

# 75.08:Sistemas Operativos - Trabajo Práctico

Gonzalo Beviglia

Tomás Boccardo

Damian Manoff

Diego Montoya

Federico Quevedo

Lucas Simonelli

13 de mayo de 2013

## Índice

<b>1. Hipótesis y aclaraciones globales</b>	<b>3</b>
<b>2. Problemas relevantes</b>	<b>4</b>
<b>3. Archivo readme</b>	<b>5</b>
<b>4. Listado de comandos y funciones</b>	<b>9</b>
4.1. Instal4.sh . . . . .	9
4.1.1. Archivos de input, intermedios y output . . . . .	9
4.1.2. Parametros y opciones . . . . .	9
4.1.3. Invocacion manual y/o automatica con ejemplos . . . . .	9
4.1.4. Hipotesis y aclaraciones especificas del comando . . . . .	9
4.2. Inicio4.sh . . . . .	9
4.2.1. Justificacion . . . . .	9
4.2.2. Archivos de input, intermedios y output . . . . .	9
4.2.3. Parametros y opciones . . . . .	9
4.2.4. Invocacion manual y/o automatica con ejemplos . . . . .	9
4.2.5. Hipotesis y aclaraciones especificas del comando . . . . .	9
4.3. Reporte4.pl . . . . .	9
4.3.1. Archivos de input, intermedios y output . . . . .	9
4.3.2. Parametros y opciones . . . . .	10
4.3.3. Invocacion manual y/o automatica con ejemplos . . . . .	10
4.3.4. Hipotesis y aclaraciones especificas del comando . . . . .	10
4.4. Glog4.sh . . . . .	10
4.4.1. Justificacion . . . . .	10
4.4.2. Archivos de input, intermedios y output . . . . .	10
4.4.3. Parametros y opciones . . . . .	10
4.4.4. Invocacion manual y/o automatica con ejemplos . . . . .	10
4.4.5. Hipotesis y aclaraciones especificas del comando . . . . .	11
4.5. Interprete.sh . . . . .	11
4.5.1. Justificacion . . . . .	11
4.5.2. Archivos de input, intermedios y output . . . . .	11
4.5.3. Parametros y opciones . . . . .	11
4.5.4. Invocacion manual y/o automatica con ejemplos . . . . .	11
4.5.5. Hipotesis y aclaraciones especificas del comando . . . . .	11
4.6. Detecta4.sh . . . . .	11
4.6.1. Justificacion . . . . .	11
4.6.2. Archivos de input, intermedios y output . . . . .	11
4.6.3. Parametros y opciones . . . . .	11
4.6.4. Invocacion manual y/o automatica con ejemplos . . . . .	11
4.6.5. Hipotesis y aclaraciones especificas del comando . . . . .	12
4.7. Mov4.sh . . . . .	12

4.7.1.	Justificacion . . . . .	12
4.7.2.	Archivos de input, intermedios y output . . . . .	12
4.7.3.	Parametros y opciones . . . . .	12
4.7.4.	Invocacion manual y/o automatica con ejemplos . . . . .	12
4.7.5.	Hipotesis y aclaraciones especificas del comando . . . . .	12
4.8.	Start4.sh . . . . .	12
4.8.1.	Justificacion . . . . .	12
4.8.2.	Archivos de input, intermedios y output . . . . .	12
4.8.3.	Parametros y opciones . . . . .	12
4.8.4.	Invocacion manual y/o automatica con ejemplos . . . . .	12
4.8.5.	Hipotesis y aclaraciones especificas del comando . . . . .	12
4.9.	Stop4.sh . . . . .	12
4.9.1.	Justificacion . . . . .	12
4.9.2.	Archivos de input, intermedios y output . . . . .	12
4.9.3.	Parametros y opciones . . . . .	12
4.9.4.	Invocacion manual y/o automatica con ejemplos . . . . .	13
4.9.5.	Hipotesis y aclaraciones especificas del comando . . . . .	13
4.10.	Vlog4.sh . . . . .	13
4.10.1.	Justificacion . . . . .	13
4.10.2.	Archivos de input, intermedios y output . . . . .	13
4.10.3.	Parametros y opciones . . . . .	13
4.10.4.	Invocacion manual y/o automatica con ejemplos . . . . .	13
4.10.5.	Hipotesis y aclaraciones especificas del comando . . . . .	13
<b>5.</b>	<b>Código fuente de los scripts</b>	<b>14</b>
5.1.	Instal4.sh . . . . .	14
5.2.	Inicio4.sh . . . . .	29
5.3.	Reporte4.pl . . . . .	39
5.4.	Glog4.sh . . . . .	54
5.5.	Interprete.sh . . . . .	56
5.6.	Detecta4.sh . . . . .	63
5.7.	Mov4.sh . . . . .	67
5.8.	Start4.sh . . . . .	70
5.9.	Stop4.sh . . . . .	71
5.10.	Vlog4.sh . . . . .	73

