

FACULDADE DE COMPUTAÇÃO E INFORMÁTICA
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO
SISTEMAS OPERACIONAIS – Aula 05 – 2º SEMESTRE/2019
PROF. LUCIANO SILVA

TEORIA: COMUNICAÇÃO ENTRE PROCESSOS (IPC)



Nossos objetivos nesta aula são:

- conhecer o mecanismo de comunicação entre processos (IPC) e sua utilidade na implementação de sistemas operacionais
- reconhecer a importância de esquemas de sincronização em IPC
- praticar com a implementação de IPC no SO MINIX



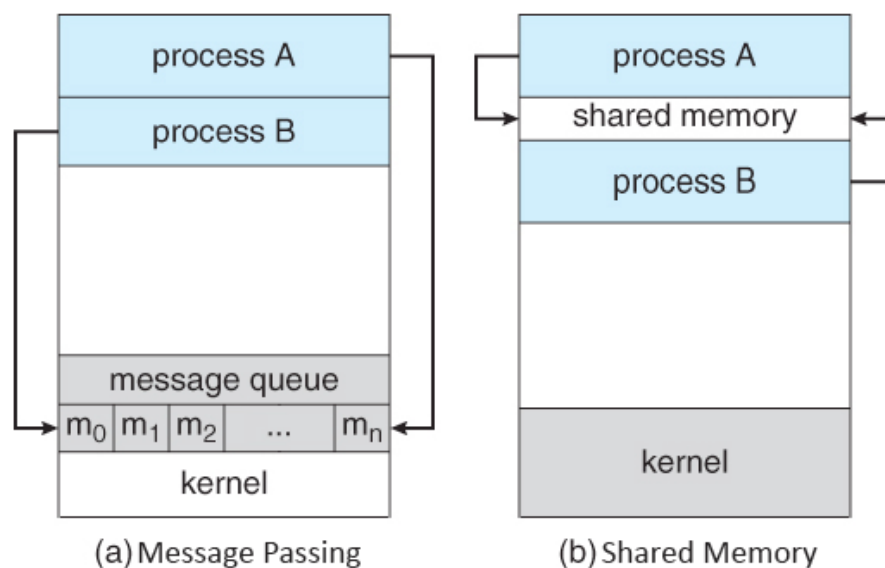
Para esta aula, usamos como referência a **Seção 5.7.6 (Troca de Mensagens)** do nosso livro-texto:

STUART, B.L. **Princípios de Sistemas Operacionais: Projetos e Aplicações**. São Paulo: Cengage Learning, 2011.

Não deixem de ler esta seção depois desta aula!

IPC (INTER-PROCESS COMMUNICATION)

- Processos podem trocar dados através de dois mecanismos importantes de comunicação: **passagem de mensagem (message passing)** e **memória compartilhada (shared memory)**.



- Na técnica de **memória compartilhada**, temos uma região comum aos dois processos, onde eles podem escrever e ler mensagens.
- Um problema clássico em processos, chamado do **Problema do Produtor-Consumidor**, pode ser resolvido com auxílio de IPC por memória compartilhada:

<pre> item nextProduced; while(1){ // Se não há espaço, fica esperando... while((free_index+1) % buff_max == full_index); shared_buff[free_index] = nextProduced; free_index = (free_index + 1) % buff_max; } </pre>	<pre> item nextConsumed; while(1){ // Se não há item produzido, espera... while((free_index == full_index); nextConsumed = shared_buff[full_index]; full_index = (full_index + 1)%buff_max; } </pre>
PROCESSO-PRODUTOR	PROCESSO-CONSUMIDOR

- No exemplo acima, a região compartilhada é o vetor **shared_buff**.

EXERCÍCIO COM DISCUSSÃO EM DUPLAS

1. Explique como funciona o problema do produtor-consumidor.
2. Observe que, nos dois algoritmos apresentados, temos uma variável global **free_index**. Pode ocorrer alguma problema no acesso esta variável ?

