



Práctica 2: Programación en Shell

Objetivo

Durante la práctica pasada aprendiste a usar los comandos de forma individual y a tomar en cuenta sus argumentos; también aprendiste el uso del símbolo de “pipeline” para unir la salida de un comando a la entrada de otro comando. En esta ocasión el utilizarás lo aprendido para crear los denominados “**Shell scripts**”.

Un **shell script** es un programa escrito para la **shell (interpretador de comandos)** de **UNIX/LINUX**; es como un comando nuevo que tú diseñas con todo y sus argumentos. Es muy útil para resolver tareas repetitivas, típicas de los Administradores. Un shell script es un archivo de texto que contiene comandos y que es ejecutable por el sistema. También aprenderás algunos otros nuevos comandos y sobre permisos de acceso y modificación a archivos.

== == Laboratorio == ==

- Para que quedes preparado para resolver lo que se te pide en el Formulario, estudia los conceptos y realiza las actividades indicadas en este documento
 - **Material de Apoyo:** [Lab-02 : Programación en Shell](#); [Bash tutorial](#)
 - **Enlace al formulario:** [Práctica 2: Programación Shell](#).

Actividad 1 :

“grep” es un popular comando para buscar en textos **grep <expresión regular> <archivo>**. Por ejemplo busca la palabra **Navajos** en el archivo **/usr/share/dict/words**, que es el diccionario de palabras de Linux:

```
ccna_admin@ccnalab12:/home/redes1$ grep Navajos /usr/share/dict/words
Navajos
```

O bien:

```
ccna_admin@ccnalab12:/home/redes1$ cat /usr/share/dict/words | grep Navajos
Navajos
```

Recuerda que el primer comando (**cat <archivo1> <archivo 2> ...**) se encarga de desplegar (más propiamente,

“enviar a standard output”) el resultado de la concatenación de dos archivos; o el contenido de un archivo si sólo se menciona uno solo en el comando. La salida del comando cat se “pipea” a la entrada del grep.

Otro ejemplo: buscar todo lo que empiece con "nav" en dicho archivo y trabajar con ellos.

```
ccna_admin@ccnalab12:/$ more /usr/share/dict/words | grep Nav | wc -l
13
```

En este caso lo que se hace es el **conteo de palabras** (word count = wc) que contienen "nav". Son trece.

Permisos

Cada usuario de un sistema Unix puede por default leer y modificar sus propios archivos en \$HOME; además, algunos archivos que descienden de la raíz ("/"), por ejemplo /usr/share/dict, pueden ser leídos por todos los usuarios, pero no modificados. Es por eso que los comandos anteriores funcionan sin problemas.

Hay tres tipos de permisos: de lectura, de escritura y de ejecución que pueden aplicarse tanto a archivos como a directorios.. Además, los usuarios de Unix pueden estar incluidos en uno o más grupos de usuarios como veremos en otra práctica. Es posible otorgar o modificar permisos al usuario dueño del archivo; a los usuarios de algún grupo al que pertenece el usuario; o a “otros usuarios” - todos los demás. A esto se denomina UGO:

- Usuario (User)
- Grupo (Group)
- Otros (Others)

Una forma sencilla de verificar estos permisos es utilizando el comando “ls -l”, como se muestra en el siguiente ejemplo:

```
usuario@Localhost:~$ ls -l /etc/shells (Renglón 1)
```

```
-rw-r--r-- 1 root root 73 Apr 25 2012 /etc/shells (Renglón 2)
```

El primer caracter del resultado (renglón 2) muestra un “-”, denotando que /etc/shells es un archivo. Si fuera un directorio, desplegaría “d”), y si fuera una liga a otro archivo, diría (“l”). Inmediatamente vienen los controles de UGO, indicando primero que el **usuario** dueño de ese archivo puede leerlo (r) o escribirlo (w) pero no ejecutarlo (-), quedando como “rw-”. Sigue después “r--” (solo lectura) para los usuarios del grupo, y también “r--” para cualquier otro usuario. “root” es el nombre del usuario root, y el segundo root es el grupo al que pertenece. Viene luego la fecha de creación del archivo https://sites.google.com/s/1qwGpfwYgpiuigPGYkSUBihuuWedfOzDN/p/0B26c--_scDIGWno5Nk9LMllrbk0/e y su tamaño. Viendo estos permisos vemos que todos los usuarios pueden leer ese archivo, pero solo el usuario root lo puede modificar y no le es posible ejecutarlo.

Para modificar estos permisos el primer requisito es que sea el usuario dueño quien lo haga (o en su defecto el usuario Root) mediante el comando “chmod”. Siguen algunos ejemplos de su uso:

```
usuario@localhost:~$ chmod u+r Archivo1 [al usuario (u) dueño del Archivo1 agregar (+) permisos de lectura (r)]
```

```
usuario@localhost:~$ chmod u-r, o+r Archivo1 [quitale (-) ese permiso y otórgaselo a
```

cualquier otro usuario (o)]

Shell scripting

Un shell script es un archivo de texto que contiene varios comandos que se ejecutan uno tras otro como si fuera un solo comando. Por ejemplo, el siguiente script, almacenado en un archivo de texto que se podría llamar cabezaCola, lista las primeras 5 líneas de Arch1 y las últimas 5:

```
#!/bin/bash    (estamos indicando que el script se ejecuta con el shell bash)
head -n 5 Arch1
tail -n 5 Arch1
```

Para poder ejecutar esas tres líneas como si fuera un solo comando, primero tenemos que dar permiso para que cabezaCola sea ejecutable:

```
$ chmod u+x cabezaCola
Y enseguida podríamos ejecutarlo:
$ ./cabezaCola
```

(El ./ dice: en este directorio donde estoy (.), busca el script cabezaCola y ejecútalo)

ACTIVIDAD 2: ¡Pruebalo! utiliza un script parecido para listar las primeras 5 y últimas 5 líneas de algún archivo que tengas en tu \$HOME.

