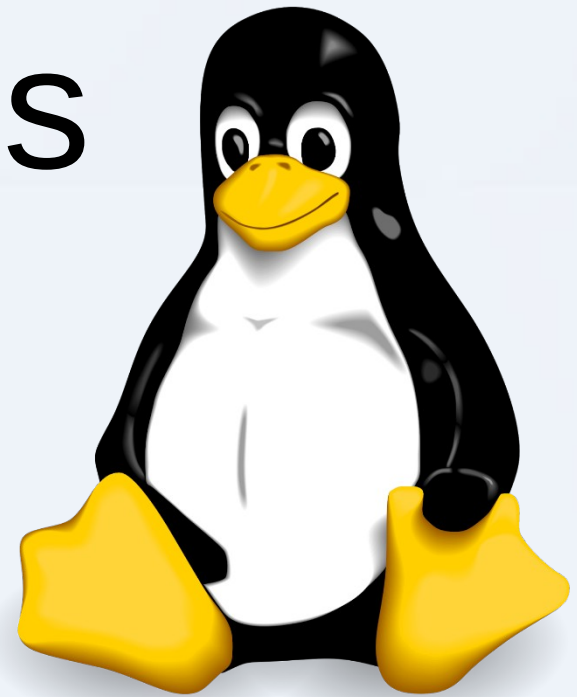




Marcos Pablo Russo
Martín Isusi

Expresiones Regulares



Qué son las expresiones regulares ?

Es una serie de caracteres especiales que permiten descubrir un texto que queremos busca. Por ejemplo, si quisiéramos buscar la palabra “linux” bastaría poner esa palabra en el programa que estemos usando. La propia palabra es una expresión regular. Hasta aquí parece muy simple, pero, ¿ y si queremos buscar todos los números que hay en un determinado fichero ?.

Expresiones regulares

Entro los comandos que vamos a utilizar para esto serán grep y egrep.

?	El ? indica que la expresión precedente es opcional, por lo que representa cero o una ocurrencia de esta. Esto no es lo mismo que usar el ? usado en globbing.
.	El . (punto) es un metacarácter que representa a un solo caracteres.
*	El * (asterisco) es un metacarácter que representa a un conjunto de caracteres.
^	El ^ indica el comienzo de una expresión
\$	El \$ indica el final de una expresión

Expresiones regulares

Crear el siguiente archivo con el contenido :

nano archivo.txt

Root

ROOt

Root Root

Juan Marcos

Marcos Pablo Russo

Luis

alumno@correo.com.ar

4444-5555

3333

Expresiones regulares

Entre los comandos que vamos a utilizar para esto serán **grep** y **egrep**.

grep 'root' archivo.txt

grep 'Root' archivo.txt

grep -i 'root' archivo.txt

grep 'o*' archivo.txt

grep '^\$' /etc/passwd → *Obtener lineas en blanco*

grep '^Root' archivo.txt

Expresiones regulares

Entro los comandos que vamos a utilizar para esto serán grep y egrep.

[.]	Cualquier carácter único de la lista o rango entre paréntesis
[^..]	Cualquier carácter que no esté en la lista o el rango
\	Ignora el significado especial del próximo carácter.
{}	Nos permiten indicar el número de repeticiones del patrón anterior que deben darse.
\$	El \$ indica el final de una expresión

Expresiones regulares

Entre los comandos que vamos a utilizar para esto serán grep y egrep.

```
# grep '[0-9]' archivo.txt
```

```
# egrep '^[0-9]{4}\-[0-9]{4}' archivo.txt
```

```
# egrep '^([a-zA-Z0-9._%+-]+@[a-zA-Z0-9]+\.[a-zA-Z]{2,3})\.[a-zA-Z]{2,3}' archivo.txt
```

```
# egrep -o '^([a-zA-Z]+\ [a-zA-Z]+)' archivo.txt
```

-o	Solo la coincidencia exacta.
----	------------------------------

Expresiones regulares

Entre los comandos que vamos a utilizar para esto serán **grep** y **egrep**.

<code>[:alnum:]</code>	Alfanuméricos.
<code>[:alpha:]</code>	Alfabéticos.
<code>[:cntrl:]</code>	De control.
<code>[:digit:]</code>	Dígitos.
<code>[:graph:]</code>	Gráficos.

Expresiones regulares

Entro los comandos que vamos a utilizar para esto serán grep y egrep.

<code>[lower:]</code>	Minúsculas.
<code>[print:]</code>	Imprimibles.
<code>[punct:]</code>	De puntuación.
<code>[space:]</code>	Espacios.
<code>[upper:]</code>	Mayúsculas.

Expresiones regulares

Entre los comandos que vamos a utilizar para esto serán **grep** y **egrep**.

[:xdigit:]	Dígitos hexadecimales.
------------	------------------------

Práctica 1

Crear el siguiente archivo con el contenido :

nano archivo2.txt

**Juan Marcos
Juan marcos
alumno@correo.com
alumno@correo2.com.ar**

**15-4444-5555
3333-4444**

*** Buscar :**

- **Nombre y apellido todo en minúscula.**
- **Nombre y apellido todo en mayúscula.**
- **Teléfono completo con el 15.**
- **Correo con com.ar.**

Expresiones regulares

Otro de los comandos es sed que nos permite realizar búsquedas, modificar el archivo, etc.

```
# sed 's/Root/ROOT/' archivo.txt
```

```
# sed 's/Root/ROOT/g' archivo.txt
```

```
# sed '1,2 d' archivo.txt
```

```
# sed -e '/Root/d' -e '/ROOT/d' archivo.txt
```

```
# sed -i '/Root/d' archivo.txt
```

s	Buscar.
g	Más de coincidencia.
d	Borra.
-e	Expresión.
-i	Grabar la modificaciones.

Práctica 2

Del archivo2.txt realizar:

- **Reemplazar el contenido de 'Juan Marcos' por 'Marcos' y grabar la modificación.**
- **Borrar la linea 4 y grabar la modificaciones.**

¿ Preguntas ?