## 1. Hipótesis y aclaraciones globales

- Todos los comandos que seteen variables de entorno deberán correrse de la siguiente forma:

\$ . Comando

Esto es necesario debido a que se generarán variables que deben estar disponibles en el entorno actual.

- Los comandos se corren desde el directorio “grupo4/”, ya que todo el manejo de rutas se hace respecto de la ubicación de los scripts.
- Cada comando cuenta con su archivo de log propio.
- Cada comando debe verificar que las variables de entorno estén correctamente seteadas.
- El orden de los parámetros que recibe el comando Glog4.sh son “Mensaje” [“Tipo”] “Comando Invocante”
- Al filtrar préstamos en Reporte4.pl se utiliza del archivo maestro aquel registro que tenga el mismo código.
- El archivo de log, en vez de estar separado por -, se separa de una manera que sea amigable para la visualización, cosa de que, complementado con el comando Vlog4, su visualización pueda ser más bella.

## 2. Problemas relevantes

- Un problema que surgió a la hora de realizar el comando Glog4 fue con respecto al tamaño máximo que podía alcanzar este archivo. Hubo un desentendimiento respecto de si la variable LOGSIZE debía ser interpretada como un tamaño en KB o si esta representaba una cantidad máxima de líneas de archivo de log. Se decidió, debido a que en el Instal4 se especificaba que la unidad era en KB, que esta debía ser la unidad a utilizar. Luego, se tuvo que adaptar lo que antes en el Glog4 se hacía con cantidad de líneas, para que sepa trabajar bien con un tamaño en KB.
- Con respecto al comando Mov4, surgió un problema de determinación de lo que representa el directorio destino (o archivo destino) a donde uno quiere mover un archivo. Dado que el comando de Linux ‘mv’ opera tanto ‘mv directorio/origen.extension otroDir/destino.extension’ como ‘mv directorio/origen.extension otroDir/’ se pensó que el Mov4 debía tener el mismo comportamiento. Para facilitar las cosas, se optó por permitir que el comando reciba únicamente archivos origen y destino, y no directorios, debido a que hacía las cosas más sencillas cuando uno quería utilizar el comando para renombrar el archivo.
- Surgió un problema respecto a la interpretación del enunciado del comando Reporte4, ya que al no tener entendimiento acerca de préstamos entre países, no se entendía bien que registros debían compararse contra cuales. Se tomó una decisión que se documenta en las hipótesis tanto generales como individuales de ese comando.
- Con respecto a la integración, un problema que ocurrió fue la falta de un número significativo de datos de prueba, que se pudieran ir procesando por los distintos comandos. Al ser publicados estos, la integración se desarrolló de mejor manera.
- Una decisión que se tuvo que tomar fue desde donde se debían correr los scripts. Como hay una diferencia entre comandos que setean variables y otros que no, muchos archivos que no modifican variables se corren utilizando “./Comando4.sh”. Esto trajo problemas a la hora de integración, ya que todos los comandos se ubicarían en BINDIR. Se optó utilizar esta carpeta contenedor y, a la vez, contar con una copia de los mismos en el directorio de grupo. Esto fue una decisión operativa que facilitó la integración entre módulos. El comportamiento de la ejecución de algunos comandos fuera del directorio en donde estos estaban ubicados era errática y, en algunos momentos, difícil de comprender con las herramientas de ‘debugging’ con las que contábamos.
- Nuevamente con respecto a la integración, un problema que surgió fue el comportamiento de algunos scripts que utilizaban funciones que estaban declaradas en otros módulos. Las funciones eran triviales, como limpiar la pantalla para emprolijar los mensajes mostrados. Al estar esta función declarada en el Instal4, y como en primera instancia la prueba de los scripts se realizaba corriéndolos uno después del otro, no había problema. En cuanto empezamos a experimentar con cerrar la terminal (restaurando de esta manera el entorno), comenzaba a haber mensajes de que la función en cuestión no estaba declarada. La decisión que se tomó fue la de duplicar esta función en el código fuente de los comandos que la invocaban.
- Dado que los scripts eran corridos por terminal para testear los mismos, ‘debuggearlos’ era una tarea engorrosa, casi completamente basada en llamadas al comando ‘echo’. Esto no es un problema al que le encontramos solución, pero de tener que trabajar en algún otro momento con scripts, un entorno que provea ‘breakpoints’ e investigador de variables sería muy útil. Por las dimensiones de este trabajo práctico se optó por no investigar acerca de tal entorno de desarrollo.

