

LXT0212-Extrair informações do começo ao fim dos arquivos

📅 Data do post	@September 14, 2022 9:00 AM (GMT)
🔑 Palavras-chave	Uso básico da Linha de Comandos
☑ Publicado	<input type="checkbox"/>
📌 Pronto	Preparado
☑ IG Post	<input checked="" type="checkbox"/>
🌟 Status	Not started

Post-Linkedin

Último artigo sobre Linux da segunda temporada do que eu denominei Cyber&Tech Articles. Foram cerca de doze artigos que ótimo conteúdo. Desde já agradecer a todos que lêem os artigos e compartilham.

Em breve avizinha-se uma nova temporada. Por enquanto, fiquem com esse último artigo.

Introdução

Neste artigo, trataremos de ferramentas para a manipulação de texto porque são frequentemente usadas por administradores de sistema ou programas para monitorar ou identificar automaticamente informações recorrentes específicas.

Comando `grep`

A primeira ferramenta a ser analisada é o comando `grep`, que é a abreviação da frase “**Global Regular Expression Print**” (impressão de expressão regular global) e sua principal funcionalidade é buscar por um padrão especificado em arquivos. O comando exibe, destacada em vermelho, a linha que contém o padrão especificado.

```
idaseg@debian:~$ grep bash /etc/passwd
root:x:0:0:root:/root:/bin/bash
idaseg:x:1000:1000:idaseg,,,:/home/idaseg:/usr/bin/bash
```

O `grep`, como a maioria dos comandos, também pode ser refinado com opções. As mais comuns são:

Opção	Descrição
<code>-i</code>	A busca não diferencia maiúsculas de minúsculas
<code>-r</code>	A busca é recursiva (pesquisa todos os arquivos dentro do diretório especificado, mais seus subdiretórios)
<code>-c</code>	A busca conta o número de correspondências
<code>-v</code>	Inverte o termo de correspondência para imprimir linhas que não correspondem ao termo de pesquisa
<code>-E</code>	Ativa expressões regulares estendidas (necessárias para alguns dos metacaracteres mais avançados, como <code> </code> , <code>+</code> e <code>?</code>)

O `grep` tem muitas outras opções úteis que podem ser encontradas na página do manual do comando.

Expressões regulares

Expressões regulares são usadas para descrever pedaços de texto em arquivos. As expressões regulares são utilíssimas na extração de dados de arquivos de texto através da construção de padrões. São comumente usadas em scripts ou em programação com linguagens de alto nível, como Perl ou Python.

Ao trabalhar com expressões regulares, é muito importante ter em mente que *todos os caracteres contam*; o padrão é escrito visando corresponder a uma sequência específica de caracteres, conhecida como string. A maioria dos padrões usa os símbolos ASCII normais, como letras, dígitos, sinais de pontuação ou outros símbolos, mas também pode usar caracteres Unicode para corresponder a qualquer outro tipo de texto.

A lista a seguir explica os metacaracteres das expressões regulares usadas para formar os padrões.

Opções	Descrição
.	Corresponde a qualquer caractere único (exceto nova linha)
[abcABC]	Corresponde a qualquer caractere dentro dos colchetes
[^abcABC]	Corresponde a qualquer caractere, exceto os que estão entre colchetes
[a-z]	Corresponde a qualquer caractere no
[^a-z]	Corresponde a qualquer caractere, exceto os do intervalo
sun moon	Busca qualquer uma das strings listadas
^	Início de uma linha
\$	Fim de uma linha

Todas as funcionalidades das expressões regulares também podem ser implementadas através do `grep`.

Além dos metacaracteres explicados anteriormente, as expressões regulares também possuem metacaracteres que permitem a multiplicação do padrão especificado anteriormente:

Opções	Descrição
*	Zero ou mais do padrão anterior
+	Um ou mais do padrão anterior
?	Zero ou um do padrão anterior

A maioria dos metacaracteres é autoexplicativa, mas no início eles podem ser meio difíceis de assimilar.

Conclusão

Esse artigo trouxe conhecimentos sobre alguns metacaracteres em expressões regulares. Também foi bom informar e transmitir o conhecimento de como criar padrões com expressões regulares. E não menos importante, falou-se sobre como buscar em arquivos.