

Sistemas Operacionais

Comunicação Entre Processos



Apresentações Adaptadas do Material do Prof. Marcelo Paravisi

Relembrando... Sessão Revisão



Revisando

- **Tarefas**

- Execução de um fluxo sequencial de instruções, construído para atender uma finalidade específica. Implementada em **Processo**, **Threads**, transações ou jobs.

- **Processos**

- Execução de uma tarefa com seus respectivos recursos (memória, arquivos, dispositivos). Um processo pode então conter várias tarefas, que compartilham esses recursos.

- **Threads**

- Uma thread é um processo simplificado. Um processo pode conter uma ou mais threads. (threads de núcleo e threads de usuário)

Revisando:

- **TROCA DE CONTEXTO**

- **Necessária** quando interrompe a execução de uma tarefa para retornar a ela mais tarde, sem corromper seu estado interno.
- **Dispatcher:** o armazenamento e recuperação do contexto e a atualização das informações providos por um conjunto de rotinas denominado **despachante**.
- **Scheduler:** a escolha da próxima tarefa a receber no processador a cada troca de contexto é estratégica, podendo sofrer influências de diversos fatores, como as prioridades, os tempos de vida e os tempos de processamento restante de cada tarefa.

Revisando:

Os estados e transições do ciclo de vida das tarefas:

- Nova
- Pronta
- Executando
- Suspensa
- Terminada

Revisando:

Os estados e transições válidos das tarefas :

- Nova → Pronta
- Pronta → Executando
- Executando → Pronta
- Executando → Terminada
- Terminada →
- Executando → Suspensa
- Suspensa → Pronta

aula.inf.poa.ifrs.edu.br

Servidor Acadêmico IFRS POA de
acesso público (Alunos SSI)

```
lange@aula: ~  
* Support: https://ubuntu.com/advantage  
  
System information as of Wed 29 Mar 2023 09:07:07 PM -03  
  
System load: 0.43          Processes: 459  
Usage of /: 27.7% of 78.19GB Users logged in: 28  
Memory usage: 32%         IPv4 address for ens160: 200.132.53.138  
Swap usage: 0%  
  
* Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s  
just raised the bar for easy, resilient and secure K8s cluster deployment.  
  
https://ubuntu.com/engage/secure-kubernetes-at-the-edge  
  
45 updates can be applied immediately.  
To see these additional updates run: apt list --upgradable  
  
New release '22.04.2 LTS' available.  
Run 'do-release-upgrade' to upgrade to it.  
  
*** System restart required ***  
Last login: Wed Mar 29 20:16:35 2023 from 201.66.137.232  
lange@aula:~$
```

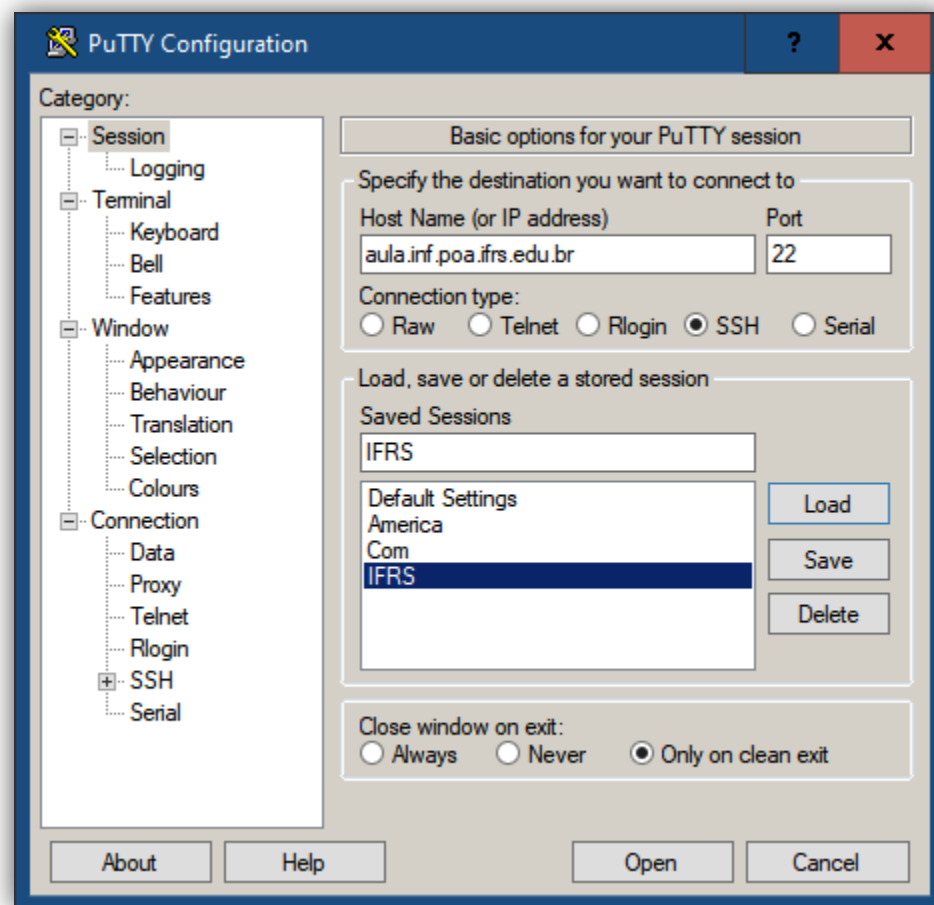
aula.inf.poa.ifrs.edu.br

Conectando no Linux (putty)

Host: aula.inf.poa.ifrs.edu.br

Usuário:

Senha:



Comandos Básicos do Linux

- ls (ls -a) (ls -l) (ls -la)
- clear
- pwd
- cd
- w (who)
- man

Criando um Shell Script (Linux)

Meu Primeiro SHELL SCRIPT

```
#!/bin/bash
contador=1
while [ $contador -ne 50 ];
do
    echo "Mostrando a linha numero
    $contador"
    sleep 1;
    ((contador=$contador+1))
done
```

Meu Primeiro SHELL SCRIPT

~\$ nano script.sh



```
GNU nano 4.8 script.sh
#!/bin/bash
contador=1
while [ $contador -ne 10 ];
do
    echo "Mostrando a linha numero $contador"
    sleep 1;
    ((contador=$contador+1))
done
```

[Read 9 lines]

^G Get Help	^O Write Out	^W Where Is	^K Cut Text	^J Justify	^C Cur Pos	M-U Undo	M-A Mark Text
^X Exit	^R Read File	^_ Replace	^V Paste Text	^T To Spell	^_ Go To Line	M-E Redo	M-C Copy Text