### 3. Archivo readme

===== Instrucciones de Instalación =====

El script Instal4.sh instalará el sistema en su computadora. Debe ser ejecutado desde el directorio en el que se encuentra el script de la siguiente manera:

```
. Instal4.sh
```

Se necesita correr como '. Instal4.sh' y no como './Instal4.sh' puesto que el script registra ciertas variables en el entorno de ejecución, y de ser ejecutado en un subentorno del actual (con './Instal4.sh' esas variables de ambiente se perderían y no sería posible continuar con los demás scripts del sistema.

Parámetros: 1 (opcional): directorio donde se realizará la instalación

===== Instrucciones de Inicio del entorno =====

El script Inicio4.sh inicializará el entorno de su computadora, preparándolo para que los scripts que componen a este sistema funciones de la manera en la que fue pensado el mismo.

El script no recibe parámetros, la única entrada que recibe por parte del usuario es, durante la ejecución del mismo, el tiempo de espera (TESPERA) entre ciclos que esperará el script Detecta4.sh y la cantidad de ciclos (CANLOOP) que realizará el demonio antes de detener su ejecución. El script Inicio4.sh ofrece la posibilidad de comenzar la ejecución del demonio.

Este script es el encargado de dar permisos de ejecución al resto de los scripts que componen al sistema, y de verificar que la instalación haya sido satisfactoria. En caso contrario, se sugerirá al usuario que corra el script Instal4.sh para reparar o instalar nuevamente el sistema. Se chequeará que todos los scripts que componen al sistema tengan permiso de ejecución y, en caso de no tenerlos, se dará permiso de ejecución en modo usuario a los mismos (chmod u+x script.sh).

Al igual que el instalador, Inicio4.sh debe ser ejecutado en el entorno actual de la terminal, y no en uno hijo, ya que setea variables que serán usadas por el demonio.

El modo correcto de ejecución es, en el directorio donde se encuentra el script:

```
. Inicio4.sh
```

===== Instrucciones del script de Log =====

El script Glog4.sh escribe un mensaje de un determinado tipo en el archivo de log de un determinado comando. Por ejemplo, si quisieramos loggear un advertencia al usuario mientras se ejecuta el comando Inicio4, utilizaríamos el comando Glog4 de la siguiente manera:

```
./Glog4.sh "Esto es una advertencia" "WARNING" "Inicio4"
```

Este comando escribiría en el log correspondiente, con un formato específico para el tipo de mensaje que se esta loggeando. Como se muestra en el ejemplo, los parámetros del comando son los siguientes:

- 1 - Mensaje a loggear.
- 2 - Tipo de mensaje (opcional), puede ser uno de los siguientes:
  - \* MENSAJE
  - \* HEADER (para loggear comienzo y fin de ejecucion)
  - \* ERROR
  - \* WARNING
  - \* SEVERE
- 3 - Comando en cuyo archivo de log se quiere escribir

Los archivos de log pueden alcanzar un tamaño (en Kb) máximo, luego del cual se borran las primeras entradas y se conservan las últimas, dejando al archivo en un tamaño más pequeño que el original. Este tamaño esta definido en la variable LOGSIZE, definida en el instalador. El tamaño con el que se deja un archivo de log luego de reducirlo es, arbitrariamente, 60 lineas. Luego de reducirlo en tamaño, se deja constancia en el mismo log que se redujo el tamaño del mismo.

Además de el mensaje, se guarda registro del usuario que provocó la entrada de log y la fecha y hora en la que se produjo la entrada.

===== Instrucciones del script de visualización del log =====

El script Vlog4.sh se encarga de mostrar por pantalla las entradas de algun log en particular que cumplan con algún filtro establecido por el usuario. Esto facilita la lectura de los archivos de log en el caso de que se quiera buscar algun registro en particular que se haya realizado en el archivo.

Los parámetros del comando son los siguientes:

```
-s "palabra o frase a buscar"
-f "comando en cuyo log se buscará"
-n "cantidad máxima de líneas a mostrar"
-i "severidad de los mensajes a mostrar" (opcional)
-u "usuario que provoco la entrada de log" (opcional)
```

Los parámetros se leen como 'short options' por lo que su orden de aparición es indistinto para el comando. A continuación se explicará, de manera breve, que resultado provoca cada parámetro.

-s: buscará la frase o palabra especificada entre comillas en las líneas del archivo de log especificado. La búsqueda es 'case insensitive' y 'whole words', es decir, no diferencia entre capitalizaciones pero busca el string especificado como una palabra completa.

-f: se especifica el comando en cuyo log se buscará. No se debe especificar en este parámetro el nombre de archivo, ya que el comando Vlog4 conoce la extension de log especificada en el instalador.

-n: especifica la cantidad máxima de líneas a mostrar. De haber más de n líneas que cumplan con la búsqueda realizada, se mostrarán las últimas n entradas del archivo de log.

-i: filtra los mensajes por la severidad especificada, de manera de obtener solo aquellos mensajes de la severidad pedida (por ejemplo, únicamente los 'SEVERE').

-u: filtra los mensajes por el nombre del usuario que ejecutó el comando que generó la entrada de log.

Ejemplo de utilización:

```
./Vlog4.sh -s "No se ha encontrado" -f "Inicio4" -n "10" -i "ERROR"
```

==== Instrucciones del script utilizado para mover archivos =====

El script Mov4.sh es el encargado de mover archivos de un directorio a otro (o de cambiarle el nombre si el directorio origen es el mismo que el destino, especificando distinto nombre de archivo). La responsabilidad de este script es de saber responder ante la existencia de un archivo duplicado en el directorio destino. Es decir, dada la siguiente estructura de archivos:

```
--grupo4/  
  --archivo1.tab  
  --tabs/  
    --archivo1.tab
```

Si se quisiera mover el archivo 'archivo1.tab' al directorio 'tabs', estaríamos ante un archivo duplicado. Este script crea una carpeta 'dup' en el directorio destino, en donde colocará una secuencia de archivos duplicados (archivo1.tab.1, archivo1.tab.2, etc). La secuencia generada es específica de cada archivo, de esta manera, podran existir los archivos duplicados del tipo 'archivo1.tab.1' y 'archivo2.tab.1' dentro del mismo directorio de archivos duplicados.

Los parámetros recibidos por este script son los siguientes:

- 1 - Path del archivo origen (ejemplo: "grupo4/archivo1.tab").
- 2 - Path del archivo destino (ejemplo: "grupo4/tabs/archivo1.tab").