OJO - la sintaxis de Bash es algo delicada, por ejemplo en el uso de espacios (a veces TIENEN que ir y a veces NO DEBEN ir) o de los diversos tipos de brackets: (, ((), {, [o diversas comillas: “, ‘, ` , y otros caracteres que a veces tienen significados especiales para Bash y otras veces no... : \$, *, \... ten mucho cuidado en el siguiente texto cada vez que dice **#ojo** ... Para darte una idea, 5+3 no funciona pero 5 + 3 sí; por otro lado, VAR = 1 no jala pero VAR=1, ¡sí!!... ¡excepto si la VAR es la que usas en un for loop! (entonces sí lleva espacios...)

Actividad 3 : variables del sistema y del usuario

Dentro de un script es posible utilizar variables, como en cualquier lenguaje de programación. Hay algunas variables que tienen valores predefinidos y se llaman “variables del sistema”. Ya conoces HOME, cuyo valor es el “path” de tu propio directorio donde guardas tus archivos.

1. Utiliza el comando **env | more** para conocer las variables del sistema y sus valores

Las **variables del usuario** son variables que el usuario define utilizando la siguiente sintaxis: NOMBRE_VARIABLE=<valor>. **#ojo: sin espacios rodeando el “=”**

El nombre de la variable es un carácter alfabético, seguido por caracteres alfanuméricos o el carácter “_”. Por ejemplo:

```
user@localhost$ MYVAR=10
```

El valor de una variable puede desplegarse utilizando el comando **echo**, que evalúa y despliega lo que le indiques. Por ejemplo, para conocer cuál es el valor almacenado en la variable “MYVAR”:

```
user@localhost$ echo $MYVAR.
```

El comando env no despliega las variables de usuario.

Las variables de usuario desaparecen al terminar la sesión.

Procesamiento de comandos

La sintaxis de un comando en Unix es como sigue:

Comando *[argumentos opcionales]* <argumentos obligatorios>

Los argumentos pueden ser opciones o archivos separados por espacios. Cada opción va precedida del carácter “-” y sirve para modificar de diversas maneras la ejecución del comando.

A un script del shell, es posible pasarle argumentos, justo como lo hacemos con cualquier otro comando. Ve la siguiente modificación a cabezaCola:

```
#!/bin/bash
head -n $1 $2
tail -n $1 $2
```

```
$/cabezaCola 3 Arch1
```

Le estamos pasando a cabezaCola el valor 3, que el script recibe como el primer argumento (\$1). El segundo argumento (\$2) es el nombre del archivo del cual nos interesan sus primeras y últimas líneas.

Cada uno de los elementos de la línea de comando se almacena en una variable predefinida, como se presenta a continuación:

\$@ junta todos los argumentos en un string

\$0 es el nombre del comando,

\$1 es el primer argumento,

\$2 es el segundo argumento, etc.

\$# Es el número de argumentos

El programa shell puede tomar estas variables para procesarlas.

Actividad 4 :

Comprueba el funcionamiento del nuevo cabezaCola.

==

El siguiente código muestra el manejo de los argumentos de un script (args.sh) en dos formas distintas. De pasada ilustramos cómo hacer ciclos en un script:

```
#!/bin/bash
# la primera línea con un '!' como primer caracter
# del comentario indica qué intérprete se utilizará
# en este caso, bash ubicado en /bin/bash
# Este programa muestra como procesar, uno por uno, los
# argumentos que se dan como entrada a un script.

# <-- este es el simbolo usado para definir un comentario
echo numero de argumentos: $#

# La variable $# contiene la cantidad de argumentos que
# se le dieron al programa (argc)

# respaldamos la lista de los argumentos
ARGUMENTOS=$@

# La variable $@ contiene la lista de argumentos (argv)
# Esta es la primera forma de hacerlo:

for arg in $@ #esto te muestra como iterar por las palabras de un string,
              #palabras separadas por espacios
do
    echo "argumento: $arg"
done

echo ""

# Esta es la segunda forma de hacerlo:
ARGNUM=$#    # numero de argumentos

for ((VAR = 1 ; VAR <= $ARGNUM ; VAR++)) #ojo: dos paréntesis (())
                                          #ojo: cuidado con espacios
do
    echo "argumento (${VAR}) = $1"
    shift
    # hace un corrimiento de la lista de argumentos: $@ ,
    # es decir, elimina el primer argumento.
    # Por ejemplo:
    #     si     $@ contiene "1 2 3 4"   entonces  $1 vale: 1
    #     después de ejecutar el comando: shift
    #     $@ es igual a "2 3 4"   y $1 vale: 2
done
```

La sintaxis de los ciclos se verá más adelante.