Meu Primeiro SHELL SCRIPT

~\$ chmod u+x script.sh

```
01575855062@aula: ~  
01575855062@aula:~$ nano script.sh  
01575855062@aula:~$ ls -la  
total 44  
drwxr-xr-x 5 01575855062 Domain Users 4096 Aug 25 13:43 .  
drwxr-xr-x 3 root root 4096 Aug 25 13:15 ..  
-rw-r--r-- 1 01575855062 Domain Users 220 Aug 25 13:15 .bash_logout  
-rw-r--r-- 1 01575855062 Domain Users 3771 Aug 25 13:15 .bashrc  
drwx----- 2 01575855062 Domain Users 4096 Aug 25 13:15 .cache  
drwx----- 4 01575855062 Domain Users 4096 Aug 25 13:29 .config  
drwxr-xr-x 3 01575855062 Domain Users 4096 Aug 25 13:40 .local  
-rw-r--r-- 1 01575855062 Domain Users 18 Aug 25 13:39 nano  
-rw-r--r-- 1 01575855062 Domain Users 807 Aug 25 13:15 .profile  
-rw-r--r-- 1 01575855062 Domain Users 150 Aug 25 13:43 script.sh  
-rw----- 1 01575855062 Domain Users 2837 Aug 25 13:39 .viminfo  
01575855062@aula:~$ chmod u+x script.sh  
01575855062@aula:~$ ls -la  
total 44  
drwxr-xr-x 5 01575855062 Domain Users 4096 Aug 25 13:43 .  
drwxr-xr-x 3 root root 4096 Aug 25 13:15 ..  
-rw-r--r-- 1 01575855062 Domain Users 220 Aug 25 13:15 .bash_logout  
-rw-r--r-- 1 01575855062 Domain Users 3771 Aug 25 13:15 .bashrc  
drwx----- 2 01575855062 Domain Users 4096 Aug 25 13:15 .cache  
drwx----- 4 01575855062 Domain Users 4096 Aug 25 13:29 .config  
drwxr-xr-x 3 01575855062 Domain Users 4096 Aug 25 13:40 .local  
-rw-r--r-- 1 01575855062 Domain Users 18 Aug 25 13:39 nano  
-rw-r--r-- 1 01575855062 Domain Users 807 Aug 25 13:15 .profile  
-rwxr--r-- 1 01575855062 Domain Users 150 Aug 25 13:43 script.sh  
-rw----- 1 01575855062 Domain Users 2837 Aug 25 13:39 .viminfo  
01575855062@aula:~$
```

Meu Primeiro SHELL SCRIPT

~\$./script.sh

```
01575855062@aula: ~  
-rw----- 1 01575855062 Domain Users 2837 Aug 25 13:39 .viminfo  
01575855062@aula:~$ ./script.sh  
"Mostrando a linha numero 1"  
"Mostrando a linha numero 2"  
"Mostrando a linha numero 3"  
"Mostrando a linha numero 4"  
"Mostrando a linha numero 5"  
"Mostrando a linha numero 6"  
"Mostrando a linha numero 7"  
"Mostrando a linha numero 8"  
"Mostrando a linha numero 9"  
"Mostrando a linha numero 10"  
"Mostrando a linha numero 11"  
"Mostrando a linha numero 12"  
"Mostrando a linha numero 13"  
"Mostrando a linha numero 14"  
"Mostrando a linha numero 15"  
"Mostrando a linha numero 16"  
"Mostrando a linha numero 17"  
"Mostrando a linha numero 18"  
"Mostrando a linha numero 19"  
"Mostrando a linha numero 20"  
"Mostrando a linha numero 21"  
"Mostrando a linha numero 22"  
"Mostrando a linha numero 23"  
"Mostrando a linha numero 24"  
"Mostrando a linha numero 25"  
"Mostrando a linha numero 26"  
"Mostrando a linha numero 27"  
"Mostrando a linha numero 28"  
"Mostrando a linha numero 29"  
"Mostrando a linha numero 30"  
"Mostrando a linha numero 31"  
"Mostrando a linha numero 32"  
"Mostrando a linha numero 33"  
"Mostrando a linha numero 34"  
"Mostrando a linha numero 35"  
"Mostrando a linha numero 36"
```

Comandos Básicos do Linux

Visualizar o processo em execução

~\$ **ps -aux | grep <usuário>**

```
01575855062@aula: ~  
01575855062@aula:~$ ps -aux |grep script  
0157585+ 633576 0.0 0.0 7024 3608 pts/1 S+ 13:54 0:00 /bin/bash ./script.sh  
0157585+ 633652 0.0 0.0 6432 724 pts/2 S+ 13:55 0:00 grep --color=auto script  
01575855062@aula:~$
```

>> ou top, pstree e htop

Comandos Básicos do Linux

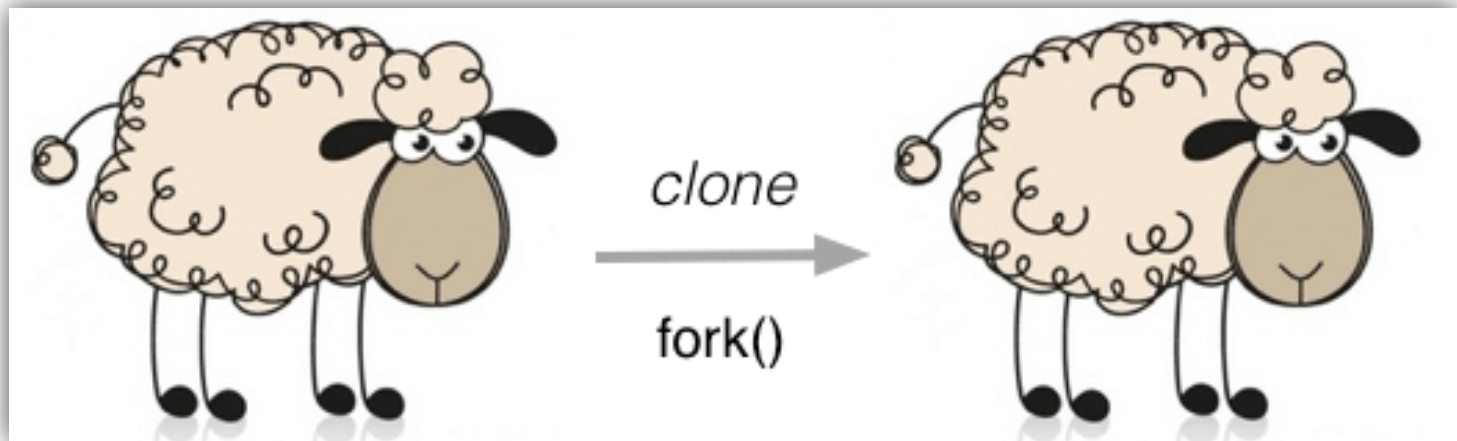
Enviando o sinal de Kill para o processo

