

```
1 =begin
2 Ejercicio 1a. Desarrollar un script donde se muestren los ficheros del pasado por
3 parámetro, así como los permisos asociados a cada uno de ellos, su identificador de
4 usuario, su
5 identificador de grupo, así como permisos como "Setuid, Setgid, o hasta incluso si
6 tiene activo
7 el bit sticky bit".
8 Sintaxis: listado.rb [Dir]
9 Dir será opcional de forma que si el usuario no lo inserta el script buscará en el
10 directorio actual
11 El script deberá capturar todas las posibles excepciones como pueden ser:
12 a. Debe permitir el paso de parámetros con una sintaxis correcta
13 b. Si se produce alguna excepción grave a la hora de la búsqueda en sí, el script
14 deberá
15 comunicarlo.
16 =end
17
18
19
20
21
22
23 =begin
24 Usage: Check if a path exists
25 Name of method: checkDirectory
26 Date of creation: 08/04/2021
27 Members: Roberto Jiménez y Alberto Pérez
28 Last modification: 08/04/2021
29 Parameters:
30   Entry:
31     - path: path that will be checked
32   Out:
33     - boolean: True if exists
34 =end
35 def checkDirectory path
36   return Dir.exist?(path)
37 end
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52 =begin
53 Usage: List files of a directory
54 Name of method: listFiles
55 Date of creation: 08/04/2021
56 Members: Roberto Jiménez y Alberto Pérez
57 Last modification: 08/04/2021
58 Parameters:
59   Entry:
60     - path: path that will be checked
61   Out:
62     - List of files of path
63 =end
64 def listFiles path
65   return `ls -l #{path}`
66 end
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

```
57 Parameters:
58   Entry:
59     - Argv: Parameter introduced by console when program is called
60   Out:
61     - None
62 =end
63 if ARGV.length == 1
64   path = ARGV[0]
65   if checkDirectory path
66     puts listFiles path
67   else
68     puts "Error, path is not correct"
69   end
70 end
```

```
1 =begin
2 Ejercicio 1b. Modificar el anterior script para que se le pida al usuario por consola
  el directorio
3 donde tendrá que buscar.
4 Sintaxis: listado.rb
5 El script deberá capturar todas las posibles excepciones como pueden ser:
6 a. Si el directorio que el usuario introduce no existe o no es un directorio, deberá
  dar un
7 error.
8 b. Si el usuario que lo invoca no tiene permisos de lectura del directorio, el script
  deberá
9 gestionar esta excepción.
10 =end
11
12
13
14
15 =begin
16 Usage: Check if a path exists
17 Name of method: checkDirectory
18 Date of creation: 08/04/2021
19 Members: Roberto Jiménez y Alberto Pérez
20 Last modification: 08/04/2021
21 Parameters:
22   Entry:
23     - path: path that will be checked
24   Out:
25     - boolean: True if exists
26 =end
27 def checkDirectory path
28   return Dir.exist?(path)
29 end
30
31
32 =begin
33 Usage: List a directory passed by path
34 Name of method: listDirectory
35 Date of creation: 08/04/2021
36 Members: Roberto Jiménez y Alberto Pérez
37 Last modification: 08/04/2021
38 Parameters:
39   Entry:
40     - path: path of the directory to list
41   Out:
42     - list: result from ls if directory has reading permissions
43 =end
44 def listDirectory path
45   if File.readable?(path)
46     list = `ls #{path}`
47     return list.split(" ")
48   end
49   return ["Error, directory has not reading permissions"]
50 end
51
52
53 =begin
54 Usage: List all directories starting from path
55 Name of method: listDirectories
56 Date of creation: 08/04/2021
57 Members: Roberto Jiménez y Alberto Pérez
```

