

Sistemas Operativos 75.08

grupo 03

ARCHIVO README

Integrantes:

Alfonso Javier Amezqueta 94732

1)Descarga:

link al repositorio:

https://github.com/alfo777/TP1_SO_GRUPO03

Hay dos formas de descargar el repositorio:

Forma 1:

En un directorio vacío abrir una terminal y tipear:

- git config --global user.name "Tu nombre de usuario de git"
- git config --global user.email "Tu mail de usuario de git"
- git init
- git remote add <Algun nombre para identificar al repositorio> <url_repositorio>
- git pull <nombre elegido> master

Forma 2:

Ir al repositorio a través del link y descargarlo. Ir a donde esta el paquete descargado, abrir una terminal y escribir:

unzip TP1_SO_GRUPO03-master.zip

Se creará un directorio **TP1_SO_GRUPO03-master**, con el directorio GRUPO03 adentro. Ahora tomar el directorio GRUPO03 y llevarlo a un directorio vacío, no dejarlo en **TP1_SO_GRUPO03-master**.

Con esto el repositorio ya esta descargado y se puede empezar la instalación.

2)Instalación:

Asi como esta ahora el sistema se lo considera en su estado inicial, o sea sin instalar. Para instalarlo ir al subdirectorio inst y abrir una terminal allí. En un principio el instalador no tiene permiso de ejecución luego de la descarga así que se los tiene que otorgar con el comando chmod así:

chmod 555 instalador.sh

Ahora le hemos dado al instalador permisos de lectura y de ejecución. Ya podemos instalar nuestro sistema.

Observación: El instalador, a excepción de la opción “-r”, no recibe parámetro. Cualquier parámetro que se le mande causa que se cierre el instalador.

Observación: Así como este estado inicial no debe ser roto quitando algún directorio o archivo dentro, ya que es necesario para que el instalador sepa si el sistema está instalado o no. Tampoco es recomendable agregarle algún directorio extra ya que podría perderse si el sistema se tiene que reinstalar.

Al principio no tenemos un sistema instalado, para iniciar la instalación en la terminal que tenemos en **inst** escribir:

./instalador.sh

Esto inicia la instalación. El primer paso a seguir es elegir el nombre de los directorios con los que vamos a trabajar. Son siete en total los directorios a crear, se mostrará en pantalla al usuario cual es el nombre por defecto que ese directorio va a tener y una breve descripción de cuál es su función en el sistema. Se le dará la opción de que el directorio conserve el nombre por defecto (Opción NO) o que lo cambie (Opción SI)

A tener en cuenta: siempre que se le da al usuario la posibilidad de intervenir en alguna decisión se le preguntará si quiere hacerlo. Se le da a elegir y el usuario tiene que tipear en mayúsculas SI o NO para validar su decisión. El instalador rechazará cualquier entrada que no sea SI o NO y se le pedirá que ingrese otra, así hasta que el usuario ingrese la opción correcta.

A tener en cuenta: Al principio tenemos una serie de nombres reservados que los directorios a crear no pueden tener, cualquier intento de asignarle a esos alguno de esos nombres se notificará al usuario como inválido y se le pedirá que ingrese otro. Los nombres reservados son: **inst**, **docs**, **pruebas**, **scripts**, **tablas_cpr** y **BACKUPS**.

A tener en cuenta: Durante la asignación de nombres también se le asignará a cada directorio un nombre por defecto que está reservado solo para él. Por ejemplo, si al directorio de ejecutables, que tiene el nombre **bin**, yo le quiero asignar el nombre **nov**, nombre por defecto del directorio de novedades, el instalador no me permite esta opción. Tampoco permite que haya nombres duplicados.

Luego de haber configurado el nombre de los directorios se le pedirá que defina la hora de cierre. Por defecto esta es las 180000 hrs, se le da al usuario la posibilidad de cambiarla (Opción SI), o no (Opción NO). La hora ingresada tiene que estar en formato HHMMSS y ser válida. Cualquier número que no cumpla esto será rechazado y se pedirá otro.

Ya configurados los directorios y la hora de cierre se le muestra al usuario como queda el sistema instalado y se le da la opción de crearlo o no. Si dice que NO entonces se le asigna a los directorios los nombres que el usuario les dio en la pasada anterior como nombres por defecto, lo mismo para la hora de cierre, y se repite la instalación. Si el usuario ingresa SI los directorios se crean y el sistema termina la instalación. Notar que el archivo de configuración ha sido creado y hay un archivo de log que ha registrado todos los pasos de la instalación.