~\$ kill -9 633576

```
01575855062@aula: ~  
01575855062@aula:~$ ps -aux |grep script  
0157585+ 633576 0.0 0.0 7024 3608 pts/1 S+ 13:54 0:00 /bin/bash ./script  
0157585+ 633652 0.0 0.0 6432 724 pts/2 S+ 13:55 0:00 grep --color=auto  
01575855062@aula:~$ kill -9 633576  
01575855062@aula:~$  
"Mostrando a linha numero 148"  
"Mostrando a linha numero 149"  
"Mostrando a linha numero 150"  
"Mostrando a linha numero 151"  
"Mostrando a linha numero 152"  
"Mostrando a linha numero 153"  
"Mostrando a linha numero 154"  
"Mostrando a linha numero 155"  
"Mostrando a linha numero 156"  
"Mostrando a linha numero 157"  
"Mostrando a linha numero 158"  
"Mostrando a linha numero 159"  
"Mostrando a linha numero 160"  
"Mostrando a linha numero 161"  
"Mostrando a linha numero 162"  
"Mostrando a linha numero 163"  
"Mostrando a linha numero 164"  
"Mostrando a linha numero 165"  
"Mostrando a linha numero 166"  
"Mostrando a linha numero 167"  
"Mostrando a linha numero 168"  
"Mostrando a linha numero 169"  
"Mostrando a linha numero 170"  
"Mostrando a linha numero 171"  
"Mostrando a linha numero 172"  
"Mostrando a linha numero 173"  
"Mostrando a linha numero 174"  
"Mostrando a linha numero 175"  
"Mostrando a linha numero 176"  
"Mostrando a linha numero 177"  
"Mostrando a linha numero 178"  
"Mostrando a linha numero 179"  
"Mostrando a linha numero 180"  
"Mostrando a linha numero 181"  
"Mostrando a linha numero 182"  
"Mostrando a linha numero 183"  
"Mostrando a linha numero 184"  
Killed  
01575855062@aula:~$
```


FORK

(Criando Processos)



Criando Processos (arquivo moodle)

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <sys/types.h>
4 #include <unistd.h>
5
6 int main(void)
7 {
8     int i;
9     pid_t pid;
10
11     if ((pid = fork()) < 0)
12     {
13         perror("Erro na Criação do fork");
14         exit(1);
15     }
16     if (pid == 0)
17     {
18         //0 código no processo filho
19         int x = 0;
20         while (x < 100)
21         {
22             printf("-PID do Filho: %d - Execução: %d\n", getpid(), x);
23             x++;
24             sleep(1);
25         }
26     }
27     else
28     {
29         //0 código no processo pai
30         int x = 0;
31         while (x < 100)
32         {
33             printf("+PID do Pai: %d - Execução: %d\n", getpid(), x);
34             x++;
35             sleep(2);
36         }
37     }
38
39     printf("Processos Finalizados - Area comum aos processos. \n Digite um valor e Pressione enter:");
40     scanf("Voce Digitou %d", &i);
41     exit(0);
42 }
43
```

Criando Processos

Realizando o FORK (arquivo moodle)

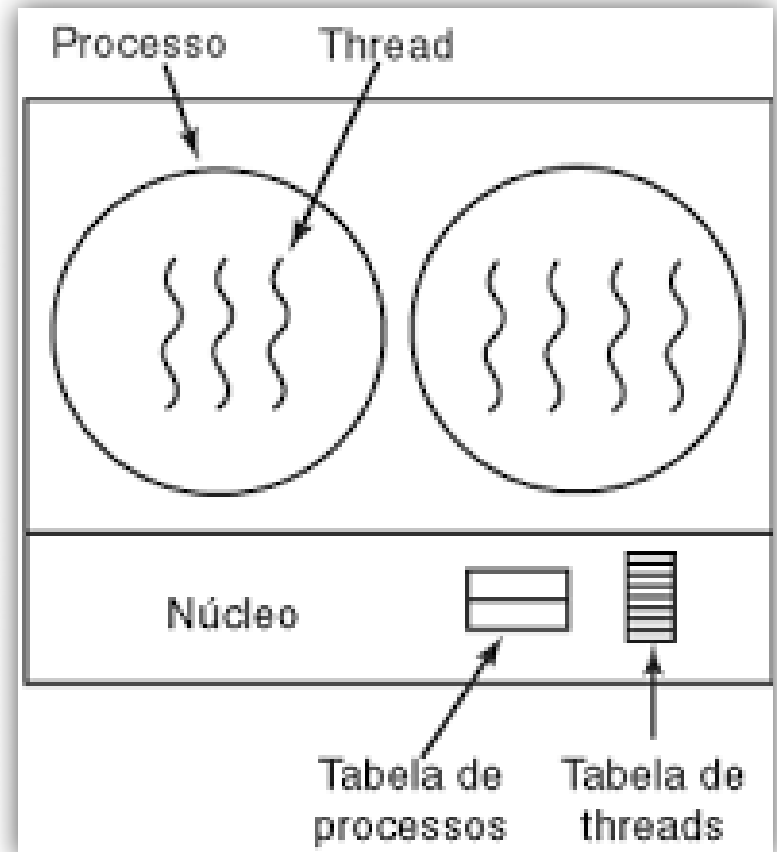
```
01575855062@aula: ~  
01575855062@aula:~$ nano fork.c  
01575855062@aula:~$ gcc fork.c -o appfork  
01575855062@aula:~$ ls -la  
total 64  
drwxr-xr-x 5 marcelasantos Domain Users 4096 Aug 25 14:41 .  
drwxr-xr-x 3 root root 4096 Aug 25 13:15 ..  
-rwxr-xr-x 1 marcelasantos Domain Users 16960 Aug 25 14:41 appfork  
-rw-r--r-- 1 marcelasantos Domain Users 220 Aug 25 13:15 .bash_logou  
t  
-rw-r--r-- 1 marcelasantos Domain Users 3771 Aug 25 13:15 .bashrc  
drwx----- 2 marcelasantos Domain Users 4096 Aug 25 13:15 .cache  
drwx----- 4 marcelasantos Domain Users 4096 Aug 25 13:29 .config  
-rw-r--r-- 1 marcelasantos Domain Users 808 Aug 25 14:33 fork.c  
drwxr-xr-x 3 marcelasantos Domain Users 4096 Aug 25 13:40 .local  
-rw-r--r-- 1 marcelasantos Domain Users 18 Aug 25 13:39 nano  
-rw-r--r-- 1 marcelasantos Domain Users 807 Aug 25 13:15 .profile  
-rw----- 1 marcelasantos Domain Users 2837 Aug 25 13:39 .viminfo  
01575855062@aula:~$ ./appfork  
+PID do Pai: 636227 - Execucao: 0  
-PID do Filho: 636228 - Execucao: 0  
-PID do Filho: 636228 - Execucao: 1  
+PID do Pai: 636227 - Execucao: 1  
-PID do Filho: 636228 - Execucao: 2  
-PID do Filho: 636228 - Execucao: 3
```



