

Trabajo Práctico: Sistema de reserva de entradas sin costo comercial

Grupo N° 8 TEMA B
2do. Cuatrimestre de 2013
75.08 Sistemas Operativos
Facultad de Ingeniería, Universidad de Buenos Aires

Índice

1. Hipótesis y Aclaraciones globales.	3
2. Problemas Relevantes	4
2.1. Instalar	4
2.2. Está Corriendo	4
2.3. Mover_B	4
2.4. Grabar_L	4
3. Archivo Readme	5
4. Esquema de Ejecución	8
4.1. Readme_Instalar	8
4.2. Readme_Iniciar	8
4.3. Readme_Recibir	8
4.4. Readme_Reservar	8
4.5. Readme_Grabar_L	9
4.6. Readme_Imprimir	9
4.7. Readme_Matar_D	10
5. Listado de nuevas funciones y/o comandos auxiliares	11
5.1. EstaInicializado.sh	11
5.2. EstaCorriendo.sh	11
5.3. Mover_B.sh opción -c	11
5.4. Eliminar_B.sh	11
6. Listado de Nuevos Archivos	12
6.1. LOGDIR/count	12
6.2. GRUPO/instalacion.log	12
6.3. CONFDIR/backup	12
7. Listado de Datos	13
8. Prueba del Camino Feliz	14

1. Hipótesis y Aclaraciones globales.

- A Mover_B y Eliminar_B se le pasan rutas absolutas.
- Matar_D, finaliza todas las instancias del programa solicitado.
- Start_D no inicia el proceso lo el mismo ya se encuentra corriendo en el sistema o el ambiente no se ha inicializado.
- Si se desea eliminar un archivo inexistente no se detendra la ejecución del script. Sólo se informará en el Log.
- Los nombres de los archivos de Logs pueden incluir puntos.
- El tamaño del Log se verifica luego de escribir, por lo tanto, después de una escritura válida el Log tiene un tamaño permitido.

2. Problemas Relevantes

2.1. Instalar

Uno de los principales problemas que surgió al momento de realizar la instalación del sistema, fue que se debía guardar el log de la misma cuando aún no se habían definido el directorio asignado para tal fin. Así mismo sucedía que al querer utilizar el script de Grabar_L o Mover_B, estos mismos iban a ser 'movidos' al realizar la instalación y esto generaba conflictos, ya que se estaba moviendo el mismo archivo que se estaba ejecutando. Para solucionarlo, se copiaron los archivos binarios que generaban el conflicto en vez de moverlos.

2.2. Está Corriendo

Para varias funcionalidades, se pide chequear que un script no se ejecute más de una vez. A la hora de plantear cómo resolver este problema, surgieron varias formas de solucionarlo. Una forma era que cada script se chequeara a sí mismo, ya fuera mediante una variable de ambiente, un archivo auxiliar que guardara dicho estado, etc. El problema de este tipo de soluciones es que podían generar la indeseada situación de que dos procesos accedieran al mismo tiempo a la variable o archivo, si bien era un caso poco probable, era posible así que fue descartada. Finalmente se optó por acudir al comando 'ps' con la opción -e y utilizando expresiones regulares, filtrar los resultados y así verificar si el proceso buscado se encontraba o no en ejecución.

2.3. Mover_B

La característica básica de este script no presentó muchos inconvenientes. Es decir, mover archivos entre directorios es algo sencillo. El problema surge cuando en el directorio destino ya se encuentra el archivo que queremos mover. En nuestro caso optamos por mover el archivo con un nombre formado por: nombreOriginal+N , en donde N es la cantidad de documentos con el mismo nombre en el directorio. Adicionalmente, a este script, se le agregó funcionalidad para copiar archivos. Siendo necesaria en su invocación agregar el parámetro c".

