 - terá sua prioridade variada ao longo da execução
 - irá ser removido da tabela no final da sua execução

74

Coordenação das atividades da máquina

- Despachante
 - Módulo do núcleo do SO cuja função é a de assegurar que os processos escalonados sejam de fato executados
 - Atividade realizada dividindo-se o tempo físico em pequenas fatias (cerca de 50ms)
 - denominada **quantum** (*time slice*)
 - Atenção do processador é revezada entre os processos
 - a cada qual é concedido um intervalo de tempo não superior a duração do quantum
 - procedimento de alternar o processador de um processo para outro é chamado de **chaveamento de processos**

75

Interrupção e Trap

- Interrupções e traps são as forças que movimentam e dirigem os sistemas operacionais
 - pois um sistema operacional só recebe o controle da execução quando ocorre alguma interrupção ou trap
- Uma interrupção
 - é um sinal de hardware que faz com que o processador sinalizado interrompa a execução do programa que vinha executando
 - guardando informações para poder continuar, mais tarde, a execução desse programa
 - e passe a executar uma rotina específica que trata da interrupção

76

Interrupção e Trap

- Um trap
 - é uma instrução especial que, quando executada pelo processador, origina as mesmas ações ocasionadas por uma interrupção
 - salvamento de informações para poder continuar, mais tarde, a execução do programa e desvio para uma rotina específica que trata do trap
 - um trap é uma interrupção ocasionada por software
 - é uma interrupção prevista
 - programada no sistema pelo próprio programador

77

Interrupção e Trap

- Origem de uma interrupção
 - podem ser originadas pelos vários dispositivos periféricos (terminais, discos, impressoras, etc.), pelo operador (através das teclas do console de operação) ou pelo relógio do sistema
- relógio (timer)
 - é um dispositivo de hardware que
 - decrementa automaticamente o conteúdo de um registrador ou posição de memória, com uma frequência constante
 - e interrompe a CPU quando o valor decrementado atinge zero
 - SO garante que ocorrerá pelo menos uma interrupção dentro de um intervalo de tempo t, colocando no relógio um valor que demore t unidades de tempo para ser decrementado até zero
 - esta atribuição de valor ao relógio é feita imediatamente antes do sistema operacional entregar a CPU para um programa de usuário.

78

Interrupção e Trap

- Uma interrupção não afeta a instrução que está sendo executada pela CPU no momento em que ela ocorre
 - CPU detecta interrupções apenas após o término da execução de uma instrução (e antes do início da execução da instrução seguinte)
- Habilitação e desabilitação de interrupções
 - computadores possuem instruções para mascarar (desabilitar, inibir) o sistema de interrupções
 - enquanto as interrupções estão mascaradas elas podem ocorrer, mas não são sentidas pelo processador
 - as interrupções ficam pendentes (enfileradas) e só serão sentidas quando uma instrução que desmascara as mesmas é executada

Roberto Willrich - INEAF/USC - 09/10/2000
79

Coordenação das atividades da máquina

- Despachante
 - Cada vez que um processo inicia o uso de sua fatia de tempo
 - despachante dispara um circuito temporizador encarregado de medir o quantum
 - Ao final do quantum o temporizador gera uma interrupção
 - processador interrompe o que está realizando
 - bloqueio do processo em andamento
 - registra o ponto da tarefa no qual se foi interrompido
 - passa a tratar a interrupção (rotina de tratamento de interrupção)
 - rotina faz parte do despachante

Roberto Willrich - INEAF/USC - 09/10/2000
80

Coordenação das atividades da máquina

- Despachante
 - No tratamento da interrupção
 - despachante permite que o escalonador atualize a tabela de processos
 - prioridade do processo interrompido poderá ser reduzida e prioridades dos demais aumentadas
 - despachante seleciona o processo de maior prioridade dentre os que se encontram prontos
 - reinicia a operação do temporizador

Roberto Willrich - INEAF/USC - 09/10/2000
81

Coordenação das atividades da máquina

- Despachante
 - No chaveamento de processos
 - Atualizar o estado do processo
 - de maneira que ele pode ser retomado posteriormente
 - Processadores projetados para operarem em sistemas de tempo compartilhado incluem recursos para guardarem tal estado a cada ocorrência de interrupção
 - possuem instruções para recuperar os estados anteriores
 - simplificam a implementação do despachante
 - projeto do sistema influencia no projeto do processador