```
01575855062@aula: ~  
-PID do Filho: 636256 - Execucao: 01575855062@aula: ~$ ps -aux | grep app  
-PID do Filho: 636256 - Execucao: marcel+s+ 636255 0.0 0.0 2488 516 pts/2 S+ 14:43 0:00 ./appfork  
+PID do Pai: 636255 - Execucao: marcel+s+ 636256 0.0 0.0 2488 76 pts/2 S+ 14:43 0:00 ./appfork  
-PID do Filho: 636256 - Execucao: marcel+s+ 636271 0.0 0.0 6300 720 pts/1 S+ 14:43 0:00 grep --color=auto app  
-PID do Filho: 636256 - Execucao: 01575855062@aula: ~$ ps -aux | grep app  
+PID do Pai: 636255 - Execucao: marcel+s+ 636255 0.0 0.0 2488 516 pts/2 S+ 14:43 0:00 ./appfork  
-PID do Filho: 636256 - Execucao: marcel+s+ 636256 0.0 0.0 2488 76 pts/2 S+ 14:43 0:00 ./appfork  
+PID do Pai: 636255 - Execucao: marcel+s+ 636277 0.0 0.0 6300 656 pts/1 S+ 14:43 0:00 grep --color=auto app  
-PID do Filho: 636256 - Execucao: 01575855062@aula: ~$  
+PID do Pai: 636255 - Execucao:  
-PID do Filho: 636256 - Execucao:  
-PID do Filho: 636256 - Execucao:  
+PID do Pai: 636255 - Execucao:  
-PID do Filho: 636256 - Execucao:  
-PID do Filho: 636256 - Execucao:  
+PID do Pai: 636255 - Execucao:  
-PID do Filho: 636256 - Execucao:  
-PID do Filho: 636256 - Execucao:  
+PID do Pai: 636255 - Execucao:  
-PID do Filho: 636256 - Execucao:  
-PID do Filho: 636256 - Execucao:
```

Threads

(Criando Fluxo de Execução)



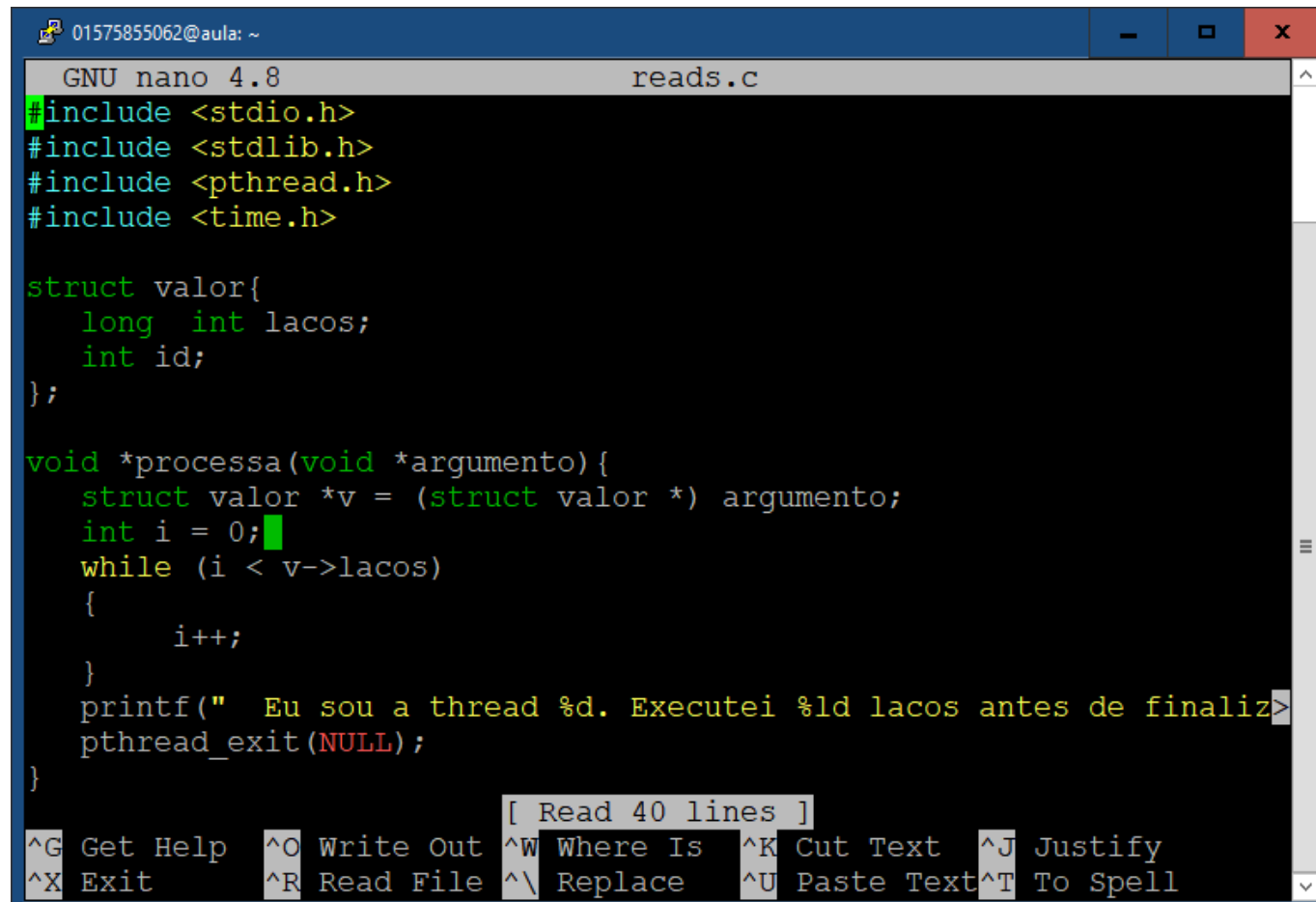


Criando Processos (arquivo moodle)

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <pthread.h>
4  #include <time.h>
5  #include <unistd.h>
6
7  struct valor{
8      int lacos;
9      int id;
10 };
11
12 void *processa(void *argumento){
13     struct valor *v = (struct valor *) argumento;
14     int i = 0;
15     while (i < v->lacos)
16     {
17         i++;
18         sleep(1);
19     }
20     printf(" Eu sou a thread %d. Executei %d lacos antes de finalizar\n",v->id,v->lacos);
21     pthread_exit(NULL);
22 }
23
24 int main(){
25     pthread_t programas[10];
26     int execute,i;
27     struct valor *v;
28
29     srand(time(NULL));
30
31     for (i=0;i<2;i++){
32         v = (struct valor *) malloc(sizeof(struct valor *));
33         v->lacos = (200) ;
34         v->id = i;
35
36         printf("Criando a thread <%d> com <%d> lacos\n",i,v->lacos);
37         execute = pthread_create(&programas[i],NULL,processa,(void *)v);
38     }
39     pthread_exit(NULL);
40 }
41
```

Criando Processos

~\$ nano threads.c



```
01575855062@aula: ~
GNU nano 4.8 reads.c
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>
#include <time.h>

struct valor{
    long int lacos;
    int id;
};

void *processa(void *argumento){
    struct valor *v = (struct valor *) argumento;
    int i = 0;
    while (i < v->lacos)
    {
        i++;
    }
    printf(" Eu sou a thread %d. Executei %ld lacos antes de finaliz>
    pthread_exit(NULL);
}

[ Read 40 lines ]
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify
^X Exit ^R Read File ^\ Replace ^U Paste Text ^T To Spell
```