2.4. Grabar_L

El motivo de llevar adelante un 'Logger' radica en la trazabilidad de un programa en ejecución. Su implementación resulta ser de una dificultad moderada, pero, por el contrario, su utilidad es mucha. Para este sistema, se necesitan generar archivos de logs para cada script utilizado. Además se le da un trato especial al de instalación, ya que este nunca se recorta y se ubica en un directorio distinto al resto. Para evitar la duplicación de código, se utiliza el mismo script para todos los archivos de log generados. Tomando como un caso especial la invocación de 'Grabar_L.sh' con el parámetro '-ins'. En esta oportunidad se genera una ruta distintiva y se anula la funcionalidad de limite de tamaño.

3. Archivo Readme

*****TP SO7508 Segundo Cuatrimestre 2013. Tema B
Copyright Grupo 08*****

*****INSTALACION*****

1. Conecte el dispositivo en la PC.
 2. Copie el archivo "TP-G8-TB.tgz" en el directorio donde desea que se instale el sistema.
 3. Descomprima el archivo. Si desea hacerlo mediante la línea de comandos ejecute lo siguiente:
\$ tar xvf TP-G8-TB.tgz
Al descomprimir se extraerán los archivos necesarios para la instalación y una carpeta "datos"
 4. Abra una terminal en este directorio y ejecute el script Instalar_TP.sh.
 5. Una vez ejecutado le pedirá que acepte los acuerdos de licencia. Deberá ingresar 'Si' para poder continuar.
 6. El instalador le solicitará que realice las configuraciones de los directorios de trabajo del sistema, para cada uno de ellos le ofrecerá una opción default, si no está de acuerdo con ella podrá cambiarla e ingresar el directorio que considere más conveniente.
5. Una vez realizadas las configuraciones, le preguntará si desea continuar con la instalación,
> Si ingresa 'Si', se crearán todos los archivos y directorios necesarios y finalizará la etapa de instalación.
> Si ingresa 'No', podrá volver y cambiar sus configuraciones.

*****EJECUCION*****

1. Para poder poner en funcionamiento el daemon que controla los arribos deberá tener inicializado el ambiente de trabajo. Ubíquese en el directorio donde ha guardado sus ejecutables y ejecute el script Iniciar_B.sh del siguiente modo:

\$. ./Iniciar_B.sh

(nota: se debe ejecutar como "punto espacio punto barra" de otro modo dará error y no podrá iniciar el programa.)

2. El inicializador de ambiente le preguntará si desea dar inicio al demonio Recibir_B.sh, si desea hacerlo en ese momento, ingrese 'Si'. Si ingresa 'No', no se preocupe! aún puede activarlo en otro momento del siguiente modo:

```
$/Start_D.sh Recibir_B.sh
```

*****DONDE PONGO MIS ARCHIVOS DE RESERVAS
?*****

En la instalación se le ha pedido que elija un directorio de arribos, coloque sus archivos de reservas en ese directorio y serán procesados automáticamente, siempre y cuando cumplan con el formato preestablecido, tanto en cuanto a su nombre como su contenido.

Consulte el directorio que ha configurado para rechazos si desea ver qué archivos han sido rechazados.

*****NO RECUERDO SI YA HE INICIALIZADO EL AMBIENTE*****

Puede volver a correr el script Iniciar_B.sh cuantas veces quiera, dentro del mismo se hace un chequeo y si el ambiente ya se encuentra inicializado no se harán modificaciones.

Del mismo modo si intenta correr el script Recibir_B y el ambiente no ha sido inicializado, se lo indicará con el mensaje: "Ambiente no inicializado."

Si aún no está seguro y quiere comprobarlo usted mismo, ejecute el siguiente comando:

```
$/EstaInicializado.sh
```

Este le mostrará un mensaje indicando si ya ha inicializado o no, el ambiente.

*****DETENER EJECUCION*****

Si en algún momento desea detener la ejecución del demonio coloque esto en la terminal:

```
$/Stop_D.sh Recibir_B.sh
```

4. Esquema de Ejecución

4.1. Readme_Instalar

```
*****Readme de Instalar_TP
*****
```

Esta funcion se encarga de el proceso de instalacion del programa de Reservas_B.
Si en algun momento se aborta la instalacion , el programa recuperara los datos almacenados por el usuario.
Se deja al usuario la libertad de la eleccion del nombre de los directorios para la instalacion.