Roberto Willrich - INEAF/USC - 09/10/2000
82

Coordenação das atividades da máquina

- Despachante
 - As vezes o tempo do processo termina antes do quantum
 - se um processo executar uma solicitação de entrada ou saída, de dados no disco
 - ocorre através de um trap
 - Fatia de tempo será truncada pelo sistema
 - senão ocorreria um desperdício de tempo guardando um resultado do controlador
 - Ações
 - Escalonador atualizará a tabela de processos
 - marcando o processo corrente como estando em estado de espera
 - Despachante fornecerá um novo quantum a outro processo que já está pronto para executar
 - Quando o controlador indicar que a operação foi concluída
 - escalonador reclassifica o processo como pronto para a execução
 - habilitando-o a competir novamente por outra fatia de tempo

Roberto Willrich - INEAF/USC - 09/10/2000
83

Coordenação das atividades da máquina

- Um sistema monoprocessado
 - executando de forma intercalada N processos pode ser visto como se possuísse N processadores virtuais
 - um para cada processo em execução
 - cada processador virtual teria 1/N da velocidade do processador real
 - desprezando-se o overhead existente na implementação da multiprogramação
 - tempo durante o qual o sistema não está produzindo trabalho útil para qualquer usuário
 - tanto no paralelismo físico como no lógico as velocidades relativas com que os processos acessarão dados compartilhados não podem ser previstas

Roberto Willrich - INEAF/USC - 09/10/2000
84

Coordenação das atividades da máquina

- Classificação dos Processos
 - Processos paralelos são denominados concorrentes ou assíncronos
 - acordo com o tipo de interação existente entre eles, podem ser classificados como
 - disjuntos (não interativos), quando operam sobre conjuntos distintos de dados
 - interativos, quando têm acesso a dados comuns
 - » podem ser competitivos, se competirem por recursos,
 - » cooperantes, se trocarem informações entre si

Roberto Willrich - INEAFSC - 09/10/2000

85

Modelo Cliente/Servidor

- Diversas unidades internas a um SO funcionam como processos independentes
 - que também competem por uma fatia de tempo partilhado
 - se comunicam para coordenar suas atividades
 - para escalar um novo processo, o escalonador solicita espaço de memória ao gerenciador de memória
 - para acessar um arquivo em disco, o processo deve obter informação do gerenciador de arquivo
- Modelo Cliente/Servidor
 - simplifica a comunicação entre os processos

Roberto Willrich - INEAFSC - 09/10/2000

86

Modelo Cliente/Servidor

- Modelo Cliente/Servidor
 - Cliente envia solicitações a outras unidades
 - Processo solicita leitura de arquivo
 - Servidor satisfaz as solicitações recebidas dos clientes
 - Gerenciador de arquivo pode funcionar como um servidor
 - fornecendo acesso a arquivos
 - Padronizam os tipos de comunicação existentes em um sistema
 - amplamente adotado em sistemas operacionais
 - se componentes do um SO forem projetados como clientes e servidores, a forma de comunicação entre eles é idêntica
 - quer seja entre componentes dentro de uma máquina ou em máquinas distintas

Roberto Willrich - INEAFSC - 09/10/2000

87

Tratamento da Competição entre Processos

- Distribuição dos recursos da máquina entre os processos
 - uma tarefa comum dos componentes do núcleo de um SO
 - gerenciador de arquivo
 - autoriza o acesso a arquivos existentes e a regiões vazias de disco para a construção de novos arquivos
 - gerenciador de memória
 - aloca espaço de memória
 - escalonador
 - aloca espaço na tabela de processos
 - despachante
 - aloca fatias de tempo
 - é uma tarefa complexa
 - envolve vários problemas que devem ser tratados para não ocorrer falha no sistema
 - para construir um sistema confiável
 - é necessário levar em conta cada possível detalhe do sistema independente de quão insignificante possa parecer

Roberto Willrich - INEAFSC - 09/10/2000

88

Tratamento da Competição entre Processos

- Dispositivos compartilhados
 - Exemplo: máquina com uma única impressora com SO de tempo compartilhado
 - se um processo deseja imprimir seus resultados:
 - deverá solicitar ao SO acesso ao programa acionador de impressora
 - SO deve decidir se o pedido deve ser atendido ou não
 - » conforme esteja a impressora livre ou ocupada por algum processo
 - se a impressora não estiver ocupada
 - » SO atenderá a solicitação e permitirá que o processo solicitante prossiga sua execução
 - se a impressora estiver ocupada
 - » o pedido é negado e o processo é marcado como estando à espera de que a impressora fique disponível
 - se fosse possível o uso da impressora por dois processos
 - a listagem seria embaralhada!