Criando Processos

Compilando & executando

~\$ gcc -o reads reads.c -lpthread

~\$./reads

```
01575855062@aula: ~
01575855062@aula:~$ gcc -o reads reads.c -lpthread
01575855062@aula:~$ ./reads
Criando a thread <0> com <200000000000> lacos
Criando a thread <1> com <200000000000> lacos
```

```
01575855062@aula: ~
1  [|||||||||||||||||||||100.0%]  Tasks: 55, 84 thr; 3 running
2  [|||||||||||||||||||||100.0%]  Load average: 1.45 1.22 1.62
3  [| 1.3%]  Uptime: 101 days(!), 05:29:20
4  [| 0.7%]
Mem[|||||||||476M/3.81G]
Swp[| 62.8M/3.82G]
```

PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%	MEM%	TIME+	Command
639967	marcel-san	20	0	19140	1660	1532	R	99.8	0.0	1:15.83	./reads
639966	marcel-san	20	0	19140	1660	1532	R	99.2	0.0	1:15.76	./reads
639990	marcel-san	20	0	8064	3848	3224	R	1.3	0.1	0:00.31	htop
1333	mysql	20	0	1753M	27872	4460	S	0.0	0.7	3:35.05	/usr/sbin/mysqld
607316	www-data	20	0	195M	8536	3868	S	0.0	0.2	0:00.00	/usr/sbin/apache
607318	www-data	20	0	195M	8536	3868	S	0.0	0.2	0:00.01	/usr/sbin/apache
607320	www-data	20	0	195M	8560	3872	S	0.0	0.2	0:00.01	/usr/sbin/apache
607741	www-data	20	0	195M	8536	3868	S	0.0	0.2	0:00.01	/usr/sbin/apache
611136	www-data	20	0	195M	8532	3868	S	0.0	0.2	0:00.02	/usr/sbin/apache
632642	www-data	20	0	195M	8516	3808	S	0.0	0.2	0:00.00	/usr/sbin/apache
632645	www-data	20	0	194M	8140	3688	S	0.0	0.2	0:00.00	/usr/sbin/apache
633257	www-data	20	0	195M	8456	3808	S	0.0	0.2	0:00.00	/usr/sbin/apache
633258	www-data	20	0	194M	6600	2176	S	0.0	0.2	0:00.00	/usr/sbin/apache
633259	www-data	20	0	194M	6600	2176	S	0.0	0.2	0:00.00	/usr/sbin/apache

F1Help F2Setup F3Search F4Filter F5Tree F6SortBy F7Nice -F8Nice +F9Kill F10Quit

Criando Processos

Compilando & executando

~\$ gcc -o reads reads.c -lpthread

~\$./reads

```
lange@aula:~$ pstree
systemd───ModemManager──2*[{ModemManager}]
          └─VGAuthService
            └─accounts-daemon──2*[{accounts-daemon}]
              └─agetty
                └─apache2──10*[{apache2}]
                  └─atd
                    └─cron
                      └─dbus-daemon
                        └─firewalld──{firewalld}
                          └─irqbalance──{irqbalance}
                            └─multipathd──6*[{multipathd}]
                              └─mysqld──29*[{mysqld}]
                                └─networkd-dispat
                                  └─nscd──19*[{nscd}]
                                    └─nslcd──5*[{nslcd}]
                                      └─ntpd──{ntpd}
                                        └─polkitd──2*[{polkitd}]
                                          └─rsyslogd──3*[{rsyslogd}]
                                            └─snapd──14*[{snapd}]
                                              └─snmpd
                                                └─sshd──18*[sshd──sshd──sftp-server]
                                                  └─9*[sshd──sshd──bash]
                                                    └─9*[sshd──sshd──sh──bash]
                                                      └─sshd──sshd──sh──bash──bash
                                                        └─2*[sshd──sshd──bash──bash]
                                                          └─3*[sshd──sshd──sh──sftp-server]
                                                            └─sshd──sshd──bash──thread──2*[{thread}]
                                                              └─2*[sshd──sshd]
                                                                └─sshd──sshd──bash──pstree
                                                                  └─3*[systemd──(sd-pam)]
                                                                    └─systemd-journal
                                                                      └─systemd-logind
                                                                        └─systemd-network
                                                                          └─systemd-resolve
                                                                            └─systemd-udev
                                                                              └─udisksd──4*[{udisksd}]
                                                                                └─unattended-upgr──{unattended-upgr}
                                                                                  └─upowerd──2*[{upowerd}]
                                                                                    └─vmttoolsd──2*[{vmttoolsd}]
lange@aula:~$
```

Criando Processos

Compilando & executando

~\$ gcc -o reads reads.c -lpthread

~\$./reads

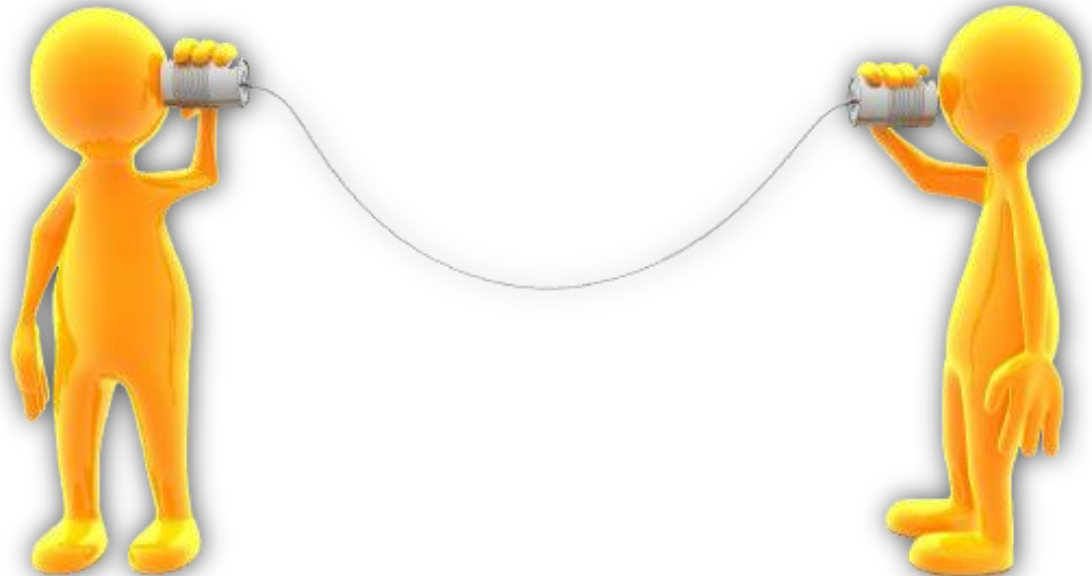
```
lange@aula:~$ ps -aux |grep thread
root          2  0.0  0.0   0      0 ?        S   Feb24   0:00 [kthreadd]
lange      664017  0.0  0.0   0      0 pts/29    Zl+  21:37   0:00 [thread] <defunct>
lange      664304  0.0  0.0 6432   720 pts/3     S+   21:39   0:00 grep --color=auto thread
lange@aula:~$
```

Criando Processos

Como medir o tempo de execução de uma tarefa?

~\$ time <nome do binário>

Comunicação entre Processos



Comunicação Entre Processos

- Durante a execução dos Processo no Sistema Operacional, estes podem **ser independentes ou cooperativos**.
- A cooperação requer que os processos comuniquem entre si e **sincronizem** suas ações.