IPC (INTER-PROCESS COMMUNICATION) (Continuação...)

- Na técnica de comunicação por **troca de mensagens**, utilizamos duas primitivas de comunicação:
 - **send (p,m)**: envia ao processo p a mensagem m
 - **receive(p,m)**: recebe, do processo p, a mensagem m.
- Vamos reescrever os processos produtor e consumidor com o esquema de troca de mensagens:

<pre>void Producer(void){ int item; Message m; while(1){ receive(Consumer, &m); item = produce(); build_message(&m , item) ; send(Consumer, &m); } }</pre>	<pre>void Consumer(void){ int item; Message m; while(1){ receive(Producer, &m); item = extracted_item(); send(Producer, &m); consume_item(item); } }</pre>
PROCESSO-PRODUTOR	PROCESSO-CONSUMIDOR

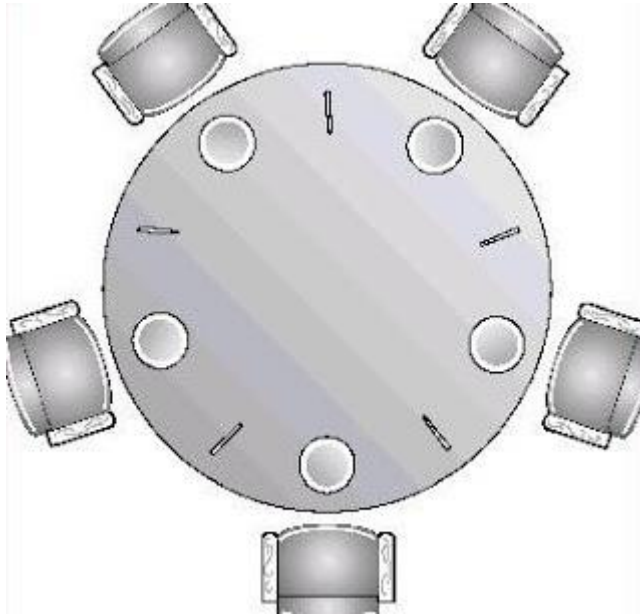
EXERCÍCIO COM DISCUSSÃO EM DUPLAS

1. Explique como funciona o problema do produtor-consumidor escrito com troca mensagens.

2. Qual dos dois processos (Consumidor ou Produtor) deve ser lançado primeiro ?

EXERCÍCIO COM DISCUSSÃO EM DUPLAS

Outro problema clássico envolvendo comunicação entre processos é o **Problema do Jantar dos Filósofos**, onde temos um conjunto de n filósofos fazendo uma refeição numa mesa circular. Entre cada par de filósofos só somente um hashi e são necessários dois hashis para se comer.



Proponha um esquema de sincronização entre os filósofos, implementando as funções `begin_eat(int i)` (filósofo i começa a comer) e `end_eat(int i)` (filósofo i para de comer).

```
void begin_eat (int i){  
    // Para começar a comer, o filósofo precisa dos  
    // hashis  $i$  e  $i+1$ 
```

```
}
```

```
void end_eat(int i){  
    // Para terminar de comer, o filósofo libera os  
    // hashis  $i$  e  $i+1$ 
```