```
58 Last modification: 9/04/2021
59 - Added different colors in terminal for directories, files...
60 Parameters:
61   Entry:
62     - path: path that will be checked
63   Out:
64 =end
65 def listDirectories path
66   auxDirectories = []
67   files = listDirectory path
68   puts "\e[1;34m#{path}\e[m"
69   files.each do |element|
70     newPath = path + "/" + element
71     if checkDirectory newPath
72       puts "\e[1;36m#{newPath}\e[m"
73       auxDirectories.push(newPath)
74     else
75       puts "\e[1;37m#{element}\e[m"
76     end
77   end
78
79   puts "\e[1;37m\n\e[m"
80
81   auxDirectories.each do |directory|
82     listDirectories directory
83   end
84 end
85
86
87 =begin
88 Usage: Body of the program
89 Date of creation: 08/04/2021
90 Members: Roberto Jiménez y Alberto Pérez
91 Last modification: 08/04/2021
92 Parameters:
93   Entry:
94     - Path: Parameter requested from the user
95   Out:
96     - None
97 =end
98 puts "Put the directory path to list"
99 path = gets.chomp
100
101 if checkDirectory path
102   listDirectories path
103 else
104   puts "Error, path is not correct"
105 end
```

```
1 =begin
2 Ejercicio 3. Escribir un script "renombrar" que renombre un conjunto de ficheros,
  posibilitando
3 el cambio de la extensión del fichero, siendo la sintaxis la siguiente:
4 ./renombrar extensión-anterior nueva-extensión
5 El script deberá contemplar los posibles casos de error, como por ejemplo el paso de
  parámetros
6 correcto
7 =end
8
9 =begin
10 Usage: Lists the files from a directory that match with extension
11 Name of method: listFiles
12 Date of creation: 29/04/2021
13 Members: Roberto Jiménez y Alberto Pérez
14 Last modification: 29/04/2021
15 Parameters:
16   Entry:
17     - extension: extension to check
18   Out:
19     - List of files from path that matched the extension
20 =end
21 def listFiles extension
22   list = []
23   list = `ls *.{extension} | nl`
24   return list.split("\n")
25 end
26
27 =begin
28 Usage: Renames a file with a new extension
29 Name of method: renameFile
30 Date of creation: 29/04/2021
31 Members: Roberto Jiménez y Alberto Pérez
32 Last modification: 29/04/2021
33 Parameters:
34   Entry:
35     - fileName: name of the file whose extension is going to be changed
36     - new_extension: new extension to the file
37   Out:
38     - String: Name of the file with the new extension
39 =end
40 def renameFile fileName, new_extension
41   auxName = fileName.split(".")
42   onlyName = auxName[0]
43   `mv #{fileName} #{onlyName}.#{new_extension} `
44   return "#{onlyName}.#{new_extension}"
45 end
46
47 =begin
48 Usage: Body of the program
49 Date of creation: 29/04/2021
50 Members: Roberto Jiménez y Alberto Pérez
51 Last modification: 29/04/2021
52 Parameters:
53   Entry:
54     - Argv: Parameter introduced by console when program is called
55   Out:
56     - None
57 =end
58 if ARGV.length == 2
```

```
59 current_extension = ARGV[0]
60 new_extension = ARGV[1]
61
62 files = listFiles current_extension
63 puts "Files available for rename: "
64 puts files
65 puts "Select the number of the file to rename: "
66 fileToRename = (STDIN.gets.chomp.to_i - 1)
67
68 if (fileToRename >= 0) && (fileToRename < files.length)
69   auxFile = files[fileToRename].split(" ")
70   renamedFile = renameFile auxFile[1], new_extension
71   puts "#{auxFile[1]} renamed to: #{renamedFile}"
72 else
73   puts "Error, number of file invalid"
74 end
75 end
```

