

LXT0208-File Globbing

📅 Data do post	@August 17, 2022 9:00 AM
🔑 Palavras-chave	Uso básico da Linha de Comandos
☑ Publicado	<input type="checkbox"/>
📌 Pronto	Preparado
☑ IG Post	<input checked="" type="checkbox"/>
⚙ Status	Next

Post-LinkedIn

File globbing é uma operação que combina caracteres especiais com partes de nomes de arquivos para a formação de todas as combinações possíveis desses nomes de arquivos. Esses nomes expandidos em combinações possíveis são passados como parâmetros para comandos do Shell que não poderiam lidar com expressões regulares em seu modo nativo. Leia o artigo na íntegra

Siga-me e veja outros posts no meu Instagram <https://www.instagram.com/idaeciosilvatech/>

Deixe o like para medirmos a repercussão do conteúdo.

Guarde para rever noutro momento.

Compartilha o conteúdo com os amigos e não só

Comente o que acha do conteúdo e da iniciativa e deixe sugestões.

[#cybersecurity](#). [#technology](#) [#linux](#) [#freesoftware](#) [#opensourceoftware](#)

Introdução

Costumamos chamar globbing uma linguagem simples de correspondência de padrões. Os Shells da linha de comando nos sistemas Linux usam essa linguagem para buscar grupos de arquivos cujos nomes correspondam a um padrão específico. O POSIX.1-2017 especifica os seguintes padrões para a correspondência de caracteres:

*****: corresponde a qualquer número de quaisquer caracteres, incluindo nenhum caractere

?: corresponde a qualquer caractere

[]: corresponde a uma classe de caracteres

Em bom português, isso significa que você pode dizer ao seu Shell para procurar por um padrão, em vez de uma sequência literal de texto. Geralmente, os usuários do Linux especificam diversos arquivos com um glob em vez de digitar o nome de cada arquivo. Execute os seguintes comandos:

```
idaseg@debian:~/linux/globs$ ls
question1 question14 question2012 star10 star2002
question13 question15 question23 star1100 star2013
idaseg@debian:~/linux/globs$ ls star1*
star10 star1100
idaseg@debian:~/linux/globs$ ls star*
star10 star1100 star2002 star2013
idaseg@debian:~/linux/globs$ ls star2*
star2002 star2013
idaseg@debian:~/linux/globs$ ls star2*2
star2002
idaseg@debian:~/linux/globs$ ls star2013*
star2013
```

Asterisco (*****)

O Shell expande `*` para qualquer número de qualquer coisa, e assim ele interpreta `star*` qualquer coisa, no contexto relevante, que comece com `star`. Quando você executa o comando `ls star*`, o Shell não executa o programa `ls` com um argumento `star*`, mas sim procura por arquivos no directório actual que correspondam ao padrão `star*` (incluindo apenas `star`), e transforma cada arquivo correspondente ao padrão em um argumento para `ls`:

```
$ ls star*
```

No que diz respeito ao `ls`, isso é o equivalente a:

```
idaseg@debian:~/linux/globs$ ls star*
star10 star1100 star2002 star2013
```

O caractere `*` não significa nada para `ls`. Para comprovar, analise o comando:

```
idaseg@debian:~/linux/globs$ ls star\*
ls: impossível aceder a 'star*': Ficheiro ou pasta inexistente
```

Quando colocamos `\` antes de um caractere, estamos instruindo seu Shell a não interpretá-lo. Neste caso, queremos que `ls` tenha um argumento `star*`, em vez da expansão feita pelo `glob star*`.

Ponto de interrogação (`?`)

O `?` expande para qualquer caractere único. Experimente os seguintes comandos para ver por si mesmo:

```
idaseg@debian:~/linux/globs$ ls
question1 question14 question2012 star10 star2002
question13 question15 question23 star1100 star2013
idaseg@debian:~/linux/globs$ ls question?
question1
idaseg@debian:~/linux/globs$ ls question1?
question13 question14 question15
idaseg@debian:~/linux/globs$ ls question?3
question13 question23
idaseg@debian:~/linux/globs$ ls question13?
ls: impossível aceder a 'question13?': Ficheiro ou pasta inexistente
```

Parenteses-rectos ou colchetes (`[]`)

Os parênteses-rectos `[]` são usados para combinar faixas ou classes de caracteres. Os parênteses-rectos `[]` funcionam como nas expressões regulares POSIX, excepto porque, nos globs, o `^` é usado no lugar de `!`.

Crie alguns arquivos para experimentar:

```
idaseg@debian:~/linux$ mkdir brackets
idaseg@debian:~/linux$ cd brackets/ && touch file1 file2 file3 file4 filea fileb filec file5 file6 file7
```

Os intervalos entre parênteses-rectos `[]` são expressos usando um `-`:

```
idaseg@debian:~/linux/brackets$ ls
file1 file2 file3 file4 file5 file6 file7 filea fileb filec
idaseg@debian:~/linux/brackets$ ls file[1-2]
file1 file2
idaseg@debian:~/linux/brackets$ ls file[1-3]
file1 file2 file3
```

É possível especificar vários intervalos:

```
idaseg@debian:~/linux/brackets$ ls file[1-25-7]
file1 file2 file5 file6 file7
idaseg@debian:~/linux/brackets$ ls file[1-35-6a-c]
file1 file2 file3 file5 file6 filea fileb filec
```

Os parênteses-rectos também podem ser usados para encontrar um conjunto específico de caracteres correspondentes.

```
idaseg@debian:~/linux/brackets$ ls file[1a5]
file1 file5 filea
```

Também usamos o caractere `^` como o primeiro caractere para encontrar todas as correspondências, excepto determinados caracteres.

```
idaseg@debian:~/linux/brackets$ ls file[^a]
file1 file2 file3 file4 file5 file6 file7 fileb filec
```

Conclusão

O recurso de Shell Bash usado para corresponder ou expandir tipos específicos de padrões é chamado globbing. Globbing é usado principalmente para combinar nomes de arquivos ou pesquisar conteúdo em um arquivo. Globbing usa caractere especial para criar o padrão.

O agrupamento de arquivos é um recurso fornecido pelo Shell UNIX / Linux para representar vários nomes de arquivos usando caracteres especiais chamados especiais com um único nome de arquivo. Um caractere especial é essencialmente um símbolo que pode ser usado para substituir um ou mais caracteres.

Gostou do artigo?

Siga-me e veja outros posts no meu Instagram <https://www.instagram.com/idaleciosilvatech/>

Deixe o like para medirmos a repercussão do conteúdo.

Guarde para rever noutro momento.

Compartilha o conteúdo com os amigos e não só

Comente o que acha do conteúdo e da iniciativa e deixe sugestões.

[#cybersecurity](#) [#technology](#) [#linux](#) [#freesoftware](#) [#opensourceoftware](#)