Comunicação Entre Processos

Principais motivos para cooperação entre processos

- Compartilhamento de informações
- Velocidade de computação
- Modularidade

Comunicação Entre Processos

Existem diferentes estratégias de comunicação entre processos: **IPC (InterProcess Communication)**

- Sockets (TCP/UDP)
- Fila de mensagens (*message queue*)
- PIPES (PID)
- Memória Compartilhada (Variáveis)

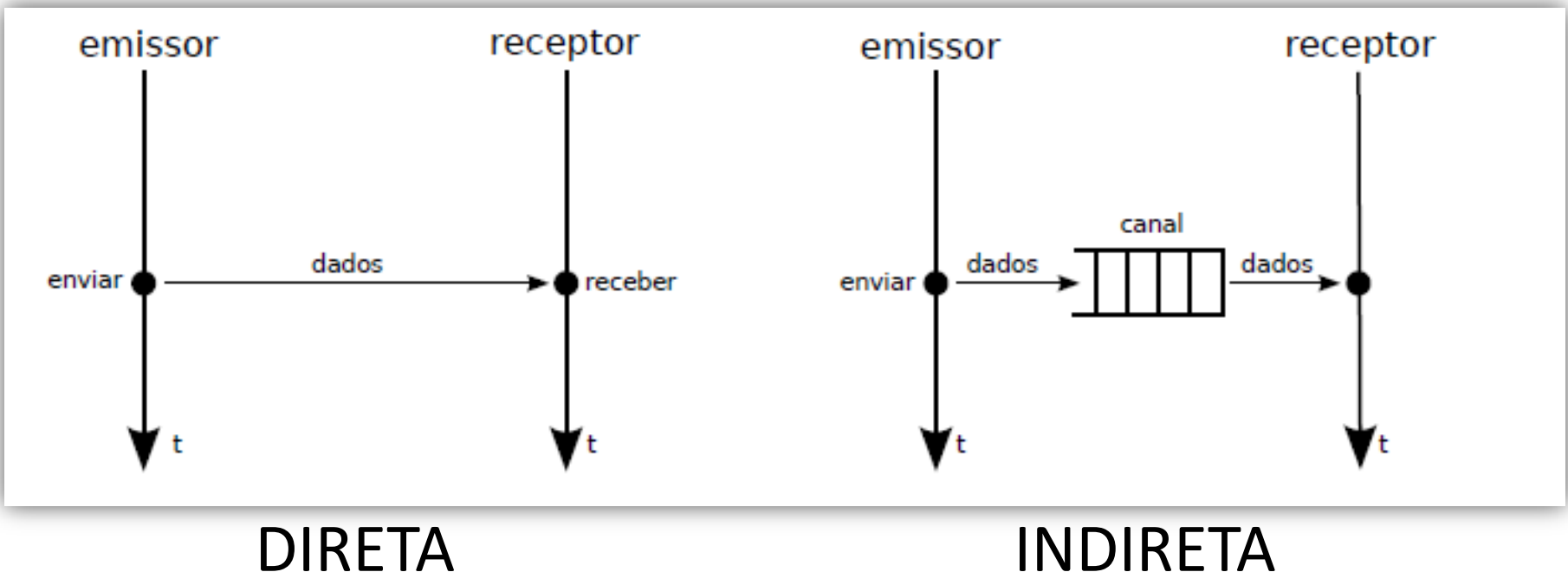
Comunicação Direta x Indireta

DIRETA

O emissor identifica claramente o receptor e vice-versa.

INDIRETA

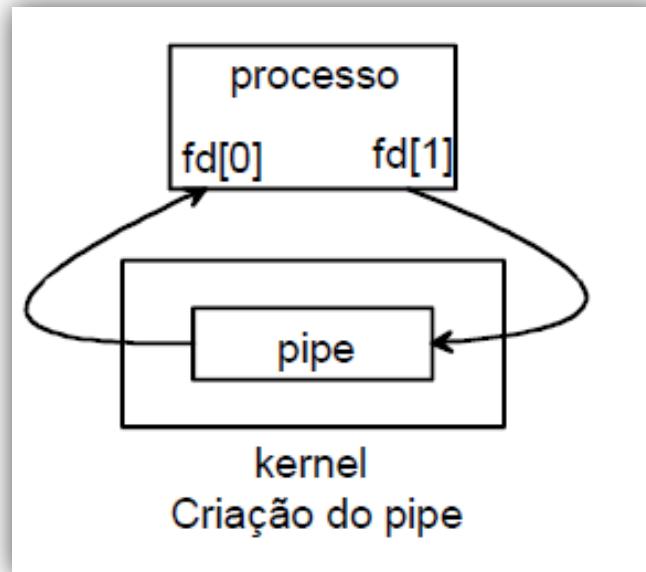
O emissor e receptor não precisam se conhecer, pois não interagem diretamente entre si.



NÚMERO DE PARTICIPANTES DA COMUNICAÇÃO

1:1 mensagem é recebida por receptor

Exemplos: **pipes**



Os **pipes** constituem um canal de comunicação unidirecional entre processos com um ascendente comum (entre um pai e um seu descendente).

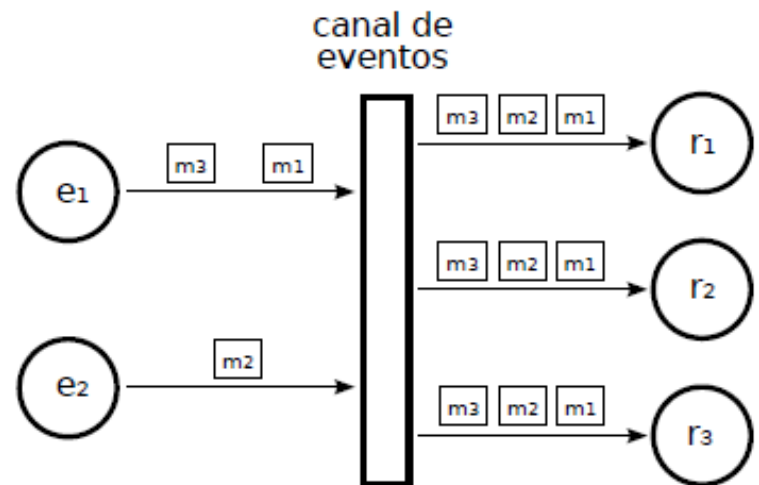
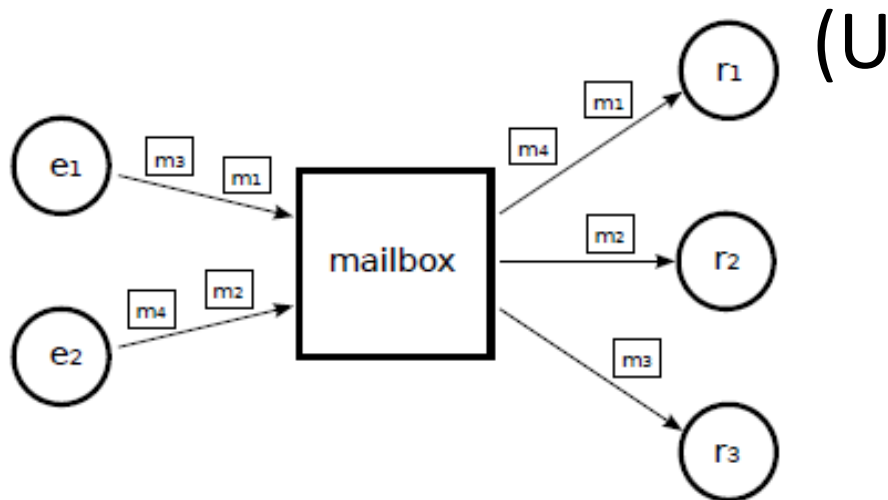
NÚMERO DE PARTICIPANTES DA COMUNICAÇÃO