```
1 =begin
2 Ejercicio 4. Realizar un script denominado diffd con la siguiente sintaxis:
3 diffd [-r] directorio1 [directorio2]
4 Donde el script deberá buscar todos los nombres de fichero ubicados en directorio1 y
5 directorio2, exceptuando los nombres de los directorios, y a continuación mostrar las
6 diferencias, o sea aquellos ficheros que aparecen en uno de los directorios, pero no
7 en el otro,
8 indicando para cada uno de ellos en que directorio se encuentran.
9 Por otra parte, será necesario introducir una serie de parámetros: directorio1 y
10 directorio2,
11 siendo los directorios donde realizaremos la comparación, y como podemos observar en
12 la
13 sintaxis
14 directorio2 es opcional, esto quiere decir que si el usuario no lo especifica el
15 script comparará
16 directorio1 con el directorio actual.
17 Finalmente, si añadimos la opción -r (opcional) el script tendrá que realizar la
18 acción inversa,
19 mostrando los nombres de aquellos ficheros que se encuentran en los dos directorios.
20
21 =end
22
23 =begin
24 Usage: Check if a path exists
25 Name of method: checkDirectory
26 Date of creation: 09/04/2021
27 Members: Roberto Jiménez y Alberto Pérez
28 Last modification: 09/04/2021
29 Parameters:
30 Entry:
31 - path: path that will be checked
32 Out:
33 - boolean: True if exists
34 =end
35
36 def checkDirectory path
37   return Dir.exist?(path)
38 end
39
40
41 =begin
42 Usage: Get parameters from ARGV
43 Name of method: getParameters
44 Date of creation: 09/04/2021
45 Members: Roberto Jiménez y Alberto Pérez
46 Last modification: 09/04/2021
47 Parameters:
48 Entry:
49 - None
50 Out:
51 - Array with parameters passed
52 =end
53
54 def getParameters
55   array = []
56   i = 0
57   for arg in ARGV
58     array[i] = arg
59     i = i + 1
60   end
61   return array
62 end
```

```
56 end
57
58 =begin
59   Usage: List files of a directory
60   Name of method: listFiles
61   Date of creation: 08/04/2021
62   Members: Roberto Jiménez y Alberto Pérez
63   Last modification: 08/04/2021
64   Parameters:
65     Entry:
66       - path: path that will be checked
67     Out:
68       - List of files of path
69 =end
70 def listFiles path
71   list = []
72   list = `ls #{path}`
73   return list.split("\n")
74 end
75
76
77 =begin
78   Usage: Remove directory name from a list of files
79   Name of method: removeDirectory
80   Date of creation: 09/04/2021
81   Members: Roberto Jiménez y Alberto Pérez
82   Last modification: 09/04/2021
83   Parameters:
84     Entry:
85       - list: list that will be checked
86     Out:
87       - Same list without directory names
88 =end
89 def removeDirectory list, dir
90   newList = []
91
92   for i in list
93     path= dir + '/' + i
94     if !checkDirectory path
95       newList.push(i)
96     end
97   end
98   return newList
99 end
100
101 =begin
102   Usage: Compare 2 directories to see if exists
103   file with the same name and reverse
104   Name of method: diff
105   Date of creation: 09/04/2021
106   Members: Roberto Jiménez y Alberto Pérez
107   Last modification: 09/04/2021
108   Parameters:
109     Entry:
110       - dir1: first directory to check
111       - dir2: second directory to check
112       - reverse: boolean, to see if -r option is enabled
113     Out:
114       - None
115 =end
```



```
116 def diff dir1, dir2, reverse
117
118   filesDir1 = removeDirectory (listFiles dir1), dir1
119   filesDir2 = removeDirectory (listFiles dir2), dir2
120
121   for i in filesDir1
122     if filesDir2.include?(i) and reverse
123       puts i + ' is in both directories'
124     elsif !filesDir2.include?(i) and !reverse
125       puts i + ' is only in ' + dir1
126     end
127   end
128   for i in filesDir2
129     if !filesDir1.include?(i) and !reverse
130       puts i + ' is only in ' + dir2
131     end
132   end
133 end
134
135
136 list = getParameters
137 actualDir = `pwd`
138
139 case list[0]
140 when "-r"
141   if list.length == 3 and checkDirectory list[1] and checkDirectory list[2]
142     diff list[1], list[2], true
143
144   elsif list.length == 2 and checkDirectory list[1]
145     diff list[1], actualDir, true
146   else
147     puts 'Syntax error, correct syntax: ruby diffd_ej4.rb [-r] dir1 [dir2]'
148   end
149 else
150   case list.length
151   when 0
152     puts 'Error, it must be a directory into a parameters'
153     puts 'Syntax: ruby diffd_ej4.rb [-r] dir1 [dir2]'
154   when 1
155     if checkDirectory list[0]
156       diff list[0], actualDir, false
157
158     else
159       puts 'Error, ' + list[0] + ' is not a directory'
160     end
161   when 2
162     if checkDirectory list[0] and checkDirectory list[1]
163       diff list[0], list[1], false
164
165     elsif checkDirectory list[0]
166       puts 'Error, ' + list[1] + ' is not a directory'
167     elsif checkDirectory list[1]
168       puts 'Error, ' + list[0] + ' is not a directory'
169     end
170   else
171     puts 'Too many arguments'
172   end
173 end
174
175
```

