



Universidad Nacional de La Matanza
Ingeniería en Informática-Taller de GNU/Linux

TP N° 6
Expresiones regulares

Objetivos:

–Utilizar expresiones regulares.

Guía:

Construcción de expresiones regulares

Las expresiones regulares se utilizan para hacer búsquedas contextuales y modificaciones sobre textos. Se pueden encontrar en muchos editores de textos avanzados, en programas de análisis gramatical y en muchos lenguajes. Para practicar las expresiones regulares utilizaremos el comando "grep".

. caracter de correspondencia simple (Cualquier caracter)
ejemplo : `'l . nu .'` podría resultar en `lynux, linux` o `linus`

*** caracter de correspondencia múltiple. Cero o más apariciones del caracter previo. Se utiliza en combinación con '.' y con los rangos.**
ejemplo: `f.*o` (texto que empiezan con f y termina con o)

[rango] Especifica un posible rango o conjunto de caracteres
Ejemplo:
`[a-d].*` (texto que empieza con a,b,c o d)
`[axf].*` (texto que empieza con a,x,o f)

[^ rango] Rango negado
Ejemplo: `[^0-9].*` (texto que no empieza con un número)

^ Ancla de comienzo de línea
Ejemplo: `^j` (línea que comienza con j)

\$ Ancla de fin de línea
Ejemplo: `j$` (línea que finaliza con j)

? El signo de interrogación especifica cero o una ocurrencia del caracter precedente.
Ejemplo: `pesos?` coincide con la cadena `peso` o con `pesos`.

+ El símbolo más indica especifica una o más ocurrencias del caracter precedente. Por lo que la expresión `[0-9]+` es equivalente a la expresión `[0-9][0-9]*`.

Nota: Existen mas constructores de expresiones regulares. Se listan los más importantes debido a que el objetivo del trabajo práctico es solo de carácter introductorio.

Comando grep

Muestra todas las líneas de un archivo dado que coinciden con cierto patrón.

Usage: `grep [OPCIONES] PATRON [FILE1] [FILE2] [FILEn]...`
Donde **PATRON** es una expresión regular y **[FILE1] [FILE2] [FILEn]...** son los archivos donde buscar.

Ejemplo:

Para buscar el texto 'hello world' dentro del archivo main.c

```
$ grep 'hello world' main.c
$ grep 'hello world' < main.c
$ cat main.c | grep 'hello world'
```

(las tres líneas hacen lo mismo)

Ejercicios

1) Para trabajar utilizaremos un archivo creado de la siguiente forma:

`ls /bin/ > b.txt` (si el `ls` utiliza colores usar "`\ls /bin/ > b.txt`")

2) buscar dentro de b.txt si está listado el comando 'mount' (utilizar grep)

3) buscar dentro de b.txt comandos que empiecen con "fs"

4) buscar dentro de b.txt comandos que terminen con "fs"

5) buscar dentro de b.txt comandos que tengan un carácter numérico mayor a 2

6) buscar dentro de b.txt comandos que tengan un carácter numérico menor o igual a 2

7) buscar dentro de b.txt comandos que empiecen con 's' y terminen con 'n'

8) buscar dentro de b.txt comandos que empiecen con 'f' o 'p', terminen con 'k', 'g' o 'f' y tengan alguna 'i' en el medio

9) Utilizar 'ps wax' para listar los procesos en ejecución y filtrar con grep las líneas que posean el texto "get".