

Gerência de Entrada e Saída

Marcelo Johann

Roteiro da disciplina

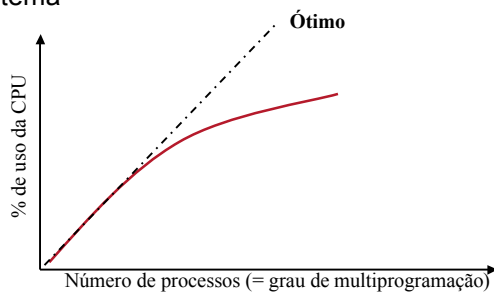
- O Sis. Op. gerencia a CPU:
 - Noção de processos/threads;
 - Escalonamento de CPU para otimizar seu uso.
- O Sis. Op. gerencia a memória (RAM):
 - Recurso mais “escasso” e limitado;
 - Paginação e segmentação são usados:
 - virtualizar a memória,
 - driblar o problema da contigüidade,
 - paginação sob-demanda permite usar mais memória do que fisicamente disponível, graças ao disco.
- Além desses recursos principais, o Sis. Op. gerencia os periféricos
 - Disco, teclado, tela, interface com a rede, ...

INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2009/2

Aula 21 : Slide 2

Ilustração do *thrashing*

- Uso da CPU vs. Número de processos no sistema

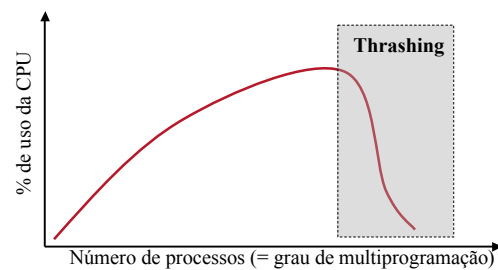


INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2009/2

Aula 21 : Slide 3

Ilustração do *thrashing*

- Uso da CPU vs. Número de processos no sistema



INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2009/2

Aula 21 : Slide 4

Plano da aula

- Gerenciamento dos dispositivos de Entrada/Saída no Sis. Op.
 - Organização do Sis. Op.
 - *Drivers*
 - A interface definida para os processos e suas funções típicas
 - Buffers
 - Tratamento de erro
 - ...

INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2009/2

Aula 21 : Slide 5

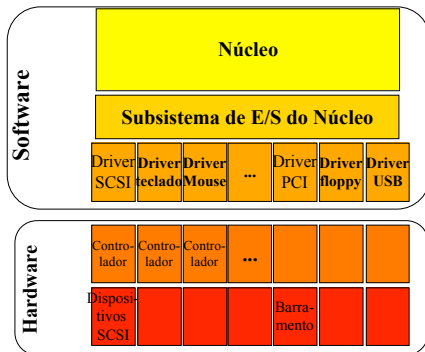
Entrada/Saída e Sis. Op.

- Dispositivos de E/S são dispositivos de HW bastante variados:
 - Orientados a caracteres ou a blocos
 - Síncronos ou não
 - Compartilhados ou não
 - Velocidades diferentes
 - ...
- O Sis. Op. deve ainda mais **abstrair** o acesso a todos esses periféricos de E/S para o usuário final.
 - Independência de dispositivos
- São dispositivos lentos
 - O Sis. Op. deve ser eficiente ao usá-los.

INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2009/2

Aula 21 : Slide 6

Organização em camadas



INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2009/2

Aula 21 : Slide 7

A camada de drivers

- Cada *driver* encapsula as características do dispositivo de HW
 - Conversa com o controlador do dispositivo;
 - É a parte mais específica do Sis. Op.
- Ou vem vendido junto com o HW, para definidos Sis. Op.
 - Em geral, vários *drivers* / dispositivo para acomodar vários Sis. Op.
- Ou vem seguindo um padrão (e.g. SCSI, USB...)

INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2009/2

Aula 21 : Slide 8

Vários tipos de *drivers*/dispositivos

- Orientados a caracteres
 - A transferência de dados se faz byte por byte;
 - Define-se um fluxo de bytes (stream)
 - Operação típica: put(), get()
 - Sempre junto com um mecanismo de *buffer*.
 - Exemplos:
 - Terminal, modem, mouse, impressoras...
- Orientados a blocos
 - Acessa-se blocos de bytes de uma vez;
 - Cada bloco deve ter um endereço;
 - Operações típicas:
 - read(), write()...
 - (possibly) seek()
 - Exemplos: disco, discos removíveis

INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2009/2

Aula 21 : Slide 9

Dispositivos orientados à rede

- O tempo de acesso à rede é ainda diferente do tempo de uso de outros *devices* usuais.
- Por isso, costuma-se definir uma interface específica para *devices* orientados a rede.
 - != put()/get()/read()/write()...
- A interface típica é o **socket**
 - pode-se criar um socket, abri-lo e fechá-lo;
 - Quando aberto, pode-se:
 - Se conectar no socket (connect);
 - Ouvir no socket até alguma coisa chegar (listen);
 - Escrever no socket (write)
 - Pode-se ouvir vários sockets e selecionar (select) uma ação apropriada ao socket de onde veio a informação.
- Existem vários protocolos de rede suportados pelos sockets:
 - Fluxo confiável de bytes (pipe);
 - Fluxo confiável de pacotes;
 - Fluxo não confiável;

INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2009/2

Aula 21 : Slide 10

Exemplo cliente/servidor

```

s = socket(PF_INET,...);
sa.sin_addr.s_addr = INADDR_ANY;
sa.sin_port = htons(7654);
bound = bind(s, &sa, sizeof(struct
sockaddr));
if (bound < 0) printf("ERRO");
listen(s, 3);
while (1) {
    recsize = recvfrom(s, (void *)
hz, 100, 0, (struct sockaddr *)&sa,
fromlen);
    printf("recsize: %d\n", recsize);
    if (recsize < 0) printf("ERRO");
    sleep(1);
}
    
```

Lado servidor

```

sprintf(buffer,
"Hello World!");
s = socket(PF_INET,...);
sa.sin_port =
htons(7654);
bytes_sent =
sendto(s, buffer,
buf_length, ... );
close(s);
    
```

Lado cliente

INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2009/2

Aula 21 : Slide 11

Outras formas de classificação

- Compartilháveis ou não: possibilidade de ser acesso por vários processos no mesmo tempo.
 - Tela vs. Impressora.
- Seqüencial ou randômico
 - Vide o seek().
- Acessível em leitura só, ou em escrita só, ou em escrita/leitura.
- Mapeado na memória / usa registradores
- Usando polling, ou interrupções

INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2009/2

Aula 21 : Slide 12

Chamadas bloqueantes e não-bloqueantes

- Um procedimento **bloqueante** bloqueia o processo enquanto não estiver terminado.
 - Possível tempo de espera.
- Um procedimento **não-bloqueante** retorna imediatamente, e o processo pode prosseguir
 - Obs: os dados retornados nem obrigatoriamente contêm todo o que se queria!
 - São os que estavam prontos no momento da chamada.
- Um procedimento **assíncrono** é não-bloqueante, mas prossegue até obter os dados requisitados.
 - Junto com um teste, o processo que efetuou o procedimento pode obter num dado momento os resultados.

INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2009/2

Aula 21 : Slide 13

O subsistema de E/S

- Basicamente, ele define **interfaces** padronizadas em cima dos *drivers*, para serem usadas pelo núcleo.
- Possibilita o acesso a vários dispositivos, independente de suas peculiaridades.
- Exemplos:
 - interface Ethernet para a rede:
 - Define um endereço IP, um endereço de roteamento (gateway), um prazo máximo de atendimento...
 - Independente da placa de rede específica!
 - Interface do Mouse:
 - Vai tratar eventos tais como 'clíc' ou movimentação.
 - Independente do tipo de mouse!

INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2009/2

Aula 21 : Slide 14

Funções típicas e genéricas da interface

- Denominação
- Bufferização
- Cache
- Tratamento de erros
- Escalonamento de dispositivo
- Compartilhamento e controle de acesso
 - Semáforos!
- Mais funções "backdoor"
 - ioctl() no linux/Unix
 - ioctl(FILE* dev, int cmd, void* args);

INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2009/2

Aula 21 : Slide 15

Tratamento de erros

- O Sis. Op. deve identificar:
 - O tipo de erro
 - Transiente/permanente
 - SW / HW
 - Qual dispositivo é à origem do erro
- Deve retornar o identificador do erro para o usuário.
- Deve logar o erro.

INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2009/2

Aula 21 : Slide 16

Bufferização

- **Buffer não é cache!**
 - Buffer = área de armazenamento temporário de dados.
 - Útil para:
 - Ajustar tamanho de dados
 - Byte vs. Pacotes
 - Ajustar velocidades entre dispositivos
 - Possibilitar o assincronismo
- **Mas cache é um buffer...**
 - Orientado ao desempenho!

INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2009/2

Aula 21 : Slide 17

Implementação de buffer

- Problemas básicos:
 - Manutenção da coerência entre o buffer e os dados originais.
 - tamanho do buffer;
 - O que fazer quando estoura o buffer?
- Aumentar o buffer...
- Usar um outro buffer...
- Usar vários outros buffers
 - Sistema de buffers circulares.

INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2009/2

Aula 21 : Slide 18

Próxima aula...

Organização e
Escalonamento de Disco!