Estado final luego de la instalación: luego de la instalación se tiene en el directorio grupo los siguientes subdirectorios:

- **inst:** Directorio de instalación, guarda el instalador, el archivo de log de instalación y el archivo de configuración.
- **Directorio de ejecutables (por defecto bin):** guarda los scripts de start, stop, inicializador y proceso.
- **Directorio de tablas (por defecto tab):** guarda las tres tablas del sistema.
- **Directorio de novedades (por defecto nov):** acá se depositan los archivos que se tienen que procesar.
- **Directorio de aceptados (por defecto ok):** es el directorio donde se procesaran los archivos de novedades aceptados.
- **Directorio de rechazados (por defecto nok):** es el directorio donde se mueven los archivos de novedades rechazados.
- **Directorio de procesados (por defecto proc):** el directorio donde se mueven los archivos de novedades aceptados que fueron procesados.
- **Directorio de salido (por defecto sal):** directorio donde se guardan los archivos de salido
- **BACKUPs:** es un directorio que se crea durante la instalación y allí se guarda el sistema

Reparación: la reparación está delegada en el script instalador. Los escenarios que nuestro sistema puede reparar son los siguientes:

- recuperación de cualquiera de los scripts que están en el subdirectorio **scripts**, siempre y cuando su copia de seguridad en BACKUPs este. Si esto último no es así, el sistema no se puede reparar y se le pedirá al usuario que recupere manualmente los scripts faltantes.
- recuperación de cualquiera de las tablas que están en el subdirectorio **tablas_cpr**, siempre y cuando su copia de seguridad en BACKUPs este. Si esto último no es así, el sistema no se puede reparar y se le pedirá al usuario que recupere manualmente los scripts faltantes.
- recuperación de cualquier directorio salvo **BACKUPs** e **inst**
- recuperación del archivo de configuración siempre y cuando su copia en la carpeta de **BACKUPs** exista. Si el archivo de configuración y su copia de seguridad no están, el sistema se reinstale.

Para restaurar el sistema abrir la terminal en la carpeta inst y tipear:

`./instalador.sh -r`

A tener en cuenta:

- Si el sistema ya se instaló y volvemos a correr el script entonces nos dice que ya se produjo la instalación, nos muestra el estado final de la instalación y el proceso instalador se cierra.
- Si el sistema ya se instaló, está roto y volvemos a correr el script entonces nos dice que hace falta repararlo y nos muestra como.
- Si el sistema ya se instaló, no está roto y volvemos a correr el script con la opción “-r” entonces nos dice que el sistema ya se reparo.
- Si el sistema no está instalado y corremos el script con la opción “-r” entonces nos dice que el sistema no se instaló.

reinstalación: La reparación va a reinstalar el sistema si no encuentra el archivo de configuración y su copia de seguridad. Cuando la reistalacion ocurra todos los directorios creados serán destruidos, salvo la copia de seguridad del sistema original que se descargó. Esta última se va a recuperar y se usará para lanzar una nueva instalacion identica a la efectuada anteriormente.

Sistema que no se puede reparar: Cuando alguno de los scripts que no es el instalador o alguna de las tablas desaparece de su respectivo directorio y del directorio que está en la copia de seguridad, entonces el sistema no se puede reparar. El usuario tendrá que descargarse el sistema devuelta o recuperar manualmente el script o la tabla a su directorio correspondiente.

3) Inicialización:

El inicializador tiene como función preparar nuestro sistema para poder lanzar el proceso.

Acciones que realiza el inicializador:

- El inicializador sabe si el sistema está o no instalado, si es el último caso entonces se cierra e informa al usuario de esta situación.
- El inicializador sabe si la instalación está rota, en este caso se cierra e informa al usuario como tiene que hacer para repararla.
- Si la instalación está completa el inicializador podrá funcionar con normalidad. Lo primero que hace es ver si todos los scripts y todas las tablas de nuestro sistema tienen los permisos adecuados, lectura para las tablas y lectura y ejecución para los scripts. Si algún elemento no tiene su permiso designado entonces lo corrige e informa al usuario de esto.
- Una vez verificada la instalación y los permisos el inicializador confirma que se lanzó el ambiente para la ejecución del proceso, lo cual nos permite usar el comando start. Se le va a preguntar al usuario si desea lanzar el proceso desde el inicializador (el

inicializador llama a start) o si se desea hacerlo manualmente, llamando a start por nuestra cuenta. Cualquiera de las dos opciones cierra al proceso.

