01

Kill, ps e grep

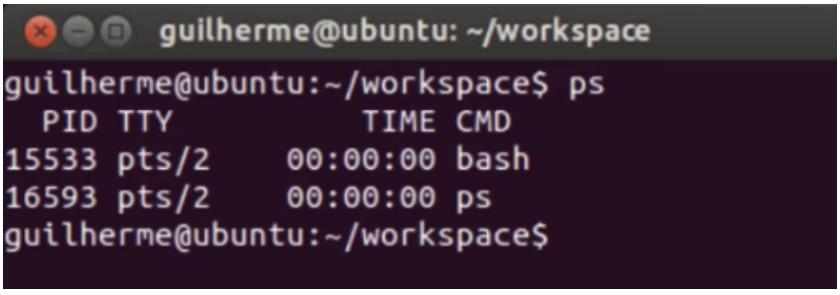
Transcrição

Bem-vindo ao Curso de Linux II: Programas, processos e pacotes! Neste curso, que é continuação do Curso de Linux I: Conhecendo e utilizando o terminal (https://cursos.alura.com.br/course/linux-ubuntu) iremos aprender diversas coisas:

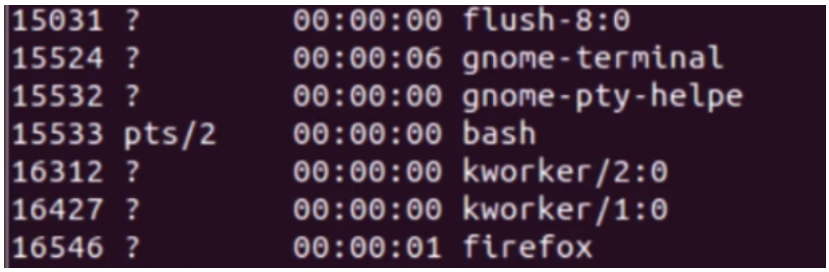
- Descobrir quais são os processos que estão sendo executados em nossa máquina e como finalizar tais processos;
- Executar programas no *background* ou no *foreground*;
- Configurar programas para ter permissão de execução;
- Procurar arquivos no Sistema Operacional;
- Gerenciar novos usuários;
- Configurar as permissões de acesso aos arquivos;
- Configurar variáveis de ambiente;
- Instalar novos programas do zero;
- Inicializar e parar serviços que rodam junto com o início da máquina;
- Compilar e instalar um programa do zero a partir de seu código fonte;
- Fazer o acesso remoto às máquinas que rodam Linux;

Vamos começar?

Cada comando que invocamos é executado pelo CPU. Alguns são finalizados rapidamente como o `ls` e o `mkdir`. Outros demoram mais tempo para parar de executar, alguns exemplos são o `firefox` e o `gedit`. Perceba que se, por exemplo, abrirmos o *Firefox* ou o editor de texto *gedit*, eles ficam em execução. Para sabermos quais processos estão sendo executados no momento, usamos o comando `ps` no Terminal.



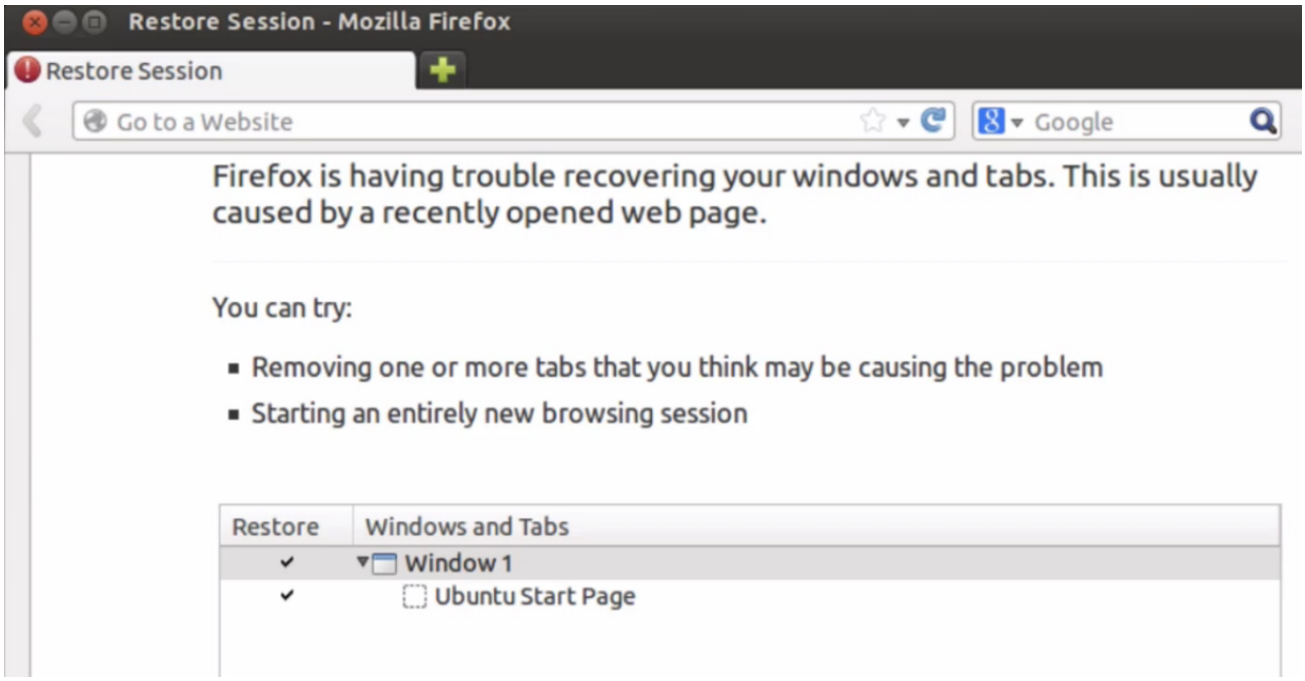
Perceba que só serão mostrados os processos do *bash* (Terminal) atual naquele instante e a execução do comando `ps`. Queremos saber de **todos** os processos em **todo** o sistema, então fazemos `ps -e`. Nos aparecerá uma lista extensa.