Cada tipo de movimiento es loggeado en el archivo de log del comando Mov4. Los distintos tipos de resultado son los siguientes:

- Archivo y directorio origen es igual a archivo y directorio destino, en cuyo caso no se realiza ninguna acción.
- No existe el archivo origen, en cuyo caso tampoco se realiza ninguna acción.
- El directorio destino no existe, aquí tampoco se realiza ninguna acción.
- No existe aún un archivo con el nombre destino especificado, en cuyo caso se mueve el archivo a este directorio sin problemas.
- Existe un archivo con el mismo nombre que el destino especificado, en cuyo caso se procede a manejar la colisión de la manera antes descripta.

==== Instrucciones de los scripts utilizados para iniciar y detener procesos =====

El script Start4.sh es el encargado de iniciar los procesos que se le pasan por parámetro.

El script se ejecuta de la siguiente manera:

```
./Start4.sh proceso_a_iniciar
```

Start4 se ocupa de verificar la existencia de ese proceso corriendo actualmente y en dicho caso, no realiza ninguna acción, asentando esto en el log del usuario.

El script Stop4.sh es el encargado de detener los procesos que se le pasan

por parámetro, ya sea con el nombre del mismo o con el PID.

Los parámetros del comando son los siguientes:

-n "nombre del proceso a detener"

-p "PID del proceso a detener"

Por defecto, si Stop4 recibe ambos parámetros, solamente interpretará el primero de los dos, desestimando el otro.

Ejemplos de ejecución del script:

./Stop4.sh -n nombre\_proceso\_a\_detener

./Stop4.sh -p PID\_proceso\_a\_detener

Stop4 se ocupa de verificar la existencia de ese proceso corriendo actualmente y en dicho caso, lo detendrá luego de obtener su PID.



## 4. Listado de comandos y funciones

### 4.1. Instal4.sh

#### 4.1.1. Archivos de input, intermedios y output

- Input: no recibe.
- Output: Instal4.log.
- Intermedios: el log se copia en un dir auxiliar antes de instalar.

#### 4.1.2. Parametros y opciones

Tiene un parámetro opcional, el directorio donde se realizará la instalación, por defecto, grupo4/

#### 4.1.3. Invocacion manual y/o automatica con ejemplos

- Invocación sin parámetros (instala en ./grupo4/): . Instal4.sh
- Invocación con parámetros (instala en \$DIR): . Instal4.sh \$DIR

#### 4.1.4. Hipotesis y aclaraciones especificas del comando

- Se eligio permitir elegir el directorio de instalación

### 4.2. Inicio4.sh

#### 4.2.1. Justificacion

Se necesita de este comando para la inicialización del ambiente. Se verifica que la instalación del sistema este completa y se agrega a la variable PATH el archivo en donde se encuentran este comando y los demás que componen al sistema.

#### 4.2.2. Archivos de input, intermedios y output

No recibe archivos de input ni genera archivos intermedios. Escribe en un archivo de log por medio de otro comando. Si este no existe, se lo crea.

#### 4.2.3. Parametros y opciones

No recibe parametros ni opciones.

#### 4.2.4. Invocacion manual y/o automatica con ejemplos

Invocación: “. Inicio4.sh”

#### 4.2.5. Hipotesis y aclaraciones especificas del comando

Como el ejemplo anterior ilustra, el comando debe ser ejecutado en el entorno actual de la terminal, debido a que setea variables que serán utilizadas por otros comandos. De ser ejecutado en un subentorno (“./Inicio4.sh”) estas variables no podrían ser utilizadas por los otros comandos.

