

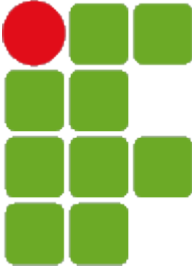
Instituto Federal de Santa Catarina  
CST em Análise e Desenvolvimento de Sistemas  
Introdução à Computação

*Processos*

Prof. Diego da Silva de Medeiros

São José, novembro de 2023

# @ que é um processo?



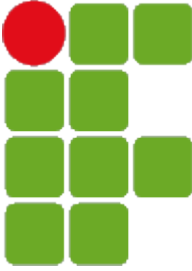
- Forma de representar um programa em execução
- Para executar um comando, é necessário que ele esteja no caminho de procura de arquivos

```
aluno@Maquina:~$ echo $PATH
```

- O path pode ser alterado no arquivo */etc/profile* (todos os usuários) ou *~/.bash\_profile* (usuário atual)

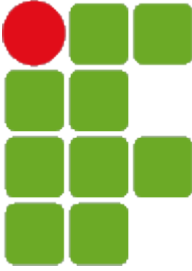
```
export PATH="/Directory1:$PATH"
```

- Comandos que não estão no path podem ser executados usando o endereçamento completo



- Em essência, os processos têm os seguintes estados:
  - Executável      Processo pode ser executado imediatamente
  - Dormente      Processo aguarda alguma coisa para ser executado
  - Zumbi      Processo morto, mas por alguma razão ainda existe
  - Parado      Processo congelado, não pode ser executado

# Primeiro e segundo planos



- Alguns programas como o **gedit** bloqueiam o terminal quando executados
  - Neste caso, diz-se que o programa está em primeiro plano (*foreground*)
  - É possível executar um programa sem que ele trave o terminal, ficando em segundo plano (*background*)
  - Para executar um programa em segundo plano, basta incluir ao final da chamada um caractere '&'
  - O terminal inclusive pode ser fechado (com *exit*) sem que o programa em background seja terminado

```
aluno@Maquina:~$ gedit &
```

- É possível parar a execução de um processo com CTRL+Z, podendo ele ser retomado com os comandos *fg* ou *bg*

```
aluno@Maquina:~$ gedit
```

**[O terminal fica travado na execução do gedit. Pressionando CTRL+Z:]**  
^Z

```
[1]+  Parado                  gedit
```

**[O terminal está liberado para execução de outro programa. O gedit está travado.]**

```
aluno@Maquina:~$ jobs
```

```
[1]+  Parado                  gedit
```

**[Ao colocar o gedit em background, ele volta a funcionar, deixando o terminal liberado:]**

```
aluno@Maquina:~$ bg 1
```

```
[1]+  gedit &
```

```
aluno@Maquina:~$ pwd
```

```
/home/user
```

**[Para retornar o gedit para foreground, usar novamente o jobs e o comando fg:]**

```
aluno@Maquina:~$ jobs
```

```
[1]+  Executando              gedit &
```

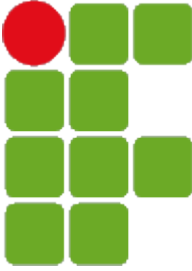
```
aluno@Maquina:~$ fg 1
```

```
gedit
```

**[Para interromper definitivamente, usar CTRL+C:]**  
^C

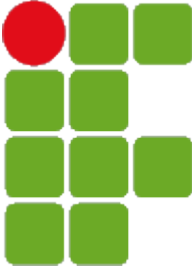
```
aluno@Maquina:~$
```

# *Características de um processo*

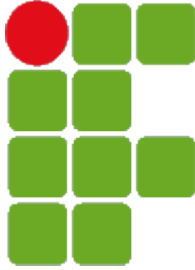


- Para controlar os processos o sistema operacional conta com algumas informações, que são:
  - Usuário proprietário;
  - Estado (em espera, em execução, etc);
  - Prioridade de execução;
  - Recursos de memória