Note que o *Firefox* ainda está em execução. A listagem dos processos exibe não só o nome do processo mas também seu ID. Podemos utilizar o ID para finalizarmos um programa usando o comando `kill` + ID, para encerrarmos o *Firefox*, neste caso, fazemos:

```
kill 16546
```

Alguns programas, quando abertos novamente, após matarmos seu processo, mostrarão uma mensagem de alerta para dizer que foram fechados de uma maneira diferente da comum (*clicando no botão de fechar, por exemplo*) e tentam se recuperar, *Firefox* por exemplo, mostra uma mensagem semelhante a da imagem a seguir:

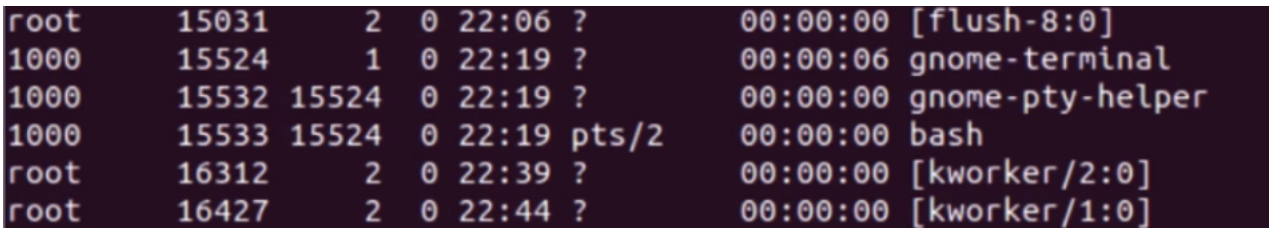


Muitas vezes não conseguimos matar o processo quando o programa trava e para termos certeza de que conseguiremos finalizá-lo, forçando a finalização, podemos usar o modificador `-9` para o comando `kill` da seguinte forma:

```
kill -9 16546
```

A diferença para do `kill` com e sem o modificador `-9` é simples: sem o modificador, o comando solicita o fechamento do programa, dando uma chance para o programa se encerrar sozinho, com o modificador essa chance não existe, o processo é encerrado imediatamente.

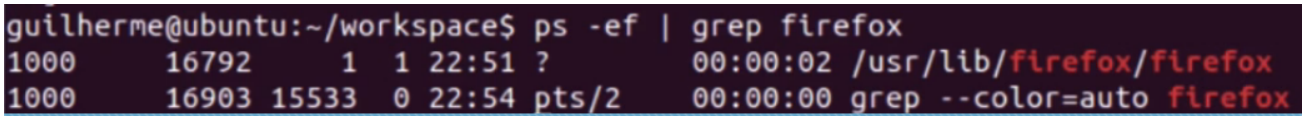
Podemos mostrar mais detalhes sobre os processos que estão em execução, além de seu nome e ID. Para isso fazemos `ps -ef` , este comando mostra a localização dos programas, o instante em que eles foram inicializados e outras informações. A imagem abaixo ilustra a saída desse comando.



Porém ainda fica difícil encontrar um programa que estejamos procurando com essa listagem enorme de processos. Podemos usar o comando `ps -ef | grep NomeDoPrograma` para filtrar os resultados da listagem:

```
ps -ef | grep firefox
```

O comando `grep` é utilizado para filtragem de dados dada uma entrada, neste caso a entrada é o resultado do comando `ps`. O `|` é um redirecionador de saídas de programas para programas, ou seja, ele redireciona a saída do comando `ps` para ser usada como entrada para o comando `grep`. Por isso, são mostrados apenas as linhas que possuam a palavra `firefox`:



O comando `grep` também serve para pesquisarmos linhas com palavras específicas dentro de um arquivo de texto, no caso do arquivo `google.txt` que criamos no curso anterior (https://cursos.alura.com.br/course/linux-ubuntu), podemos pesquisar as linhas que tenham a palavra `California` da seguinte forma:

```
grep California google.txt
```

O *grep* será um comando muito poderoso, pois o utilizaremos sempre que precisarmos encontrar os processos que queremos encerrar ou obter informações.