Comillas del Shell

Como vemos en los ejemplos anteriores, hay ciertos caracteres que tienen un significado especial para el Shell, como por ejemplo “\$”, “#”, “*” y otros. Hay formas de hacer que el shell ignore que son caracteres

especiales; esto sucede si un caracter o tira de caracteres se encierra en comillas sencillas ('). Por otro lado si una tira de caracteres se encierra en comillas inversas (``), la tira se interpreta como un comando a ejecutarse. Y si se encierra en comillas dobles, se interpreta como un string común y corriente... excepto si incluye los caracteres \$ o \ que conservan su significado para el shell... esto debe aclararse con los siguientes ejemplos; pero es un error común el confundir cuál tipo de comillas debe utilizarse.

El shell tiene definido tres tipos de comillas para ser usados con comandos o variables o strings:

- **(“) comilla doble:** Los caracteres delimitados por la comilla doble pierden su significado del carácter, excepto los caracteres “\$” y “\”. El backslash puede utilizarse para indicar “hey, esto que sigue es verdaderamente el caracter indicado”. Por ejemplo * indica que el asterisco es realmente un asterisco... pronto sabrás por qué hay que decirle eso al shell...
- **(‘) comilla sencilla:** Los caracteres o strings limitados por comilla simple pierden su significado para el shell. Recuerda: \\$ es realmente un signo de pesos, no una indicación para obtener el valor de una variable o expresión. Pruébalo:
MYVAR=10
echo \$MYVAR
echo \ \$MYVAR
echo ‘\$MYVAR’
- **(`) comilla inversa:** Los strings en comilla inversa forman un comando para ejecutar.

Ejemplos:

```
user@localhost$ a=5
user@localhost$ echo “$a”
5
user@localhost$> echo ‘$a’
$a
user@localhost$ echo `expr $a + 3`
8
```

Actividad 5 :

Investiga qué hace el comando “date”, y explica con sus propias palabras qué haría cada uno de los siguientes comandos. DESPUES ejecútalos para verificar tu entendimiento:

```
$ date
$ fecha=date
$ echo hoy es: “$fecha”
$ echo hoy es: ‘$fecha’
$ echo hoy es: ` $fecha `
```

Meta caracteres:

Los meta caracteres son comodines que se pueden utilizar al momento de referirse a algún nombre o dato y permiten elegir variaciones de cierto patrón; estos pueden ser: *, ?, [abc]* y se manejan como si

fuesen expresiones regulares.

Actividad 6:

Prueba los siguientes comandos e indica con tus propias palabras qué hace cada uno:

```
$ mkdir -p {meet,meat,beat,feet,mean,seat}
$ ls m*
$ ls me?t
$ ls [sm]eat
```

Aritmética del Shell

expr es un comando que evalúa las expresiones que recibe como argumentos. La sintaxis para realizar operaciones aritméticas son: **expr <op1> <operador_matematico> <op2>**

Ejemplo:

Suma:

```
user@localhost home $ expr 1 + 3 #ojo: esta vez con espacios rodeando a los operadores...
```

Resta:

```
user@localhost home $ expr 2 - 1
```

División:

```
user@localhost home $ expr 11 / 2 #ojo: ¿trunca?
```

Módulo:

```
user@localhost home $ expr 20 % 3
```

Para hacer operaciones con números flotantes, tenemos que utilizar el comando **bc** que es el intérprete de un lenguaje de aritmética con precisión arbitraria.

```
user@localhost home $ a=8.32 ; b=5.64
user@localhost home $ c=`echo $a + $b | bc`#ojo: comillas inversas para ejecutar comando echo
user@localhost home $ echo "$a + $b = $c"
```

Actividad 7 :

Si intentas utilizar la multiplicación directamente en consola notarás que marca un error.

```
ccna_admin@ccnalab12:~$ expr 7 * 3
expr: syntax error
```

La razón es sencilla, el símbolo de multiplicación coincide con un símbolo de meta-caracter, por lo tanto primero se tiene que hacer que el shell no lo vea como tal, ¿Que sintaxis utilizarás para lograr la multiplicación?

Comandos para Control de Flujo

Además de variables y comandos, cuando estamos utilizando “shell scripting” podemos hacer uso de ciertos estatutos para lograr un control de flujo. Estos son similares a los lenguaje de programación pero

tienen una sintaxis más estricta.

La sintaxis del comando **if** es la siguiente:

```
if [ condition ]; then  #ojo -espacios nuevamente... [condicion] no funciona...
    comando1
    ...
else
    comando2
    ...
fi
```

Ejemplos de expresiones o condiciones:

```
5 > 2  #ojo: espacios rodean a los operadores
3 + 6
$a > 3
```

Podemos ver un ejemplo del uso de este estatuto en un archivo que denominaremos **signo.sh** el cual contiene lo siguiente

```
#!/bin/bash
#
# Script que dado un número como argumento
# nos dice si el número es positivo o negativo
#
if [ $# -eq 0 ]
then
    echo "Uso: comando argumento"
    exit 1
fi

if [ $1 -lt 0 ]
then
    echo "$1 es negativo"
else
    echo "$1 es positivo"
fi
```

Habrás notado que utilizamos “**-lt**” (less than) en vez de “**<**” y esto se debe a que el primero es un comparador aritmético y el segundo léxico. Como quiera la siguiente tabla da un indicador de los tipos de condiciones que podemos usar:

Comparaciones (\$VAR1 OP \$VAR2) :

-eq	equal to
-ne	not equal to
-lt	less than
-le	less than or equal to
-gt	greater than

-ge greater than or equal to

Operaciones de archivos (OP \$FILE):