```
1 =begin
2 Ejercicio 5. Desarrollar un script que nos muestre por consola información relevante
3 del
4 identificador de proceso que el usuario inserte por consola.
5 Sintaxis: proceso.rb
6 El script deberá mostrar la siguiente información del proceso: identificador UID,
7 propietario
8 PID, PPID, Prioridad absoluta "C" , Prioridad Relativa "PRI" y la Dirección en
9 memoria o en
10 disco del proceso "ADDR"
11 =end
12
13 Usage: Body of code program who shows info about a process id
14 Name of method:
15 Date of creation: 10/04/2021
16 Members: Roberto Jiménez y Alberto Pérez
17 Last modification: 10/04/2021
18 Parameters:
19   Entry:
20     - None, only a parameter passed by ARGV
21   Out:
22     - None, output is shown in terminal
23 =end
24 if ARGV.length == 0
25   puts 'Error, a process id is necessary'
26 elsif ARGV.length == 1
27   query = 'ps -o uid,pid,ppid,c,pri,addr ' + ARGV[0]
28   result = `#{query}`
29   puts result
30 else
31   puts 'Error, only process id parameter is necessary'
32 end
```

```
1 =begin
2 Ejercicio 6. Escribir un script en ruby donde el usuario pasará un conjunto de
  valores por
3 consola y automáticamente devolverá todos los valores sin ordenar por consola y
4 posteriormente ordenados. Su sintaxis será:
5 ./ordenar.rb 13 5 6 7 0 -6
6 El script deberá contemplar todos los casos de error existentes, como podría ser el
  paso de algún
7 parámetro de tipo no numérico. ¿qué sucedería si le pasamos al script más de 10
  parámetros?
8 Implementar las oportunas modificaciones para que lo soporte.
9 =end
10
11 =begin
12 Usage: Check if the object passed is a number
13 Name of method: is_number?
14 Date of creation: 15/04/2021
15 Members: Roberto Jiménez y Alberto Pérez
16 Last modification: 15/04/2021
17   - Now you can check all type of objects, not only Strings.
18 Parameters:
19   Entry:
20     - object: object to check
21   Out:
22     - boolean: return true if the object is a number
23 =end
24 def is_number? object
25   true if Float(object) rescue false
26 end
27
28
29 =begin
30 Usage: Deletes from argv all elements that aren't numbers
31 Name of method: listFiles
32 Date of creation: 15/04/2021
33 Members: Roberto Jiménez y Alberto Pérez
34 Last modification: 15/04/2021
35 Parameters:
36   Entry:
37     - argv: list of elements to check
38   Out:
39     - aux: argv, only formed by numbers
40 =end
41 def cleanArgv argv
42   aux = []
43   argv.each do |element|
44     if is_number? element
45       aux.push(element)
46     else
47       puts "Error, #{element} is not a number"
48     end
49   end
50   return aux
51 end
52
53
54 =begin
55 Usage: Generates a string from an array
56 Name of method: generateStringFromArray
57 Date of creation: 15/04/2021
```

```
58 Members: Roberto Jiménez y Alberto Pérez
59 Last modification: 15/04/2021
60 Parameters:
61   Entry:
62     - array: array of elements
63   Out:
64     - string: string got from the array
65 =end
66 def generateStringFromArray array
67   puts ""
68   string = ""
69   array.each do |n|
70     string += "#{n} "
71   end
72   return string
73 end
74
75
76 =begin
77 Usage: Body of the program
78 Date of creation: 15/04/2021
79 Members: Roberto Jiménez y Alberto Pérez
80 Last modification: 15/04/2021
81 Parameters:
82   Entry:
83     - Argv: Parameter introduced by console when program is called
84   Out:
85     - None
86 =end
87
88 if ARGV.length > 0
89   argvCleaned = cleanArgv(ARGV)
90
91   numbers = generateStringFromArray argvCleaned
92   sortedNumbers = generateStringFromArray argvCleaned.sort
93
94   puts "The numbers without sorting are: #{numbers}"
95   puts "The sorted numbers are: #{sortedNumbers}"
96 else
97   puts "Error, you have to pass parameters"
98 end
99
```