### 4.3. Reporte4.pl

#### 4.3.1. Archivos de input, intermedios y output

- Input: PPI.mae, prestamos.<pais>
- Output: salida de texto del programa, outputReporte.fecha (opcional)

#### 4.3.2. Parametros y opciones

- -c (codigo de pais) - OBLIGATORIO
- -s (codigo del sistema)
- -a (anio)
- -p [periodo (AAAA/MM)]
- -r [Rango de periodos (AAAA/MM-AAAA/MM)]
- -w (graba un reporte de salida)
- -h (imprime menu de ayuda)

#### 4.3.3. Invocacion manual y/o automatica con ejemplos

- Invocación con parámetros (ejecuta consulta filtrada del pais de código A, año 2012, código de sistema 12):  

```
$ ./Reporte4.pl -c A -a 2012 -s 12
```
- Invocación con el parámetro mínimo [código país] para funcionar (ejecuta consulta filtrada del pais de código A):  

```
$ ./Reporte4.pl -c A
```

#### 4.3.4. Hipotesis y aclaraciones especificas del comando

- En la consulta filtrada, se compara cada registro del maestro con el de préstamos correspondiente, encontrado según el enunciado indica.
- El recálculo se calcula para cada registro del archivo de préstamos aunque la recomendación sea solo para uno del de préstamos.

### 4.4. Glog4.sh

#### 4.4.1. Justificacion

Se necesita de este comando para escribir y generar los archivos de log que utilizan los demás comandos (menos Instal4, que genera su propio log).

#### 4.4.2. Archivos de input, intermedios y output

No recibe archivos de input ni genera archivos intermedios. Genera un archivo output, el archivo de log, si este no existe aún.

#### 4.4.3. Parametros y opciones

Recibe dos parámetros obligatorios y uno opcional. Los parámetros obligatorios son el mensaje que se va a loggear y el comando que invoca la escritura en archivo de log. El parámetro opcional es la severidad del mensaje a loggear.

#### 4.4.4. Invocacion manual y/o automatica con ejemplos

“./Glog4.sh ‘Esto es un mensaje’ ‘SEVERE’ ‘Comando’”

#### **4.4.5. Hipotesis y aclaraciones especificas del comando**

Los parámetros deben ser provistos en el orden ilustrado anteriormente. De no proporcionar tipo de mensaje, la invocación sería la siguiente: “./Glog4.sh ‘Esto es un mensaje’ ‘Comando’”. Los tipos de mensaje son: MENSAJE, HEADER, ERROR, WARNING y SEVERE. La escritura en los archivos no esta separada por -, sino que se escribe de una manera que sea más amigable a la vista. Por ejemplo: “ user (2013-05-08 17:27:13) MENSAJE Usuario ingresa TESPORA=2”

### **4.5. Interprete.sh**

#### **4.5.1. Justificacion**

El propósito del intérprete es traducir los distintos archivos provenientes de los diferentes países, que están en ACEPDIR, llevándolos a un formato estándar para el sistema. El hecho de traducir se refiere a interpretar cada campo y transformarlo a un formato que entienda el sistema.

#### **4.5.2. Archivos de input, intermedios y output**

El intérprete no recibe archivos de input.  
Los archivos de output del intérprete son:

- Archivos de Préstamos Personales por país PROCDIR/PRESTAMOS.pais
- Archivos (duplicados) Rechazados RECHDIR/nombre del archivo
- Archivos Procesados PROCDIR/pais-sistema-año-mes
- Log LOGDIR/Interprete.LOGEXT

#### **4.5.3. Parametros y opciones**

El intérprete no recibe parámetro alguno, pero obligatoriamente debe ejecutar luego del instalador ya que necesita las variables de entorno seteadas por el mismo.

#### **4.5.4. Invocacion manual y/o automatica con ejemplos**

“./Intérprete.sh

#### **4.5.5. Hipotesis y aclaraciones especificas del comando**

### **4.6. Detecta4.sh**

#### **4.6.1. Justificacion**

Se necesita este comando para validar los archivos que llegan y llamar al siguiente comando para los archivos validos.

#### **4.6.2. Archivos de input, intermedios y output**

- Input: Maestro de Países y Sistemas. Archivos a verificar.
- Output: Archivos aceptados y rechazados. Detecta4.log

#### **4.6.3. Parametros y opciones**

No recibe parametros ni opciones.