- s** file exists and is not empty
- f** file exists and is not a directory
- d** directory exists
- x** file is executable
- w** file is writable
- r** file is readable

Puedes consultar otras condiciones en la siguiente liga:

http://tldp.org/LDP/Bash-Beginners-Guide/html/sect_07_01.html

La sintaxis de comando **if** multinivel:

```
if [ condition ]; then
    comando1_then
    ...
elif [ condition2 ]; then
    comando1_elif
    ...
elif [ condition3 ]; then
    comando1_elif_2
    ...
else
    comando1_else
    ...
fi
```

Comando de Repetición

La sintaxis del **for** que en cada iteración le asigna a la variable **var** el siguiente elemento de la lista:

```
for var in list #donde cada elemento de la lista está
                #separado del siguiente por blancos
do
    comando1
    ...
done
```

y la sintaxis del **for** que repite los comandos hasta que la **expr2** es verdadera:

```
for (( expr1; expr2; expr3 )) #dos parentesis...
do
    comando1
    ...
done
```

Ejemplos: Suponiendo que un archivo **dobles.sh** almacena los siguientes comandos:

```
#!/bin/bash
#
#Script que imprime el doble de cada numero entre 1 y 5.
#
for i in 1 2 3 4 5
do
    echo "v=`expr $((2 * $i))`"
done
```

esa línea dice: evalúa la expresión `2 * $i`; después obtén el valor de esa evaluación... `$(())` y ejecuta el comando `expr` (comillas invertidas). La sintaxis del shell es algo complicada...

Otra forma de realizar el programa es como sigue en el script **doble2.sh**:

```
#!/bin/bash
#
#Script que imprime el doble de cada numero entre 1 y 5.
#
for (( i = 0 ; i <= 5; i++ )) #ojo esta vez i=0 sin espacios nope..
do
echo "Welcome $((2 * $i)) times"
done
```

La sintaxis del comando **while** es como sigue

```
while [ condicion ]
do
    comando
    ....
done
```

Ejemplo (test_while.sh):

```
#!/bin/sh
#
#Script to test while statement
#
#
n=$1
i=1
while [ $i -le 10 ]
do
    echo "v= `expr $((2 * $i))`"
    i=`expr $i + 1`
done
```

Se puen obtener más ejemplos a de esta fuente: <http://tldp.org/LDP/abs/html/arrays.html>

Últimas observaciones

Shell scripting corresponde al término “[Lenguaje de dominio-especifico](#)” y tiene el poder de ejecutar no sólo los comandos del S.O. sino también las aplicaciones escritas en cualquier otro lenguaje de programación.

Tomemos por ejemplo un caso de resultados de control de clima. Podemos alimentar los datos del clima a un programa de Matlab, que arrojaría pronósticos y estos los podemos introducir a un programa en Java que despliega estos formatos en servidores Web (mediante Servlets). Todo esto lo podemos

automatizar utilizando un simple “shell scripting” que se ejecute cada cierto tiempo; e incluso si se presenta un problema, el script podría tener líneas preparadas para enviar un correo (mediante los comandos propios del S.O. o desde cualquier programa hecho con cualquier lenguaje de programación que lo permita).

Investigación

Investiga **para qué sirven** los siguientes comandos ya que serán utilizados para la responder al formulario. No olvides revisar los argumentos que pueden manejar, ya que algunos comandos con ciertos argumentos pueden realizar las funciones base de otros.

1. join
2. paste
3. cut
4. tr
5. sed
6. uniq
7. sort
8. bc

TIP: Se espera que utilices los **man**(uales) de cada uno de ellos.

No olvides que también utilizaremos el comando **GREP**

Texto

Lo siguientes textos se utilizan en la práctica. Par que puedas hacerle copy paste a estos archivos, abre esta práctica desde un navegador que corra en Linux, por ejemplo el Firefox que viene con Ubuntu.

- **Primer ejercicio de texto**

Archivo1.txt

```
El miedo  
lado oscuro  
de la  
ira, del  
odio, del  
Sufrimiento .
```

Archivo2.txt

```
el camino hacia.  
El miedo lleva  
la ira lleva  
odio lleva al  
miedo en ti.https://www.youtube.com/watch?v=Sk5I2gz-Jvc
```

Lo siguiente es el texto que se utilizarán en los 3 últimos ejercicios de la parte de texto. Recuerde que loN debe quedar en ION y no cualquier otra letra “o” en otras palabras.

Archivo3.txt

DIME VENCEDOR RAPAZ
ESTE AMOROSO TORMENTO VERDE EMBELESO
COGIOME SIN PREVENCION

DETENTE SOMBRA

Detente, sombra de mi bien esquivo,
imagen del hechizo que más quiero,
bella ilusión por quien alegre muero,
dulce ficción por quien penosa vivo.

Si al imán de tus gracias, atractivo,
sirve mi pecho de obediente acero,
¿para qué me enamoras lisonjero
si has de burlarme luego fugitivo?

Mas blasonar no puedes, satisfecho,
de que triunfa de mí tu tiranía:
que aunque dejas burlado el lazo estrecho

que tu forma fantástica ceñía,
poco importa burlar brazos y pecho
si te labra prisión mi fantasía.

IoN

REDONDILLAS

Hombres necios que acusáis
a la mujer, sin razón,
sin ver que sois la ocasión
de lo mismo que culpáis;

si con ansia sin igual
solicitáis su desdén,
por qué queréis que obren bien
si las incitáis al mal?