```
1 =begin
2 Ejercicio 7. Realizar un script en ruby denominado nchmod que nos visualice por el
3 código en
4 decimal asociado a los permisos que le queramos dar a un fichero.
5 Ejemplo:
6 nchmod.rb r-sr-xr-x el resultado que devolverá será 4555
7 El script tendrá que controlar el número de caracteres insertados por el usuario,
8 mostrando un
9 mensaje de error donde especifiquemos el uso
10 =end
11
12
13 =begin
14 Usage: doing nothing in a match
15 Name of method: pass
16 Date of creation: 22/04/2021
17 Members: Roberto Jiménez y Alberto Pérez
18 Last modification: 22/04/2021
19 Parameters:
20     Entry:
21         - None
22     Out:
23         - None
24 =end
25 def pass
26 end
27
28 =begin
29 Usage: Checks the firsts characters
30 Name of method: checkFirsts
31 Date of creation: 22/04/2021
32 Members: Roberto Jiménez y Alberto Pérez
33 Last modification: 22/04/2021
34 Parameters:
35     Entry:
36         - characters: string to check
37         - array: gets the number of permissions in octal
38     Out:
39         - None
40 =end
41 def checkFirsts characters, array
42
43     for i in [1,4,7]
44         if (i == 1) && (characters[i] == "r")
45             array[1] = array[1] + 4
46         elsif (i == 4) && (characters[i] == "r")
47             array[2] = array[2] + 4
48         elsif (i == 7) && (characters[i] == "r")
49             array[3] = array[3] + 4
50         elsif (characters[i] == "-")
51             pass
52         else
53             puts "Error, the letter #{characters[i]} does not belong to a permission"
54             exit
55         end
56     end
57 end
58
```

```
59
60
61 =begin
62 Usage: Checks the seconds characters
63 Name of method: checkSeconds
64 Date of creation: 22/04/2021
65 Members: Roberto Jiménez y Alberto Pérez
66 Last modification: 22/04/2021
67 Parameters:
68   Entry:
69     - characters: string to check
70     - array: gets the number of permissions in octal
71   Out:
72     - None
73 =end
74 def checkSeconds characters, array
75   for i in [2,5,8]
76     if (i == 2) && (characters[i] == "w")
77       array[1] = array[1] + 2
78     elsif (i == 5) && (characters[i] == "w")
79       array[2] = array[2] + 2
80     elsif (i == 8) && (characters[i] == "w")
81       array[3] = array[3] + 2
82     elsif (characters[i] == "-")
83       pass
84     else
85       puts "Error, the letter #{characters[i]} does not belong to a permission"
86       exit
87     end
88   end
89 end
90
91
92 =begin
93 Usage: Checks the seconds characters
94 Name of method: checkSeconds
95 Date of creation: 22/04/2021
96 Members: Roberto Jiménez y Alberto Pérez
97 Last modification: 22/04/2021
98 Parameters:
99   Entry:
100     - characters: string to check
101     - array: gets the number of permissions in octal
102   Out:
103     - None
104 =end
105 def checkThirds characters, array
106   for i in [3,6,9]
107     if (i == 3) && (characters[i] == "x")
108       array[1] = array[1] + 1
109     elsif (i == 3) && (characters[i] == "s")
110       array[0] = array[0] + 4
111     elsif (i == 6) && (characters[i] == "x")
112       array[2] = array[2] + 1
113     elsif (i == 6) && (characters[i] == "s")
114       array[0] = array[0] + 2
115     elsif (i == 9) && (characters[i] == "x")
116       array[3] = array[3] + 1
117     elsif (i == 9) && (characters[i] == "t")
118       array[0] = array[0] + 1
```