#### **4.6.4. Invocacion manual y/o automatica con ejemplos**

Invocación: ‘./Detecta4.sh & ’ Se utiliza el ‘& ’ para que se ejecute en el background, ya que el proceso es de tipo demonio.

#### **4.6.5. Hipotesis y aclaraciones especificas del comando**

### **4.7. Mov4.sh**

#### **4.7.1. Justificacion**

Se necesita de este comando para mover archivos de una ubicación a otra dejando que este se ocupe de las colisiones entre archivos. El comando maneja y soluciona el caso en el que exista un archivo con el nombre del que se está moviendo en el directorio al que se lo quiere mover.

#### **4.7.2. Archivos de input, intermedios y output**

No recibe archivos de inputs ni genera intermedios. Se podría decir que al mover el archivo de un directorio al otro se está generando un archivo output.

#### **4.7.3. Parametros y opciones**

Recibe dos parámetros obligatorios y uno opcional. Los dos parámetros obligatorios que recibe son “ruta/origen/origen.extension” y “ruta/destino/destino.extension”. El parámetro opcional que recibe es el comando que invoca al Mov4. El comando invocante debe ser pasado sin la extensión “.sh/.pl”.

#### **4.7.4. Invocacion manual y/o automatica con ejemplos**

“./Mov4.sh ‘ruta/origen/origen.txt’ ‘ruta/destino/destino.txt’ ‘Comando’ ”

#### **4.7.5. Hipotesis y aclaraciones especificas del comando**

### **4.8. Start4.sh**

#### **4.8.1. Justificacion**

Se necesita de este comando para poder iniciar los procesos dentro de un entorno controlado, verificando no iniciarlo si ya se encuentra corriendo.

#### **4.8.2. Archivos de input, intermedios y output**

No recibe archivos de inputs ni genera intermedios.

#### **4.8.3. Parametros y opciones**

Recibe como parámetro el nombre del proceso que se debe iniciar.

#### **4.8.4. Invocacion manual y/o automatica con ejemplos**

“./Start4.sh ‘nombre del proceso a iniciar’ ”

#### **4.8.5. Hipotesis y aclaraciones especificas del comando**

### **4.9. Stop4.sh**

#### **4.9.1. Justificacion**

Se necesita de este comando para poder detener los procesos, verificando su existencia.

#### **4.9.2. Archivos de input, intermedios y output**

No recibe archivos de inputs ni genera intermedios.

#### **4.9.3. Parametros y opciones**

Recibe como parámetro el nombre del proceso que se debe detener, o su PID, según la opción elegida. -n: Opción para detener el proceso con el nombre del mismo -p: Opción para detener el proceso con el PID

#### **4.9.4. Invocacion manual y/o automatica con ejemplos**

“./Stop4.sh -n ‘nombre del proceso a detener’ ” “./Stop4.sh -p ‘PID del proceso a detener’ ”

#### **4.9.5. Hipotesis y aclaraciones especificas del comando**

Si se invoca el comando utilizando ambas opciones, solamente se tendrá en cuenta la primera de ellas

### **4.10. Vlog4.sh**

#### **4.10.1. Justificacion**

Se necesita de este comando para generar una visualización del contenido de los archivos de log. Como estos pueden ser muy extensos, este comando proporciona de una manera cómoda de consultarlos.

#### **4.10.2. Archivos de input, intermedios y output**

No recibe archivos de input y no genera archivos intermedios ni archivos output.

#### **4.10.3. Parametros y opciones**

Recibe parámetros de tipo “short options”, cuya descripción se encuentra en la sección “Codigo fuente de los scripts”.

#### **4.10.4. Invocacion manual y/o automatica con ejemplos**

“./Vlog4.sh -s ‘mensaje buscado’ -f ‘comando’ -n ‘15’”

#### **4.10.5. Hipotesis y aclaraciones especificas del comando**

El comando provisto no debe contar con la extensión del mismo (“.sh/.pl”).