Combatís su resistencia
y luego, con gravedad,
decís que fue liviandad
lo que hizo la diligencia.

Parecer quiere el denuedo
de vuestro parecer loco,
al niño que pone el coco
y luego le tiene miedo.

Queréis, con presunción necia,
hallar a la que buscáis
para pretendida, Thais,
y en la posesión, Lucrecia.

¿Qué humor puede ser más raro
que el que, falto de consejo,
él mismo empaña el espejo

y siente que no esté claro?

Con el favor y el desdén
tenéis condición igual,
quejándoos, si os tratan mal,
burlándoos, si os quieren bien.

Opinión, ninguna gana,
pues la que más se recata,
si no os admite, es ingrata,
y si os admite, es liviana.

Siempre tan necios andáis
que, con desigual nivel,
a una culpáis por cruel
y a otra por fácil culpáis.

¿Pues como ha de estar templada
la que vuestro amor pretende?,
¿si la que es ingrata ofende,
y la que es fácil enfada?

Mas, entre el enfado y la pena
que vuestro gusto refiere,
bien haya la que no os quiere
y quejaos en hora buena.

Dan vuestras amantes penas
a sus libertades alas,
y después de hacerlas malas
las queréis hallar muy buenas.

¿Cuál mayor culpa ha tenido
en una pasión errada:
la que cae de rogada,
o el que ruega de caído?

¿O cuál es de más culpar,
aunque cualquiera mal haga;
la que peca por la paga
o el que paga por pecar?

¿Pues, para qué os espantáis
de la culpa que tenéis?
Queredlas cual las hacéis
o hacedlas cual las buscáis.

Dejad de solicitar,
y después, con más razón,
acusaréis la afición
de la que os fuere a rogar.

Bien con muchas armas fundo
que lidia vuestra arrogancia,
pues en promesa e instancia
juntáis diablo, carne y mundo.

IoN

FINJAMOS QUE SOY FELIZ

Finjamos que soy feliz,
triste pensamiento, un rato;
quizá prodréis persuadirme,
aunque yo sé lo contrario,
que pues sólo en la aprehensión
dicen que estriban los daños,
si os imagináis dichoso
no seréis tan desdichado.

Sírvame el entendimiento
alguna vez de descanso,
y no siempre esté el ingenio
con el provecho encontrado.
Todo el mundo es opiniones
de pareceres tan varios,
que lo que el uno que es negro
el otro prueba que es blanco.

A unos sirve de atractivo
lo que otro concibe enfado;
y lo que éste por alivio,
aquél tiene por trabajo.

El que está triste, censura
al alegre de liviano;
y el que esta alegre se burla
de ver al triste penando.

Los dos filósofos griegos
bien esta verdad probaron:
pues lo que en el uno risa,
causaba en el otro llanto.

Célebre su oposición
ha sido por siglos tantos,
sin que cuál acertó, esté
hasta agora averiguado.

Antes, en sus dos banderas
el mundo todo alistado,
conforme el humor le dicta,
sigue cada cual el bando.

Uno dice que de risa
sólo es digno el mundo vario;
y otro, que sus infortunios
son sólo para llorados.

Para todo se halla prueba
y razón en qué fundarlo;
y no hay razón para nada,
de haber razón para tanto.

Todos son iguales jueces;
y siendo iguales y varios,
no hay quien pueda decidir

cuál es lo más acertado.

Pues, si no hay quien lo sentencie,
¿por qué pensáis, vos, errado,
que os cometió Dios a vos
la decisión de los casos?

O ¿por qué, contra vos mismo,
severamente inhumano,
entre lo amargo y lo dulce,
queréis elegir lo amargo?

Si es mío mi entendimiento,
¿por qué siempre he de encontrarlo
tan torpe para el alivio,
tan agudo para el daño?

El discurso es un acero
que sirve para ambos cabos:
de dar muerte, por la punta,
por el pomo, de resguardo.

Si vos, sabiendo el peligro
queréis por la punta usarlo,
¿qué culpa tiene el acero
del mal uso de la mano?

No es saber, saber hacer
discursos sutiles, vanos;
que el saber consiste sólo
en elegir lo más sano.

Especular las desdichas
y examinar los presagios,
sólo sirve de que el mal
crezca con anticiparlo.

En los trabajos futuros,
la atención, sutilizando,
más formidable que el riesgo
suele fingir el amago.

Qué feliz es la ignorancia
del que, indoctamente sabio,
halla de lo que padece,
en lo que ignora, sagrado!

No siempre suben seguros
vuelos del ingenio osados,
que buscan trono en el fuego
y hallan sepulcro en el llanto.

También es vicio el saber,
que si no se va atajando,
cuando menos se conoce
es más nocivo el estrago;
y si el vuelo no le abaten,
en sutilezas cebado,
por cuidar de lo curioso

olvida lo necesario.

Si culta mano no impide
crecer al árbol copado,
quita la sustancia al fruto
la locura de los ramos.

Si andar a nave ligera
no estorba lastre pesado,
sirve el vuelo de que sea
el precipicio más alto.

En amenidad inútil,
¿qué importa al florido campo,
si no halla fruto el otoño,
que ostente flores el mayo?

¿De qué sirve al ingenio
el producir muchos partos,
si a la multitud se sigue
el malogro de abortarlos?

Y a esta desdicha por fuerza
ha de seguirse el fracaso
de quedar el que produce,
si no muerto, lastimado.