```
119     elsif (characters[i] == "-")
120         pass
121     else
122         puts "Error, the letter #{characters[i]} does not belong to a permission"
123         exit
124     end
125 end
126 end
127
128
129
130
131 =begin
132 Para la solucion de este ejercicio, se va a seguir el siguiente metodo de resolucio:
133 - Se va a usar de ejemplo la cadena rwx-wxr-x.
134 - Para el analisis de una cadena, se realizara de la siguiente forma:
135     - checkFirsts --> analiza todas las primeras posiciones, es decir, r - r
136     - checkSeconds --> analiza todas las segundas posiciones, es decir, w w -
137     - checkThirds --> analiza todas las terceras posiciones, es decir, x x x
138 - Mientras se analizan, se van sumando los valores a la posicion correspondiente
139 del numero
140 =end
141
142 =begin
143 Usage: Body of the program
144 Date of creation: 22/04/2021
145 Members: Roberto Jiménez y Alberto Pérez
146 Last modification: 22/04/2021
147 Parameters:
148     Entry:
149     - Argv: Parameter introduced by console when program is called
150     Out:
151     - None
152 =end
153 if ARGV.length == 1
154     characters = ARGV[0].chars
155     if characters.length == 9
156         permissions = [0,0,0,0]
157         checkFirsts characters, permissions
158         checkSeconds characters, permissions
159         checkThirds characters, permissions
160         puts permissions
161     else
162         puts "Error, the string is not correct (length is not 9)"
163     end
164 else
165     puts "Error 1 parameter is necessary to execute the script"
166 end
167
168
```

```
1 require 'nokogiri'
2 require 'open-uri'
3
4 =begin
5 Ejercicio 8. Construir un Web Crawler (rastreador, araña, o robot) en Ruby capaz de
6 analizar
7 una página web concreta y que cree una base de datos con la información obtenida
8 =end
9
10 =begin
11 Usage: Gets source code and urls from a page
12 Name of method: getSoruceCode
13 Date of creation: 29/04/2021
14 Members: Roberto Jiménez y Alberto Pérez
15 Last modification: 06/05/2021
16 Parameters:
17   Entry:
18     - URL: Parameter of the program
19   Out:
20     - list with urls without parsing
21 =end
22
23 def getSourceCode url
24   # Gets source code from page
25   parsed_data = Nokogiri::HTML.parse(URI.open(url))
26   list = []
27   puts parsed_data.class
28
29   #Gets all urls from source code
30   tags = parsed_data.xpath("//a")
31   tags.each do |tag|
32     list.append("#{tag[:href]}\t#{tag.text}")
33   end
34   return list
35 end
36
37 =begin
38 Usage: Insert data received into a DB
39 Name of method: insertDB
40 Date of creation: 29/04/2021
41 Members: Roberto Jiménez y Alberto Pérez
42 Last modification: 06/05/2021
43 Parameters:
44   Entry:
45     - list: list with urls
46     - url: URL from page to analyze
47   Out:
48     - None, output is saved in DB
49 =end
50
51 def insertDB list, url
52   #Insert data into DB
53   `echo -n #{url} > aux.csv`
54   `echo " ~" >> aux.csv`
55   `echo #{list} >> aux.csv`
56   `sudo mysql --local-infile=1 -h "localhost" -u rober "-prober" < insertarBD.sql`
57 end
58
59 #Parse information with Grep command
60 list = getSourceCode ARGV[0]
```



```
60 parsedList = `echo #{list} | grep -E 'http[s]*://[a-zA-Z0-9./?@=%&:_#-]*'`  
61 finallist = parsedList.split(',')  
62 insertDB finallist, ARGV[0]  
63 for url in finallist  
64     puts url  
65 end  
66  
67
```

```
1
2 CREATE DATABASE IF NOT EXISTS practicaAS2;
3 USE practicaAS2;
4
5 /*
6     Create table pagina and add urls from main url page
7     Delete each time to not generate too much info
8 */
9 DROP TABLE IF EXISTS pagina;
10 CREATE TABLE IF NOT EXISTS pagina(
11     urlPagina TEXT,
12     subenlaces TEXT
13 );
14
15
16 /*
17     Load CSV created from info received
18 */
19 LOAD DATA
20 LOCAL INFILE 'aux.csv'
21 INTO TABLE pagina
22 FIELDS TERMINATED BY '~'
23 (urlPagina, subenlaces);
24
25
```