M:N

1 ou M emissores enviam mensagens

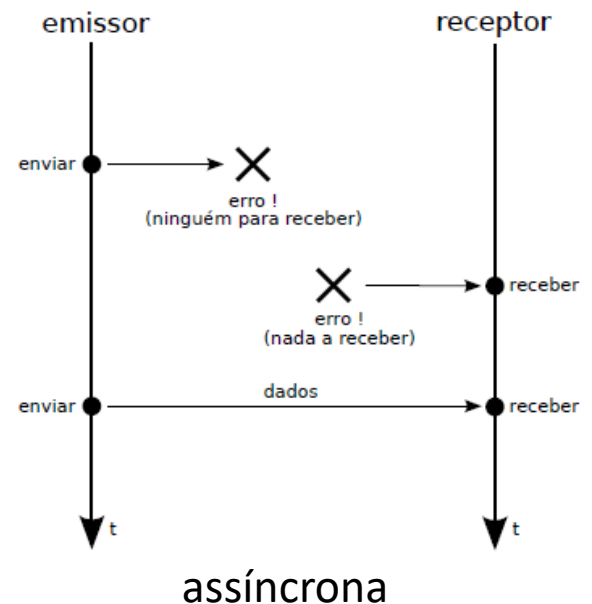
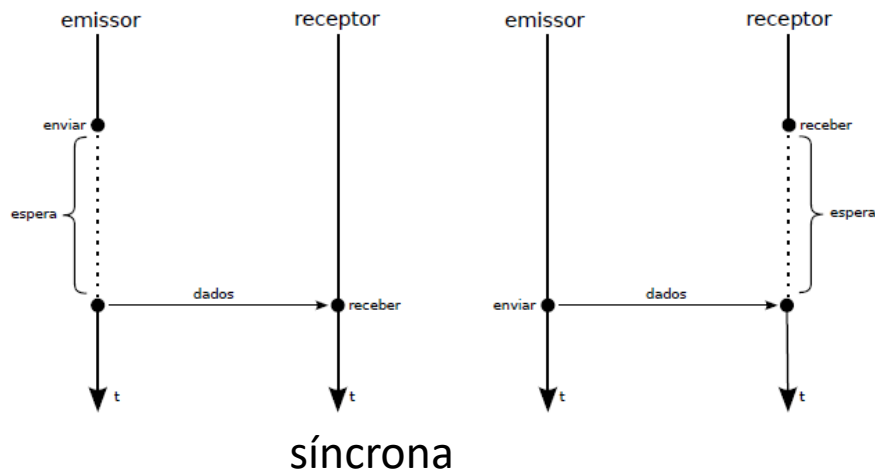
1 ou N receptores recebem mensagens

Exemplos: message queues (linux)



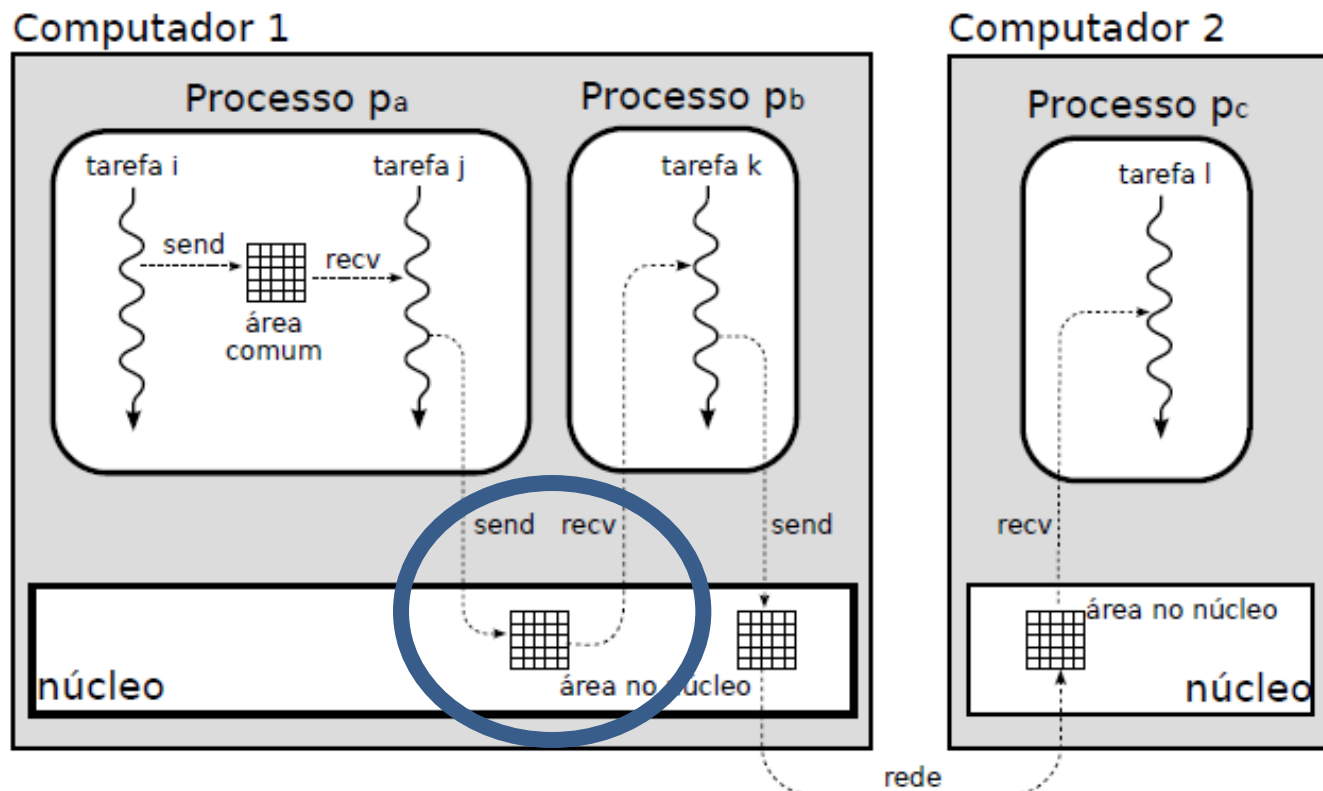
FILAS DE MENSAGENS

- O envio e recepção ordenada de mensagens *tipadas* entre processos locais.
- Envio e recepção podem ser síncronas ou assíncronas (escolha do programador)