El ingenio es como el fuego,
que, con la materia ingrato,
tanto la consume más
cuando él se ostenta más claro.

Es de su propio Señor
tan rebelado vasallo,
que convierte en sus ofensas
las armas de su resguardo.

Este pésimo ejercicio,
este duro afán pesado,
a los ojos de los hombres
dio Dios para ejercitarlos.

¿Qué loca ambición nos lleva
de nosotros olvidados?
Si es para vivir tan poco,
¿de qué sirve saber tanto?
¡Oh, si como hay de saber,
hubiera algún seminario
o escuela donde a ignorar
se enseñaran los trabajos!

¡Qué felizmente viviera
el que, flojamente cauto,
burlara las amenazas
del influjo de los astros!

Aprendamos a ignorar,
pensamiento, pues hallamos
que cuanto añadido al discurso,

tanto le usurpo a los años.

IoN

PUES ESTOY CONDENADA

Pues estoy condenada,
Fabio, a la muerte, por decreto tuyo,
y la sentencia airada
ni la apelo, resisto ni la huyo,
óyeme, que no hay reo tan culpado
a quien el confesar le sea negado.

Porque te han informado,
dices, de que mi pecho te ha ofendido,
me has, fiero, condenado.
¿Y pueden, en tu pecho endurecido
más la noticia incierta, que no es ciencia,
que de tantas verdades la experiencia?

Si a otros crédito has dado,
Fabio, ¿por qué a tus ojos se lo niegas,
y el sentido trocado
de la ley, al cordel mi cuello entregas,
pues liberal me amplías los rigores
y avaro me restringes los favores?

Si a otros ojos he visto,
mátenme, Fabio, tus airados ojos;
si a otro cariño asisto,
asistanme implacables tus enojos;
y si otro amor del tuyo me divierte,
tú, que has sido mi vida, me des muerte.

Si a otro, alegre, he mirado,
nunca alegre me mires ni te vea;
si le hablé con agrado,
eterno desagrado en tí posea;
y si otro amor inquieta mi sentido,
sáqueseme el alma tú, que mi alma has sido.

Mas, supuesto que muero,
sin resistir a mi infeliz suerte,
que me des sólo quiero
licencia de que escoja yo mi muerte;
deja la muerte a mi elección medida,
pues en la tuya pongo yo la vida.

IoN

ESTA TARDE MI BIEN

Esta tarde, mi bien, cuando te hablaba,
como en tu rostro y tus acciones vía
que con palabras no te persuadía,
que el corazón me vieses deseaba;

y Amor, que mis intentos ayudaba,
venció lo que imposible parecía:

pues entre el llanto, que el dolor vertía,
el corazón deshecho destilaba.

Baste ya de rigores, mi bien, baste:
no te atormenten más celos tiranos,
ni el vil recelo tu inquietud contraste

con sombras necias, con indicios vanos,
pues ya en líquido humor viste y tocaste
mi corazón deshecho entre tus manos.

IoN

ESTOS VERSOS LECTOR MIO

Estos versos, lector mío,
que a tu deleite consagro,
y sólo tienen de buenos
conocer yo que son malos,
ni disputártelos quiero,
ni quiero recomendarlos,
porque eso fuera querer
hacer de ellos mucho caso.

No agradecido te busco:
pues no debes, bien mirado,
estimar lo que yo nunca
juzgué que fuera a tus manos.
En tu libertad te pongo,
si quisieres censurarlos;
pues de que, al cabo, te estás
en ella, estoy muy al cabo.

No hay cosa más libre que
el entendimiento humano;
pues lo que Dios no violenta,
por qué yo he de violentarlo?

Di cuanto quisieres de ellos,
que, cuanto más inhumano
me los mordieres, entonces
me quedas más obligado,
pues le debes a mi musa
el más sazonado plato
(que es el murmurar), según
un adagio cortesano.

Y siempre te sirvo, pues,
o te agrado, o no te agrado:
si te agrado, te diviertes;
murmuras, si no te cuadro.

Bien pudiera yo decirte
por disculpa, que no ha dado
lugar para corregirlos
la priesa de los traslados;
que van de diversas letras,
y que algunos, de muchachos,
matan de suerte el sentido

que es cadáver el vocablo;
y que, cuando los he hecho,
ha sido en el corto espacio
que ferian al ocio las
precisiones de mi estado;
que tengo poca salud
y continuos embarazos,
tales, que aun diciendo esto,
llevo la pluma trotando.

Pero todo eso no sirve,
pues pensarás que me jacto
de que quizá fueran buenos
a haberlos hecho despacio;
y no quiero que tal creas,
sino sólo que es el darlos
a la luz, tan sólo por
obedecer un mandato.

Esto es, si gustas creerlo,
que sobre eso no me mato,
pues al cabo harás lo que
se te pusiere en los cascos.
Y adiós, que esto no es más de
darte la muestra del paño:
si no te agrada la pieza,
no desenvuelvas el fardo.

IoN

YA QUE PARA DESPEDIRME

Ya que para despedirme,
dulce idolatrado dueño,
ni me da licencia el llanto
ni me da lugar el tiempo,

háblente los tristes rasgos,
entre lastimosos ecos,
de mi triste pluma, nunca
con más justa causa negros.

Y aun ésta te hablará torpe
con las lágrimas que vierto,
porque va borrando el agua
lo que va dictando el fuego.

Hablar me impiden mis ojos;
y es que se anticipan ellos,
viendo lo que he de decirte,
a decírtelo primero.

