

# *Fundamentos de Sistemas de Operação*

---

*Unix Windows NT Netware MacOS DOS/VS Vax/VMS  
Linux Solaris HP/UX AIX Mach Chorus*

*A API de filas de mensagens  
Do UNIX System V*

# IPCs no UNIX System V

- APIs sintaticamente similares, e conceitos comuns
  - Criar e/ou “associar” o processo a um recurso:
    - `xxxget()`: `semget()`, `shmget()`, `msgget()`
  - Alterar atributos ou destruir o recurso:
    - `xxxctl()`: `semctl()`, `shmctl()`, `msgctl()`
  - Operações únicas para um dado tipo de recurso
    - Semáforos: `semop()`
    - Memória partilhada: `shmat()`, `shmdt()`
    - Filas de mensagens: `msgsnd()`, `msgrcv()`

# “Criar” uma fila de mensagens

- `int msgget(int key, int flags)`
  - *Retorna o identificador da fila identificada pela chave única **key**.*
    - *Se a fila não existe e as flags incluem a opção **IPC\_CREAT**, a fila é criada; se tal opção não é incluída, a chamada falha.*
    - *Se a fila já existe e as flags incluem não só a opção **IPC\_CREAT** mas também a opção **IPC\_EXCL**, a chamada falha.*
    - *Se a fila já existe e as flags não incluem a combinação acima descrita (**IPC\_CREAT** | **IPC\_EXCL**), a fila é associada ao processo e o seu identificador devolvido.*
  - *Nota 1: Quando se cria uma fila, deve indicar-se o conjunto de permissões (“modo”); caso contrário não se consegue aceder à fila.*

# Enviar e receber mensagens numa fila

- Mensagens com etiqueta + corpo



- Etiqueta (ou tipo): Um inteiro positivo ( $> 0$ )
  - Corpo: Bloco de  $B \geq 0$  bytes
- Recepção:
    - FIFO
    - FIFO para as mensagens com uma dada etiqueta
    - FIFO para as mensagens com etiqueta  $\leq$  valor  $V$

# Enviar mensagens: msgsnd (1)

- `int msgsnd(int msqid, const void *msgp, size_t msgsz, int msgflg);`
  - *Envia msgsz bytes do corpo da mensagem contida na estrutura apontada por msgp para a fila identificada por msqid.*
  - *A estrutura tem o seguinte formato:*

```
struct msgbuf {  
    long mtype;  
    // corpo: qualquer estrutura de dados;  
}
```

# Enviar mensagens: `msgsnd` (2)

- *Semântica e opções.*
  - *A mensagem é colocada na fila, e a execução do processo continua (i.e., o envio é não-bloqueante), a menos que a fila já esteja cheia; nesse caso,*
    - *Se o valor de `msgflg` é zero, a chamada (processo) bloqueia.*
    - *Se o valor de `msgflg` é `IPC_NOWAIT`, a função retorna imediatamente com erro.*

# Receber mensagens: msgrcv (1)

- `int msgrcv(int msqid, void *msgp, size_t msgsz, long msgtyp, int msgflg);`
  - *Retira da fila identificada por `msqid` uma mensagem e coloca-a na estrutura apontada por `msgp`. O parâmetro `msgtyp` controla o processo de selecção das mensagens; se o seu valor é:*
    - *Zero: a mensagem mais antiga existente na fila é seleccionada;*
    - *Um valor positivo: de entre as mensagens cuja etiqueta tem esse valor, a mais antiga existente na fila é seleccionada;*
    - *Um valor negativo: de entre as mensagens cuja etiqueta é menor ou igual ao valor absoluto (desse valor), a mais antiga existente na fila, para a menor etiqueta existente, é seleccionada.*

# Receber mensagens: `msgrcv` (2)

- **Opções:**
  - Se nenhuma mensagem satisfaz o critério de selecção e `msgflg` tem o valor zero, então a função é bloqueada;
  - Se nenhuma mensagem satisfaz o critério de selecção e `msgflg` tem o valor `IPC_NOWAIT`, então a função retorna imediatamente com erro;
  - Se nenhuma o corpo da mensagem é de dimensão superior a `msgsz`: Se `msgflg` inclui a opção `MSG_NOERROR`, então a cópia é truncada; caso contrário a função falha com erro.

# Remover uma fila de mensagens

- ```
int msgctl(int msqid, int cmd,  
           struct msqid_ds *buf);
```

  - *Para remover a fila identificada por `msqid` basta fazer:*
    - Colocar o comando `cmd` a `IPC_RMID`;
    - Colocar `buf` a `NULL`;