

Variáveis de ambiente e PATH

Transcrição

Dentro do diretório `workspace` vamos criar um novo programa chamado `oi`, entraremos no diretório usando o comando `cd` e abriremos o `gedit` em background:

```
cd workspace
```

E depois,

```
gedit oi &
```

Dentro do `gedit` digitaremos o código do programa, este simplesmente imprime uma mensagem de `oi, tudo bem?`:

```
echo "Oi, tudo bem?"
```

Após salvar o programa e fechar o `gedit`, podemos visualizar as permissões desse programa, onde veremos que não temos permissão de execução:

```
-rw-rw-r-- 1 guilherme guilherme 21 Jun 12 12:36 oi
```

Para executá-lo, precisamos adicionar a permissão de execução para ele, para isso usamos o comando `chmod` como já vimos anteriormente:

```
chmod +x oi
```

Se executarmos o programa com o `./oi` o mesmo funciona:

```
guilherme@ubuntu:~/workspace$ ./oi  
Oi, tudo bem?
```

Porém, como anteriormente, queremos que ele funcione como os programas normais do sistema, sem que precisemos digitar algo antes do nome dele.

Uma opção é como fizemos com o programa `dorme`: movê-lo para o diretório `/usr/bin`. Mas dessa forma todos os usuários passam a ter acesso a esse programa, o que não queremos. O diretório `bin` existe para armazenar programas globais, aos quais todos os usuários têm acesso. Além disso teríamos que mover todo o programa para esse diretório, o que daria muito trabalho se ele possuísse muitos arquivos.

Precisamos saber onde o `bash` procura os programas e os `scripts` para executar. Esse lugar é o **Path**, uma **Variável de Ambiente**. Se executarmos o comando `env` conseguimos visualizar todas as variáveis de ambiente. Para visualizarmos apenas o `PATH` podemos fazer `env | grep PATH`, e na listagem, a linha que nos interessa é semelhante a apresentada na imagem a seguir:

```
PATH=/usr/lib/lightdm/lightdm:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games
```

Perceba que nessa listagem da variável `PATH` encontra-se o diretório `/usr/bin` e que os diretórios são separados por `:`. Estes diretórios são onde o `bash` procura os programas para serem executados.

Vamos adicionar nessa variável o diretório `workspace`. Se conseguirmos fazer isso, para executarmos nosso programa só será necessário digitar seu nome: `oi`. Fazemos isso da seguinte forma:

```
PATH=$PATH:/home/guilherme/workspace
```

Isso significa que o novo `PATH` será igual a o `PATH` atual (`$PATH`) adicionado a ele (`:`) o diretório `/home/guilherme/workspace`. Podemos verificar se o diretório `workspace` foi adicionado, verificando novamente a variável `PATH`:

```
PATH=/usr/lib/lightdm/lightdm:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/home/guilherme/workspace
```

Agora conseguimos executar o `script "oi"` digitando apenas seu nome, em qualquer diretório que estivermos no `bash`.

```
guilherme@ubuntu:~$ oi
Oi, tudo bem?
```

Porém, se utilizarmos um outro `bash` no Terminal, a variável `PATH` volta a ter os mesmos diretórios por padrão, sem que o `workspace` esteja presente na listagem. O que fizemos no outro `bash` só fica gravado apenas naquela sessão e não globalmente.

Experimente abrir uma nova aba no terminal e verificar a variável `PATH` novamente. Verá que na listagem o diretório `workspace` deixou de existir.

Precisamos de um arquivo o qual diga que, toda vez que abrirmos um `bash`, o `PATH` seja configurado como queremos. Tal arquivo já existe e tem o nome de `.bashrc`. Vamos editá-lo com o `gedit`:

```
gedit .bashrc &
```

Esse arquivo é carregado toda vez que abrirmos uma aba nova no Terminal. Ele já possui vários comandos. O que vamos fazer é: Ao final dele vamos adicionar a seguinte linha e salvar:

```
PATH=$PATH:/home/guilherme/workspace
```

Dessa forma, para toda sessão e todo `bash`, o `PATH` será configurado para possuir o diretório que queríamos, permanentemente.

Experimente verificar novamente, fechar o terminal, abri-lo, tentar executar o programa `oi`, abrir novas abas e verificar a variável `PATH`.

