

Controle de tarefas

Descrição de tarefas e sessões

Com o recurso do shell de *controle de tarefas*, uma única instância do shell pode executar e gerenciar vários comandos.

Uma

tarefa é associada a cada *pipeline* inserido em um prompt do shell. Todos os processos em um pipeline fazem parte da tarefa e são membros do mesmo *grupo de processos*. Você pode considerar um pipeline mínimo como um único comando inserido no prompt do shell para criar uma tarefa com apenas um membro.

Somente uma tarefa por vez pode ler a entrada e os sinais gerados pelo teclado em uma janela específica de terminal. Os processos que fazem parte dessa tarefa são processos de *primeiro plano* do *terminal de controle*.

Um processo em

segundo plano no terminal de controle é qualquer outra tarefa associada ao terminal. Processos em segundo plano de um terminal não podem ler a entrada ou receber interrupções geradas pelo teclado a partir do terminal, mas podem gravar no terminal. Uma tarefa em segundo plano pode estar interrompida (suspensa) ou estar em execução. Se uma tarefa executada em segundo plano tentar ler a partir do terminal, será suspensa automaticamente.

Cada terminal é executada na sua própria *sessão*, e pode ter um processo em primeiro plano e qualquer número de processos em segundo plano. Um trabalho executado em sua própria sessão pertence a seu terminal de controle.

O comando

`ps` mostra o nome de dispositivo do terminal de controle na coluna `TTY`. Alguns processos, como *daemons do sistema*, são iniciados pelo sistema, e não a partir de um terminal de controle. Esses processos não são membros de uma

tarefa e não podem ser colocados em primeiro plano. O comando `ps` exibe um ponto de interrogação (?) na coluna `TTY` para esses processos.

Execução de tarefas em segundo plano

É possível iniciar qualquer comando ou pipeline em segundo plano anexando um caractere de "e" comercial (&) ao comando. O shell `Bash` exibe um *número de tarefa* (único para a sessão) e a PID do novo processo filho. O shell não aguarda o processo filho ser concluído; em vez disso, ele exibe o prompt do shell.

```
[user@host ~]$sleep 10000 &
[1] 5947
[user@host ~]$
```

Quando uma linha de comando com um pipe (|) é enviada para o segundo plano, a PID do último comando no pipeline é exibida. Todos os processos de pipeline serão membros dessa tarefa.

```
[user@host ~]$example_command | sort | mail -s "Sort output" &
[1] 5998
```

Use o comando `jobs` para exibir a lista de tarefas para a sessão do shell.

```
[user@host ~]$jobs
[1]+  Running      sleep 10000 &
[user@host ~]$
```

Use o comando `fg` para trazer uma tarefa em segundo plano para o primeiro plano. Use o formato (% *jobNumber*) para especificar o processo em primeiro plano.

```
[user@host ~]$fg %1  
sleep 10000
```

No exemplo anterior, o comando **sleep** é executado em primeiro plano no terminal de controle. O shell em si está suspenso e aguardando a saída desse processo filho.

Para enviar um processo em primeiro plano ao segundo plano, pressione a solicitação de *suspensão* gerada pelo teclado (Ctrl+z) no terminal. A tarefa é colocada em segundo plano e suspensa.

```
[user@host ~]$sleep 10000^Z  
[1]+  Stopped                  sleep 10000  
[user@host ~]$
```

O comando **ps j** exibe informações sobre as tarefas. Para encontrar as informações de sessões e processos, use o comando **ps j**.

- A PID é a única ID do processo do processo.
- A PPID é a PID do *processo pai* deste processo, o processo que o iniciou (ramificou).
- A PGID é a PID do *líder do grupo de processo*, normalmente o primeiro processo no pipeline da tarefa.
- A SID é a PID do *líder da sessão*, o que para uma tarefa normalmente é o shell interativo em execução no terminal de controle.

Neste exemplo, o comando **sleep** agora está suspenso e o estado de processo é **T**.

```
[user@host ~]$ps j  
PPID  PID  PGID  SID TTY          TPGID  STAT   UID    TIME CO  
MMAND  
2764  2768  2768  2768 pts/0        6377  Ss     1000    0:00 /b
```

```
in/bash
 2768  5947  5947  2768 pts/0      6377  T      1000    0:00 sl
leep 10000
 2768  6377  6377  2768 pts/0      6377  R+     1000    0:00 ps
j
```

Use o comando **bg** com a ID da tarefa para começar a executar o processo suspenso.

```
[user@host ~]$bg %1
[1]+  sleep 10000 &
```