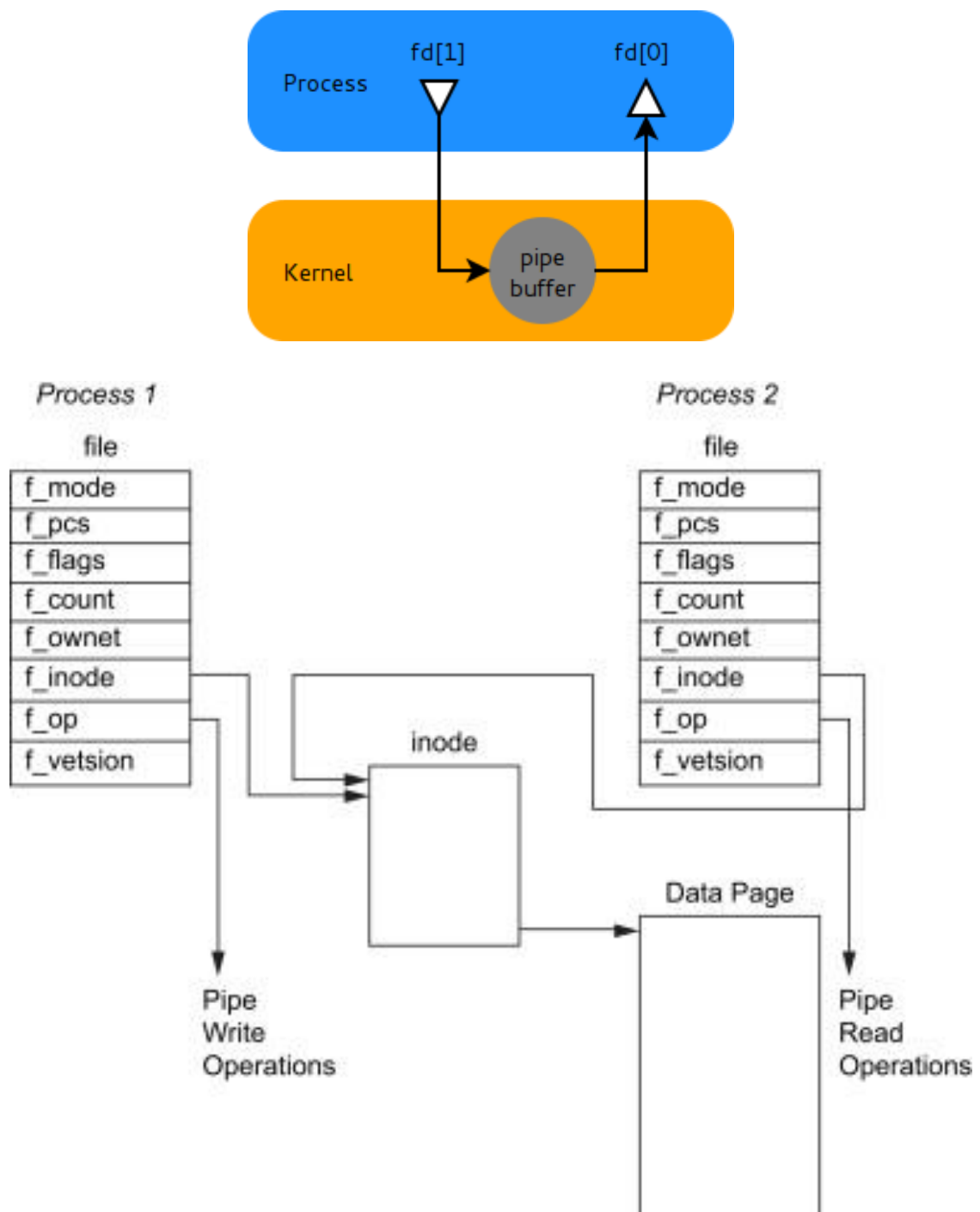
```
}
```

PIPES em LINUX

- O sistema operacional Linux possui um mecanismo muito interessante para comunicação entre processos chamado **pipe** | (de **pipeline**). Este esquema pode ser usado em chamadas do tipo:

```
ls | wc  
who | sort  
cat file.txt | sort | wc
```

- Neste esquema, são utilizadas as primitivas read e write para escrever num arquivo compartilhado entre os dois processos comunicantes:



EXERCÍCIOS EXTRA-CLASSE

1. Produza uma tabela com os principais nomes de sistemas operacionais e seu(s) tipo(s) de IPC.
2. Proponha uma aplicação prática num sistema operacional para o Problema do Produtor-Consumidor.
3. Proponha uma aplicação prática num sistema operacional para o Problema do Jantar dos Filósofos.