- Parametros que recibe la funcion: Ninguno.
- Invocacion de la funcion : ./Instalar_Tp.sh
- Escritura en el log : Siempre.

4.2. Readme_Iniciar

4.3. Readme_Recibir

Este Script se encarga de validar el nombre y tipo de los archivos que se encuentren en el directorio ARRIDIR
Si el archivo tiene el formato correcto se moverá a ACEPDIR o REPODIR según corresponda.
Si el archivo no tiene el formato correcto , se moverá a RECHDIR.

No recibe parámetros.

Se debe iniciar SIEMPRE con "Start_D" del siguiente modo:

```
./Start_D.sh Recibir_B.sh
```

Se debe detener SIEMPRE con "Matar_D" del siguiente modo:

```
./Matar_D.sh Recibir_B.sh
```

4.4. Readme_Reservar

Esta funcion procesa los archivos que encuentre en ACEPDIR y los edita apropiadamente.

- Si el archivo ya fue procesado o es invalido , lo mueve al directorio de rechazados
- Si sus contenidos son invalidos , se los vuelca procesados en reservas.nok y se mueve el archivo a PROCDIR

- Si sus contenidos son validos , se los vuelca procesados en reservas.ok y se mueve el archivo a PROCDIR
- Al final del proceso se edita el archivo combos.dis con las nuevas disponibilidades

Se invoca sin parametro alguno.

Siempre escribe al log.

4.5. **Readme_Grabar_L**

Esta funcio crea y actualiza archivos de Log:

- Verifica que el ambiente este inicializado.
- No construye directorios.
- Se invoca como: "Grabar_L comandoASolicitante opcion mensaje" o "Grabar_L comandoASolicitante mensaje".
- Si en opción se coloca "-ins" se utiliza el Log de instalación.
- Opciones disponibles: "-i" = "Informativo" ; "-e" = "Error" ; "-fe" = "Fatal Error" ; "-w" = "Warning" ;

4.6. **Readme_Imprimir**

——— Descripcion:

Este proceso nos permite realizar consultas de disponibilidad , emitir el ranking de los 10 principales , generar la lista de invitados para las reservas confirmadas y generar los archivos para imprimir los tickets de entrada.

——— Modo de uso:

Para iniciar este script , primero debe:

- * Instalar el programa sin errores (con Instalar_TP.sh)
- * Ingresar a la carpeta de archivos ejecutables (por defecto bin, o la que haya configurado en la instalacion) y haber ejecutado Iniciar_B.sh

——— Formas de ejecucion:

./Imprimir_B.pl -a

Ayuda. Muestra un resumen del modo de uso.

./Imprimir_B.pl -i

Genera un listado de invitados a un evento.

./Imprimir_B.pl -d

Debe ingresarse un id de obra o de sala (o un rango de ids) y muestra información sobre salas y obras posibles y el numero de butacas disponibles.

```
./Imprimir_B.pl -r
```

Muestra un ranking de los 10 solicitantes más frecuentes.

```
./Imprimir_B.pl -t
```

Muestra un listado de tickets listo para imprimir.

Opcion `-w`:

Opcion combinable con `-r,-t,-d` o `-i`. Escribe en un archivo la salida de la ejecución, que de otra forma solo sería mostrada por pantalla.

Debe ejecutarse como, por ejemplo

```
./Imprimir_B.pl -d -w
```

o

```
./Imprimir_B.pl -w -d
```

4.7. *Readme_Matar_D*

Esta funcion mata procesos:

- No verifica inicialización del ambiente.
- Se puede llamar con nombre de proceso o ID.
- Siempre escribe al Log.
- Se invoca como: "`Matar_D comandoAMatar`".

