

# Grupo J1

## Examen de shell script

Nombre y Apellidos: \_\_\_\_\_

Realizar un script denominado *elGrande*. No tiene parámetros. La invocación será pues, por ejemplo

```
$ ./elGrande
```

### La especificación del script es la siguiente:

1. Escribirá en pantalla el nombre y apellidos del autor.
2. Control del número de parámetros de entrada. El programa avisa y termina si hay parámetros.
3. Generará un fichero denominado *unaLinea* que contenga una única línea. La línea consta de un número aleatorio de palabras, entre 0 y 19. Cada palabra es un número aleatorio entre 0 y 199. Las palabras estarán separadas por un espacio en blanco. El fichero debe acabar con un fin de línea
4. Determinará el número más grande del fichero *unaLinea*, lo guardará en la variable *grande* y lo imprimirá por pantalla
5. Escriba una función denominada *dameElGrande* que genere un fichero, cuyo nombre toma por parámetro, de las mismas características que *unaLinea* (si el fichero ya existiese se sobrescribe). Además *dameElGrande* devolverá el valor más grande.
6. Use la función *dameElGrande* para generar un fichero denominado *Matriz* que contenga 20 líneas de las características de *unaLinea* y determine el número más grande de *Matriz*.

### Recomendaciones:

El tiempo es escaso. Asegúrese de que el programa funciona paso a paso.

**Tiempo:** 1 hora y 15 minutos

**Entrega:** El fichero *elGrande* se entregará en el campus virtual en la tarea correspondiente al grupo.

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

```
function dameElGrande
{
    nrPalabras=`expr $RANDOM % 20`
    rm $1 2>/dev/null
    ii=0
    while [ $ii -le $nrPalabras ]
    do
        palabra=`expr $RANDOM % 200`
        echo -n "$palabra " >> $1
        ii=`expr $ii + 1`
    done
    echo >> $1
    #echo El fichero creado en la función se llama $1
    #echo ... y contiene
    #cat $1
    read todos < $1

    G=0
    for palabra in $todos
    do
        if [ $palabra -gt $G ]; then
            G=$palabra
        fi
    done
    # echo devuelvo $G
    return $G
}
```

```
#
# 1. Escribirá en pantalla el nombre y apellidos del autor.
#
echo Juan Carlos Díaz Martín
```

```
#
# 2. Control del número de parámetros de entrada. El programa avisa y termina si hay parámetros.
#
if [ ! $# -eq 0 ]; then
    echo Use: $(basename $0)
    exit
fi
```

```
#
# 3. Generará un fichero denominado unaLinea que contenga una única línea y acabe en fin de línea
#
nrPalabras=`expr $RANDOM % 20`
i=0
while [ $i -le $nrPalabras ]
do
    palabra=`expr $RANDOM % 200`
    # echo $palabra
    echo -n "$palabra " >> unaLinea
    i=`expr $i + 1`
done
echo >> unaLinea
echo el fichero resultante es:
cat unaLinea
echo
```

```
#
# 4. Determinará el número más grande de la línea y lo guardará en la variable numeroDeLineas
#
#read -p "Voy por el mas grande " a
read todos < unaLinea
# echo $todos
rm unaLinea

grande=0
for palabra in $todos
do
    if [ $palabra -gt $grande ]; then
        grande=$palabra
    fi
done
echo El más grande es: $grande
```

```
#
# 5. Escriba una función que genere el fichero parámetro y devuelva el valor más grande.
#
#read -p "Voy por la función " a
dameElGrande myFile
echo El grande de la funcion es $?
```

```
#
# 6. Use la función para generar un fichero denominado Matriz que contenga 20 líneas de
# las características de una línea y devuelva su número más grande
#
```

```
#read -p "Voy por la matriz " a
rm Matrix 2>/dev/null
```

```
mGrande=0;
for i in {1..20}
do
    dameElGrande fAux
    GG=$?
    if [ $GG -gt $mGrande ]; then
        mGrande=$GG
    fi
    cat fAux >> Matrix
done

echo; echo Matriz es:
cat Matrix
echo
echo El numero mas grande es $mGrande

rm myFile fAux
```