FILAS DE MENSAGENS

- O produtor de mensagens deve ser executado após o consumidor, pois é este último quem cria a fila de mensagens
- As mensagens transitam apenas pela memória do núcleo




PIPES

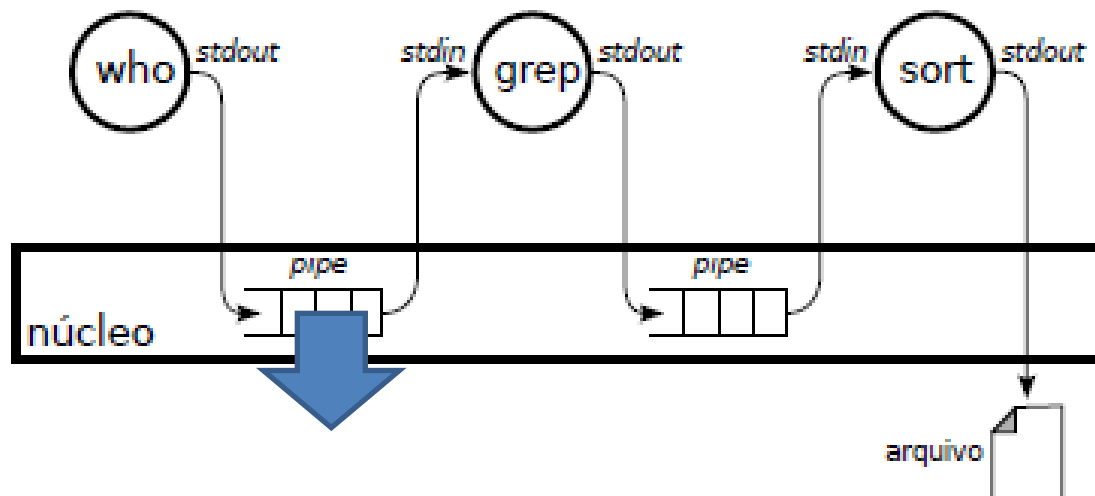
Muito utilizado para conectar stdout com stdin no shell

Exemplo:

pipe pipe



```
# who | grep ifrs | sort > login.txt
```



Buffer de 4KB no linux

MEMÓRIA COMPARTILHADA

- É gerenciados pelo núcleo
- Acesso ao conteúdo sem intermediação ou coordenação do núcleo
- Pode exigir mecanismos de coordenação para evitar condições de corrida.

MEMÓRIA COMPARTILHADA

- É gerenciados pelo núcleo
- Acesso ao conteúdo sem intermediação ou coordenação do núcleo
- Pode exigir mecanismos de coordenação para evitar condições de corrida.

SEÇÃO CRÍTICA

- Devem existir mecanismos para assegurar que somente um processo possa acessar um determinado recurso durante um certo tempo
- Estabelecem-se sessões de código que envolvem o recurso, denominadas seções críticas, e organiza-se o acesso a elas de tal modo que somente uma delas pode ser executada por um processo de cada vez

SEÇÃO CRÍTICA

Um processo impede que outros processos acessem as seções críticas que possui um determinado recurso compartilhado quando ele estiver acessando

Quando um processo finaliza a execução da seção crítica, outro processo pode executá-la

Mecanismo conhecido como **exclusão mútua**

SEÇÃO CRÍTICA

O Mecanismo mais simples de assegurar exclusão mútua para acesso a seções críticas é através de uma **FLAG**

A FLAG é uma variável de 1 bit que possui o valor 1 quando existe algum processo na seção crítica e 0 quando nenhum processo está na seção crítica

SEÇÃO CRÍTICA

Um processo que chega a seção crítica e a FLAG liberada pode entrar, bloqueando para prevenir que nenhum outro processo acesse.

Assim que o processo finaliza a execução da seção crítica, ele libera a FLAG e sai da sessão crítica.

SEÇÃO CRÍTICA - DEADLOCK

Pode ocorrer com dois processos quando um deles precisa de um recurso que está com outro e esse precisa de um recurso que está com o primeiro.

Exclusão mútua pode ser implementado utilizando **Semáforos**.

SEÇÃO CRÍTICA - DEADLOCK

```
1  #include <stdio.h>
2  #include <pthread.h>
3
4  pthread_mutex_t MeuSemaforo = PTHREAD_MUTEX_INITIALIZER;
5
6  int soma = 0;
7
8  void *contapar(void *param) {
9      int i;
10
11     // pthread_mutex_lock(&MeuSemaforo);
12
13     for (i = 0; i < 1000000; i++) {
14         soma += 2;
15     }
16     // pthread_mutex_unlock(&MeuSemaforo);
17     return NULL;
18 }
19
20
21 void *contaimpar(void *param) {
22     int i;
23
24     // pthread_mutex_lock(&MeuSemaforo);
25
26     for (i = 0; i < 1000000; i++) {
27         soma += 3;
28     }
29     // pthread_mutex_unlock(&MeuSemaforo);
30     return NULL;
31 }
32
33
34 int main() {
35     pthread_t tid1, tid2;
36     pthread_create(&tid1, NULL, contapar, NULL);
37     pthread_create(&tid2, NULL, contaimpar, NULL);
38
39     pthread_join(tid1, NULL);
40     pthread_join(tid2, NULL);
41
42     printf("A Soma das Threads eh %d\n", soma);
43     return 0;
44 }
```

Atividades

Atividade 01

- Usando o código fonte de threads disponível no moodle, crie uma variável com o mesmo nome para serem acessadas em nas threads.
- Durante a execução das threads, mostre o endereço das variáveis (threads diferentes) e atribua um valor aleatório para as variáveis.
- **Responda:**
 - O Endereço de memória das variáveis é o mesmo ou é diferente?
 - O valor definido nas variáveis é o mesmo ou é diferente?
 - O que isso significa?

Atividade 02

- Usando o código fonte de fork disponível no moodle, crie uma variável com o mesmo nome para serem acessadas em em tres processos, pai e dois filhos.
- Durante a execução das processos, mostre o endereço das variáveis (processos diferentes) e atribua um valor aleatório para as variáveis.
- **Responda:**
 - O Endereço de memória das variáveis é o mesmo ou é diferente?
 - O valor definido nas variáveis é o mesmo ou é diferente?
 - O que isso significa?

Atividade 03

O Programa gera_RANDOM.c está demorando muito para finalizar a execução.

O programa consiste em gerar dois arquivos com valores Randômicos divisíveis conforme critério.

Observando o código, é possível dividir a tarefa para que os arquivos sejam criados em paralelo e não sequencialmente, diminuindo o tempo de execução próximo aos 50%

Atividade 03

Sua missão,

- a) Dividir a execução em DOIS Processos e medir o tempo.
- b) Dividir a execução em DUAS THREADS e medir o tempo.

Qual dos métodos executou mais rápido?

Qual percentual % foi o ganho de tempo?