Oye la elocuencia muda
que hay en mi dolor, sirviendo
los suspiros, de palabras,
las lágrimas, de conceptos.

Mira la fiera borrasca
que pasa en el mar del pecho,

donde zozobran, turbados,
mis confusos pensamientos.

Mira cómo ya el vivir
me sirve de afán grosero;
que se avergüenza la vida
de durarme tanto tiempo.

Mira la muerte, que esquiva
huye porque la deseo;
que aun la muerte, si es buscada,
se quiere subir de precio.

Mira cómo el cuerpo amante,
rendido a tanto tormento,
siendo en lo demás cadáver,
sólo en el sentir es cuerpo.

Mira cómo el alma misma
aun teme, en su ser exento,
que quiera el dolor violar
la inmunidad de lo eterno.

En lágrimas y suspiros
alma y corazón a un tiempo,
aquél se convierte en agua,
y ésta se resuelve en viento.

Ya no me sirve de vida
esta vida que poseo,
sino de condición sola
necesaria al sentimiento.

Mas, por qué gasto razones
en contar mi pena y dejo
de decir lo que es preciso,
por decir lo que estás viendo?

En fin, te vas, ay de mi!
Dudosamente lo pienso:
pues si es verdad, no estoy viva,
y si viva, no lo creo.

Posible es que ha de haber día
tan infausto, funesto,
en que sin ver yo las tuyas
esparza sus luces Febo?

Posible es que ha de llegar
el rigor a tan severo,
que no ha de darle tu vista
a mis pesares aliento?

Ay, mi bien, ay prenda mía,
dulce fin de mis deseos!
Por qué me llevas el alma,
dejándome el sentimiento?

Mira que es contradicción

que no cabe en un sujeto,
tanta muerte en una vida,
tanto dolor en un muerto.

Mas ya que es preciso, ay triste!,
en mi infeliz suceso,
ni vivir con la esperanza,
ni morir con el tormento,

dame algún consuelo tú
en el dolor que padezco;
y quien en el suyo muere,
viva siquiera en tu pecho.

No te olvides que te adoro,
y sírvante de recuerdo
las finezas que me debes,
si no las prendas que tengo.

Acuérdate que mi amor,
haciendo gala de riesgo,
sólo por atropellarlo
se alegraba de tenerlo.

Y si mi amor no es bastante,
el tuyo mismo te acuerdo,
que no es poco empeño haber
empezado ya en empeño.

Acuérdate, señor mío,
de tus nobles juramentos;
y lo que juró la boca
no lo desmientan tus hechos.

Y perdona si en temer
mi agravio, mi bien, te ofendo,
que no es dolor, el dolor
que se contiene atento.

Y adiós; que con el ahogo
que me embarga los alientos,
ni sé ya lo que te digo
ni lo que te escribo leo.

IoN

DIME VENCEDOR RAPAZ

Dime vencedor Rapaz,
vencido de mi constancia,
¿Qué ha sacado tu arrogancia
de alterar mi firme paz?
Que aunque de vencer capaz
es la punta de tu arpón,
¿qué importa el tiro violento,
si a pesar del vencimiento
queda viva la razón?

Tienes grande señorío;

pero tu jurisdicción
domina la inclinación,
mas no pasa el albedrío.
Y así librarme confío
de tu loco atrevimiento,
pues aunque rendida siento
y presa la libertad,
se rinde la voluntad
pero no el consentimiento.

En dos partes dividida
tengo el alma en confusión:
una, esclava a la pasión,
y otra, a la razón medida.
Guerra civil, encendida,
aflige el pecho importuna:
quiere vencer cada una,
y entre fortunas tan varias,
morirán ambas contrarias
pero vencerá ninguna.

Cuando fuera, Amor, te vía,
no merecí de ti palma;
y hoy, que estás dentro del alma,
es resistir valentía.
Córrase, pues, tu porfía,
de los triunfos que te gano:
pues cuando ocupas, tirano,
el alma, sin resistillo,
tienes vencido el Castillo
e invencible el Castellano.

Invicta razón alienta
armas contra tu vil saña,
y el pecho es corta campaña
a batalla tan sangrienta.
Y así, Amor, en vano intenta
tu esfuerzo loco ofenderme:
pues podré decir, al verme
expirar sin entregarme,
que conseguiste matarme
mas no pudiste vencerme.

IoN

COGIOME SIN PREVENCION

Cogióme sin prevención
Amor, astuto y tirano:
con capa de cortesano
se me entró en el corazón.
Descuidada la razón
y sin armas los sentidos,
dieron puerta inadvertidos;
y él, por lograr sus enojos,
mientras suspendió los ojos
me salteó los oídos.

Disfrazado entró y mañoso;
mas ya que dentro se vio
del Paladión, salió
de aquel disfraz engañoso;
y, con ánimo furioso,
tomando las armas luego,
se descubrió astuto Griego
que, iras brotando y furores,
matando los defensores,
puso a toda el Alma fuego.

Y buscando sus violencias
en ella al príamo fuerte,
dio al Entendimiento muerte,
que era Rey de las potencias;
y sin hacer diferencias
de real o plebeya grey,
haciendo general ley
murieron a sus puñales
los discursos racionales
porque eran hijos del Rey.

A Casandra su fiereza
buscó, y con modos tiranos,
ató a la Razón las manos,
que era del Alma princesa.
En prisiones su belleza
de soldados atrevidos,
lamenta los no creídos
desastres que adivinó,
pues por más voces que dio
no la oyeron los sentidos.