5. Listado de nuevas funciones y/o comandos auxiliares

5.1. EstaInicializado.sh

Como su nombre lo indica, permite detectar si el ambiente en el que se encuentra corriendo el script ha sido inicializado o no. Ya que en varias partes del sistema se realiza este chequeo, se ha encapsulado aquí para evitar la repetición de código.

5.2. EstaCorriendo.sh

Esta funcionalidad permite verificar si un proceso se encuentra en ejecución o no. Surge de la necesidad de comprobar que un script como Recibir o Reservar, no se ejecuten más de una vez ya que podrían generarse inconsistencias en el sistema.

5.3. Mover_B.sh opción -c

Si bien en las especificaciones se incluye el comando 'Mover_B' se le ha agregado la opción para que permita copiar ciertos archivos en vez de moverlos. De este modo, con unas pequeñas modificaciones, utilizamos el mismo comando para la etapa de instalación del sistema.

5.4. Eliminar_B.sh

Dado que en ocasiones se necesita eliminar archivos, se construyo una función que suprime archivos a partir su ubicación. Adicionalmente se graba información sobre su uso en el Log.

6. Listado de Nuevos Archivos

6.1. LOGDIR/count

Este archivo es creado por Recibir_B. Tiene como propósito llevar la cuenta de la cantidad de veces que se ha ejecutado el ciclo 'Recibir'. Consta de un único campo, de la forma n=Nº de Ciclo.

6.2. GRUPO/instalacion.log

Este archivo guardará el log de la instalación. Como al momento de realizar la misma, aún no se encuentran definidos los directorios donde el usuario desea guardar sus datos, se usa este archivo especial para evitar dicho inconveniente.

6.3. CONFDIR/backup

Esta carpeta se crea en el momento de la instalación. En ella se copiarán todos los archivos binarios que precisa el sistema para funcionar hasta que el usuario decida dónde va a ubicarlos. Se trata de un directorio temporal, pues una vez que el usuario termina de realizar todas las configuraciones, es eliminado.

7. Listado de Datos

Para realizar la 'Prueba del camino Feliz' (ver sección 8) no sólo se han utilizado los datos provistos por la cátedra, sino que también se han agregado ciertos casos no contemplados por los mismos. Como por ejemplo reservas fuera de fecha, nombres de archivos inválidos, etc.

Para esta parte de las pruebas, los archivos `combos.dis`, `obras.mae` y `salas.mae` no han sido modificados.

A continuación, listamos los escenarios testeados y sus archivos asociados.

Archivo	Provisto	Caso testado
1-LEYER_DEMIAN@GMAIL.COM-EstreñoenelPayro332	SI	Obra disponible.
1-LEYER_DEMIAN@GMAIL.COM-EstreñoenelPayro333	SI	Obra no disponible.
1-MAIL_INVALIDO@GMAIL.COM-EstreñoenelPayro332	NO	Mail inválido.
147-LEYER_DEMIAN@GMAIL.COM-EstreñoenelPayro332	NO	ID obra inválido.
133-GRIFFO_JULIO@GMAIL.COM-muchaAnticipacion	NO	Reserva excede los 30 días.
NOMBRE_INVALIDO	NO	Formato de nombre incorrecto.
133-GRIFFO_JULIO@GMAIL.COM-vacio	NO	Archivo vacío.
6-Teatro.Corrientes.Azul@reservas.com.ar-TCA.3Nov13	SI	Reserva por sala. No disponible.
18-Teatro.Payro@reservas.com.ar-PAYro251013a	SI	Reserva por sala. Disponible.
18-MAIL_INVALIDO@GMAIL.COM-PAYro251013a	NO	Mail inválido.
180-Teatro.Payro@reservas.com.ar-ID.INVALIDO	NO	ID sala inválido.

8. Prueba del Camino Feliz

Para realizar la prueba de una ejecución normal, desde que se instala el sistema, hasta llegar al comando imprimir, se cuenta con un script que va realizando cada paso. La salida de cada comando, así como los archivos generados, se vuelcan en un archivo denominado 'caminoFeliz.out'