# *O PID e o PPID*



- PID (Process Identifier) - número de identificação único que o sistema dá a cada processo
- PPID (Parent Process Identifier) - número de identificação do processo que criou o processo atual (processo-pai)



- A comunicação do sistema com os processos é dada através de sinais
- Ao receber um sinal, o processo executa as rotinas programadas para o sinal em questão
- Entre os sinais mais importantes estão os seguintes:
  - STOP    pausa a execução de um processo
  - CONT    continua a execução de um processo, após ser pausado com STOP
  - TERM    termina completamente o processo
  - KILL    “mata” um processo, usado em momentos de criticidade



Sinal	Valor	Ação	Comentário
HUP	1	A	Travamento detectado no terminal de controle ou finalização do processo controlado
INT	2	A	Interrupção através do teclado
QUIT	3	C	Sair através do teclado
ILL	4	C	Instrução Ilegal
ABRT	6	C	Sinal de abortar enviado pela função abort
FPE	8	C	Exceção de ponto Flutuante
KILL	9	AEF	Sinal de destruição do processo
SEGV	11	C	Referência Inválida de memória
PIPE	13	A	Pipe Quebrado: escreveu para o pipe sem leitores
ALRM	14	A	Sinal do Temporizador da chamada do sistema alarm
TERM	15	A	Sinal de Término
USR1	30,10,16	A	Sinal definido pelo usuário 1
USR2	31,12,17	A	Sinal definido pelo usuário 2
CHLD	20,17,18	B	Processo filho parado ou terminado
CONT	19,18,25		Continuar a execução, se interrompido
STOP	17,19,23	DEF	Interromper processo
TSTP	18,20,24	D	Interromper digitação no terminal
TTIN	21,21,26	D	Entrada do terminal para o processo em segundo plano
TTOU	22,22,27	D	Saída do terminal para o processo em segundo plano

As letras da coluna Ação tem o seguinte significado:

- A - A ação padrão é terminar o processo.
- B - A ação padrão é ignorar o sinal.
- C - A ação padrão é terminar o processo e mostrar o core.
- D - A ação padrão é parar o processo.
- E - O sinal não pode ser pego.
- F - O sinal não pode ser ignorado.

# Verificando processos

- O comando ***ps*** é usado:
  - Sem opções o comando mostra apenas os processos do usuário atual
- Sintaxe: ***ps*** *opcoes*
- Opções interessantes:
  - a todos os processos existentes;
  - l mais campos no resultado;
  - f árvore de execução dos processos;
  - u nome do usuário que iniciou o processo, a hora de início, consumo de recursos;
  - x exibe os processos que não estão associados a terminais;
  - w se o resultado de processo não couber em uma linha, essa opção faz com que o restante seja exibido na linha seguinte.

# *Campos do comando ps*

- Os campos retornados, em todas as opções são:
- USER            nome do usuário dono do processo;
- UID             número de identificação do usuário dono do processo;
- PID             número de identificação do processo;
- PPID            número de identificação do processo pai;
- %CPU            porcentagem do processamento usado;
- %MEM           porcentagem da memória usada;
- VSZ             indica o tamanho virtual do processo;
- RSS             sigla de Resident Set Size, indica a quantidade de memória usada (em KB);
- TTY             identificador do terminal do processo;
- START          hora em que o processo foi iniciado;
- TIME            tempo de processamento já consumido pelo processo;
- COMMAND      nome do comando que executa aquele processo;

# *Campos do comando ps (2)*

- PRI            valor da prioridade do processo;
- NI            valor preciso da prioridade (geralmente igual aos valores de PRI);
- STAT        indica o estado atual do processo, sendo representado por uma letra:
  - R    executável;
  - D    em espera no disco;
  - T    interrompido;
  - S    Suspenso;
  - Z    Zumbi.
- Essas letras podem ser combinadas e ainda acrescidas de:
  - <    prioridade maior que o convencional;
  - N    prioridade menor que o convencional;
  - s    líder de sessão
  - l    dividido em várias threads
  - +    executado em primeiro plano