Roberto Willrich - INEAFSC - 09/10/2000

89

Tratamento da Competição entre Processos

- Dispositivos compartilhados
 - Exemplo: máquina com uma única impressora com SO de tempo compartilhado
 - acesso exige que o SO mantenha atualizada a informação do estado de alocação da impressora
 - uma solução possível: empregar uma variável sinalizadora com dois estados (disponível, ocupada)
 - inicialmente o SO atribui o valor disponível
 - em uma impressão, o SO atribui o valor ocupada
 - durante a impressão, os acessos serão negados sempre que a variável sinalizadora indica ocupada.
 - no final da impressão a variável é setada como disponível

Roberto Willrich - INEAFSC - 09/10/2000

90

Tratamento da Competição entre Processos

- Dispositivos compartilhados
 - Exemplo: máquina com uma única impressora com SO de tempo compartilhado
 - problema
 - tarefa de testar e ligar sinalizador requer execução de várias instruções
 - » é possível que a tarefa seja interrompida logo após ser detectada a situação de sinalizador "disponível" e antes de setar o sinalizador como "ocupada"
 - supondo que outro processo requisita impressão após interrupção
 - » sinalizador será verificado como "disponível"
 - » será setado como "ocupada"
 - no retorno do primeiro processo, dois processos terão direito de escrita na impressora!
 - não pode ocorrer interrupções durante o processo de verificar o sinalizador e modificar seu valor
 - solução é desabilitar a interrupção no início deste processo e habilitar a interrupção no final
 - executar uma instrução test-and-set disponíveis em alguns processadores

91

Tratamento da Competição entre Processos

- Semáforo
 - Implementação adequada do sinalizador
 - mesma ideia do semáforo ferroviário
 - correspondendo aos segmentos de trilho que podem conter um trem por vez
 - sequência de instruções que podem ser executadas por um processo a cada vez
 - região crítica

92

Tratamento da Competição entre Processos

- Deadlock
 - Outro problema que pode ocorrer durante a distribuição de recursos
 - paralisação completa
 - situação na qual dois ou mais processos ficam impedidos de prosseguir suas execuções devido ao fato de cada um estar aguardando acesso a recursos já alocados a outro
 - Exemplo
 - um processo tem acesso a impressora mas está esperando pelo dispositivo de fita
 - outro processo tem acesso ao dispositivo de fita mas está aguardando a impressora
 - Ocorrência reduz drasticamente o desempenho do sistema

93

Tratamento da Competição entre Processos

- Deadlock
 - Não ocorre a menos que sejam satisfeitas simultaneamente as três condições seguintes
 - Haver competição por recursos não compartilháveis
 - Serem os recursos solicitados de forma parcial
 - já estando de posse de alguns recursos, um processo volta a solicitar mais recursos em instante posterior
 - Uma vez alocado, um recurso não pode ser retomado pelo sistema
 - a não ser que o processo o libere espontaneamente
 - Para resolver o problema do deadlock
 - deve-se eliminar as três condições acima

94

Tratamento da Competição entre Processos

- Deadlock
 - Outra solução: eliminar a terceira condição
 - possibilitar a retirada pelo SO do recurso alocado pelo processo
 - são conhecidos como esquemas de detecção e correção de deadlocks
 - deadlock pode ocorrer mas pode ser corrigido
 - exemplo:
 - se um processo cria sub-processo e se por acaso a tabela de processos estiver completa
 - » processo é bloqueado
 - administrador pode remover (kill) alguns processos que lotam a tabela

95

Tratamento da Competição entre Processos

- Deadlock
 - Outra solução: tratar as duas primeiras condições
 - solução a) forçando que cada processo reservasse todos os recursos necessários de uma só vez
 - solução b) removendo a competição
 - transformando recursos não compartilhados em virtualmente compartilháveis
 - impressão
 - » todo pedido será atendido
 - » em vez de conectar o processo ao acionador da impressora
 - » processos são conectados a acionadores que armazenarão em arquivo os textos a serem impressos
 - » entrarão em uma fila de impressão comandada por um processo único
 - » processo chamado de spooling

96

| | |
|--|---|
| Roberto Willrich - INFO@FSC - 09/10/2008 97 | <h2>Tratamento da Competição entre Processos</h2> |
| | <ul style="list-style-type: none">• Deadlock<ul style="list-style-type: none">- Problema de utilização de um mesmo arquivo por vários processos<ul style="list-style-type: none">• gerenciador de arquivo concede a vários processos o acesso a um mesmo arquivo<ul style="list-style-type: none">- desde que se trata de acesso apenas leitura• conflitos ocorrerão se vários processos desejarem alterar o arquivo ao mesmo tempo• gerenciador permite o acesso ao arquivo de acordo com as necessidades dos processos<ul style="list-style-type: none">- vários podem ter permissão de leitura- apenas um pode gravar o arquivo em um determinado instante• outros sistemas dividem o arquivo em partes<ul style="list-style-type: none">- de forma que diversos processos podem alterar diferentes partes do arquivo• problemas adicionais devem ser solucionados<ul style="list-style-type: none">- de que maneira os processos são notificados que outros processos estiverem modificando o arquivo? |