

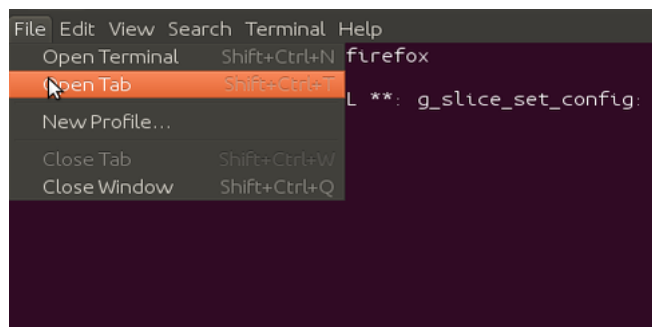
## Jobs, bg, fg, pstree, &

### Transcrição

Agora que já sabemos visualizar quais processos estão rodando, vamos nos aprofundar um pouco mais em sua análise. Vamos inicialmente abrir o Firefox pelo terminal. Para isso digitamos no terminal o comando `firefox`.

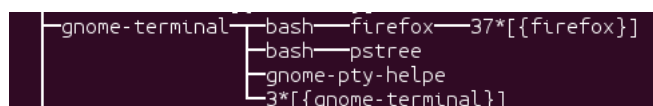
Quando mandamos executar o Firefox por meio do Terminal, o Terminal fica travado enquanto o Firefox estiver aberto, nós não conseguimos mais digitar quaisquer comandos. Isso acontece porque o Terminal está executando o processo do Firefox dentro dele.

Para podermos continuar trabalhando com os comandos do Terminal, basta abrirmos uma nova aba através do menu superior do Terminal:



Agora podemos voltar a utilizar normalmente o terminal enquanto o outro se ocupa com o Firefox.

Em aulas passadas vimos o comando `ps -ef | grep`, que nos mostra os processos abertos. Um outro comando útil para visualizarmos esses processos de uma maneira gráfica é o `pstree`. Com ele podemos ver quais processos estão em execução e dentro de quais programas esses processos estão executando. Veja por exemplo, que o Firefox está rodando dentro do `bash` (aba) no Terminal:



O processo do Firefox está em execução dentro do Terminal e, com isso, trava o `bash` no qual foi executado. Queremos que ele continue executando, mas no *background* desse `bash`, ou seja, sem travar o Terminal. Apertando `CTRL + Z` nós paramos temporariamente o processo.



E para visualizarmos os processos que estão parados, utilizamos o comando `jobs`:

```
guilherme@ubuntu:~$ jobs
[1]+  Stopped                  firefox
```

Para jogar o Firefox no *background*, ou seja, para ser executado em segundo plano, usamos o comando `bg` + seu número de identificação, neste caso executamos `bg 1` ou só `bg` se houver apenas um programa na lista de processos pausados. Este comando apresentará uma saída semelhante a listada abaixo:

```
[1]+  firefox &
```

E assim nosso *bash* fica destravado, mesmo executando o Firefox. A presença do `&` da saída anterior, significa que o programa está rodando no *background*. E se executarmos novamente o comando `jobs`, veremos algo como:

```
[1]+  Running                  firefox &
```

Indicando que o Firefox está realmente executando em *background*. Mas e se quiséssemos fazer o contrário, ou seja, trazer a execução do processo para o *foreground*? Para trazermos o programa para o *foreground* fazemos `fg 1` e assim, teremos nosso *bash* travado novamente.

Também podemos encerrar a execução do programa de vez, para isso basta que apertemos as teclas: `CTRL + C`.

Mas perceba que é um pouco trabalhoso jogarmos um programa para *background*: temos que abri-lo, fazê-lo parar e dar o comando necessário. Podemos fazer isso direto, ao abrirmos o programa da seguinte forma:

```
firefox &
```

Dessa forma já abrimos o programa em *background* e temos o terminal livre para continuarmos com demais comandos necessários para nossas atividades.

Nesta aula vimos os seguintes comandos:

- `jobs` : mostra os processos que estão sendo executados dentro do *bash*;
- `fg` e `bg` : jogam os processos para *foreground* e *background*, respectivamente;
- `[programa] &` : abre o `[programa]` diretamente em *background*;
- `pstree` : mostra todos os processos em um gráfico de árvore.