- O comando **ps** é como se fosse uma fotografia
- O comando **top** coleta e atualiza as informações de processos regularmente (execução em foreground)
- Opções interessantes:
  - -d               ajusta o intervalo de tempo de atualização
  - <espaço>       atualiza imediatamente as informações
  - <q>             sai do programa
  - <h>             lista de opções e teclas de atalho

# *Enviando sinais aos processos*

- Sinais de comunicação podem ser enviados aos processos
- O comando ***kill*** *-SINAL* *PID* é utilizado
- Após dar o sinal, dar um **[ENTER]** para ver uma mensagem

```
aluno@Maquina:~$ ps
```

PID	TTY	TIME	CMD
1741	pts/2	00:00:00	bash
2443	pts/2	00:00:00	ps

```
aluno@Maquina:~$ gedit &
```

```
[1] 2444
```

**[Note que o PID do processo aparece quando colocamos em BG]**

```
aluno@Maquina:~$ ps
```

PID	TTY	TIME	CMD
1741	pts/2	00:00:00	bash
2444	pts/2	00:00:00	gedit
2446	pts/2	00:00:00	ps

```
aluno@Maquina:~$ kill -KILL 2444
```

```
aluno@Maquina:~$
```

**[ENTER]**

```
[1]+  Morto
```

gedit

# Kill - outro exemplo

```
aluno@Maquina:~$ ps
```

PID	TTY	TIME	CMD
1741	pts/2	00:00:00	bash
2467	pts/2	00:00:00	ps

```
aluno@Maquina:~$ gedit &
```

```
[1] 2468
```

```
aluno@Maquina:~$ ps l
```

F	UID	PID	PPID	PRI	NI	VSZ	RSS	WCHAN	STAT	TTY	TIME	COMMAND
0	1000	1741	1651	20	0	6784	3364	wait	Ss	pts/2	0:00	bash
0	1000	2468	1741	20	0	82320	18384	poll_s	Sl	pts/2	0:00	gedit
0	1000	2470	1741	20	0	4372	680	-	R+	pts/2	0:00	ps l

```
aluno@Ubuntu-VirtualBox:~$ kill -STOP 2468
```

```
aluno@Ubuntu-VirtualBox:~$ [Após este comando, a janela do Gedit estará  
[completamente travada]
```

```
[1]+  Parado gedit
```

```
aluno@Ubuntu-VirtualBox:~$ ps l
```

F	UID	PID	PPID	PRI	NI	VSZ	RSS	WCHAN	STAT	TTY	TIME	COMMAND
0	1000	1741	1651	20	0	6784	3364	wait	Ss	pts/2	0:00	bash
0	1000	2468	1741	20	0	82320	18384	signal	Tl	pts/2	0:00	gedit
0	1000	2472	1741	20	0	4372	676	-	R+	pts/2	0:00	ps l

```
aluno@Ubuntu-VirtualBox:~$ kill -CONT 2468 [Agora o Gedit voltou a responder]
```

```
aluno@Ubuntu-VirtualBox:~$ ps l
```

F	UID	PID	PPID	PRI	NI	VSZ	RSS	WCHAN	STAT	TTY	TIME	COMMAND
0	1000	1741	1651	20	0	6784	3364	wait	Ss	pts/2	0:00	bash
0	1000	2468	1741	20	0	82320	18384	poll_s	Sl	pts/2	0:00	gedit
0	1000	2473	1741	20	0	4372	676	-	R+	pts/2	0:00	ps l

# Exercício

- Estudar o funcionamento dos seguintes comandos, entendendo a diferença deles para os comandos apresentados na aula
  - Killall
  - Killall5
  - Nohup
  - Nice
  - Fuser
  - Tload
  - Vmstat
  - Pidof
  - pstree