Todo el palacio abrasado
se ve, todo destruido;
Deifobo allí mal herido,
aquí Paris maltratado.
Prende también su cuidado
la modestia en Polixena;
y en medio de tanta pena,
tanta muerte y confusión,
a la ilícita afición
sólo reserva en Elena.

Ya la Ciudad, que vecina
fue al Cielo, con tanto arder,
sólo guarda de su ser
vestigios, en su ruina.
Todo el amor lo extermina;
y con ardiente furor,
sólo se oye, entre el rumor
con que su crueldad apoya:
"Aquí yace un Alma Troya
¡Victoria por el Amor!"

IoN

ESTE AMOROSO TORMENTO

Este amoroso tormento
que en mi corazón se ve,
se que lo siento y no se
la causa porque lo siento

Siento una grave agonía
por lograr un devaneo,
que empieza como deseo
y para en melancolía.

y cuando con mas terneza
mi infeliz estado lloro
se que estoy triste e ignoro
la causa de mi tristeza. "

Siento un anhelo tirano
por la ocasión a que aspiro,
y cuando cerca la miro
yo misma aparto la mano.
Porque si acaso se ofrece,
después de tanto desvelo
la desazona el recelo
o el susto la desvanece.

Y si alguna vez sin susto
consigo tal posesión
(cualquiera) leve ocasión
me malogra todo el gusto.

Siento mal del mismo bien
con receloso temor
y me obliga el mismo amor
tal vez a mostrar desdén.

IoN

VERDE EMBELESO

Verde embeleso de la vida humana,
loca esperanza, frenesí dorado,
sueño de los despiertos intrincado,
como de sueños, de tesoros vana;

alma del mundo, senectud lozana,
decrépito verdor imaginado;
el hoy de los dichosos esperado,
y de los desdichados el mañana:

sigan tu sombra en busca de tu día
los que, con verdes vidrios por anteojos,
todo lo ven pintado a su deseo;

que yo, más cuerda en la fortuna mía,
tengo en entrambas manos ambos ojos
y solamente lo que toco veo.

Códigos ejemplo

Despliegue de correos oficiales a partir de matrículas

```
#!/bin/bash
```

```
# Version: 2 ( 31/08/15 )
```

```
# Objetivo: Enseñar la forma de leer archivos y manejar strings
```

```
# Entrada por stdin. Cada línea puede ser:
```

```
# - una matrícula, con o sin la "A". Las matriculas siempre son de 8 dígitos
```

```
# - una dirección de correo ITESM válida, que siempre mide 18 caracteres.
```

```
# se verifica cuidadosamente que el input sea correcto; por cada línea incorrecta
```

```
# (por ejemplo una matrícula de 50 dígitos... ) se genera un mensaje de error.
```

```
# Salida por stdout: por cada línea de entrada, se genera la dirección de correo correspondiente
```

```
# en caso de que la entrada sea ya una dirección de correo, simplemente se copia a la salida.
```

```
# Expresiones regulares utilizadas para identificar los posibles formatos:
```

```
MATRCONA="^[0-9]*" # matricula con A...
```

```
MATRSINA="^[1-9]*" # matricula sin la A...
```

```
DIRCORREO="^[0-9]*@itesm.mx" #dirección de correo
```

```
HazLaChamba() { #ojo: declaración de función en Bash
```

```
# El code pattern "while read line do (...) done </dev/stdin"
```

```
# reads lines from stdin$
```

```
while read line
```

```
do
```

```
    WORD=$line
```

```
    # echo "WORD ES" $WORD
```

```
    # Aqui es donde buscaremos WORD, en los posibles formatos de entrada:
```

```
    # Veamos primero si es matricula con A:
```

```
    Match=`echo "$WORD" | grep -xwi "$MATRCONA"` #ojo con comillas invertidas$
```

```
    # echo "match"$Match
```

```

if [[ "$Match" != "" && "${#Match}" -eq 9 ]] #ojo checamos la longitud
then
    # echo "Es matricula con A "
    echo "$Match""@itesm.mx" #>> /dev/stdout
else
    # Puede estar ya bien en el formato definitivo
    Match=`echo "$WORD" | grep -xwi "$DIRCORREO"`
    if [[ "$Match" != "" && "${#Match}" ]]
    then
        # echo "Un correo hallado" "$DIRCORREO" ": " $Match
        echo "$Match" #>> /dev/stdout
    else
        #Puede ser meramente número (Sin A0)
        Match=`echo "$WORD" | grep -xwi "$MATRSINA"`
        if [[ "$Match" != "" && "${#Match}" -eq 8 ]]
        then
            # echo es una matrícula sin A.
            echo "A""$Match""@itesm.mx" #>> /dev/stdout
        else
            echo "formato o longitud erróneas" "$WORD"
        fi
    fi
fi
done </dev/stdin #end while
}

```

```

#####
##### Equivalente a un MAIN #####
#####

```

```

# echo "BASH para convertir matriculas a cuentas de @itesm.mx"

```

```

# Validamos argumentos de entrada

```

```

if [ $# -neq 0 ]
then

```

```
    echo "sobran argumentos. El script lee de stdin y su salida es stdout"
    exit 1
else
    HazLaChamba #invoca la función. ojo; una función sin argumentos se invoca sin paréntesis
    exit 0
fi
```