- Por último el inicializador sabe que si el sistema ya se inicializó o no, así que volver a correrlo con el ambiente inicializado causa que el inicializador informe al usuario de esto.

Para lanzar el inicializador se tiene que abrir una terminal en el directorio de ejecutables y correr el script en foreground sin proceso hijo, así:

```
. ./inicializador.sh
```

observación: El inicializador y cualquiera de los otros scripts no reciben parámetros, si se les manda alguno los mismos se van a cerrar e informar al usuario de esto.

4) start:

Start se encarga de lanzar el proceso manualmente si el ambiente está levantado y si no hay otro proceso en ejecución. Además start sabe si el sistema está o no instalado correctamente, o si el ambiente se levantó o no. Cualquier escenario que impida a start poner en marcha al proceso será reportado al usuario. Si start se ejecuta con normalidad entonces el proceso será lanzado y se le informará al usuario cual es la id de proceso y la hora de cierre. Si hay otro proceso corriendo start lo informa y no hace nada.

Start se ejecuta de la misma forma que el inicializador en el directorio de ejecutables:

```
. ./start.sh
```

observación: start graba en el archivo de log del inicializador.

5) proceso.sh:

La funcionalidad del proceso se ajusta a lo pedido por el enunciado. Una vez lanzado el proceso ir exportando al directorio de novedades los archivos que se desean probar. Tener en cuenta que el proceso duerme un minuto en cada ciclo que encuentra al directorio de novedades vacío, así que el proceso no actuará sobre archivos que sean depositados en el directorio de aceptados directamente. En resumen, **el proceso solo actuará sobre archivos que estén en el directorio de novedades, mientras este esté vacío el proceso no actuará y dormirá hasta que haya efectuado todos los ciclos correspondientes.**

Una vez entregado los archivos de prueba al directorio de novedades los resultados pueden verse en el directorio de rechazados, en el de salida o en el archivo RejectedData del directorio de rechazados.

Observación: El proceso **solo** se puede lanzar con el script start o el inicializador, cualquier intento de lanzarlo manualmente en background produce que el proceso se cierre.

Pruebas que se pueden realizar con el proceso ejecutándose:

Recomendación: es recomendable que antes de realizar cada una de estas pruebas se vacíe los directorios de novedades, aceptados, salida, rechazados y procesados. De esta forma es más fácil identificar resultados en el archivo de salida.

- **Para probar que un archivo que tiene todos sus registros bien genera una salida correcta:**
 - 1) Ir al \$GRUPO/BACKUPS/pruebas/aceptados.
 - 2) Tomar cualquier archivo que esté ahí y copiarlo en el directorio de novedades.
 - 3) Confirmar que no fue enviado al archivo de rechazados.
 - 4) confirmar que no se creó el archivo RejectedData, o alguna entrada al mismo.
 - 5) confirmar que el archivo de novedades ahora está en el directorio de procesados y se creó una salida correcta.
- **Para probar que un archivo que ya fue procesado no se puede procesar de nuevo:**
 - 1) Luego de realizar la prueba anterior ir al directorio de procesador y copiar el archivo que se proceso.
 - 2) Pegar el archivo de novedades y confirmar que lo manda al directorio de rechazados.
- **Para probar que un archivo que tiene su nombre mal, un formato incorrecto, está vacío es enviado al directorio de rechazados:**
 - 1) Ir al \$GRUPO/BACKUPS/pruebas/aceptados.
 - 2) Tomar cualquier archivo que esté ahí y copiarlo en el directorio de novedades.
 - 3) Confirmar que no fue enviado al archivo de rechazados.
 - 4) Dentro del archivo, salvo que esté vacío, está explicada la razón de porque será rechazado.
- **Para probar que si la hora de cierre el menor a la hora de la transacción en algún registro la salida muestra el día que esta en nombre mas uno:**
 - 1) En un sistema cuya hora de cierre sea 180000 o menor.
 - 2) Ir al \$GRUPO/BACKUPS/pruebas/ajustar_fecha.
 - 3) Tomar el archivo que está allí y llevarlo al directorio de novedades
 - 4) Confirmar que la salida tienen 0202 en su nombre, el día fue cambiado correctamente.
 - 5) Con un sistema de
- **Para probar que hace de manera correcta el cambio de fin de mes en el archivo de salida:**
 - 1) Ir a \$GRUPO/BACKUPS/pruebas/fin de mes.
 - 2) Tomar todos los archivos que están ahí y llevarlos a novedades.
 - 3) Comprobar que en la salida el cambio de mes se ha efectuado para los archivos finales.

- **Para probar que se crea un archivo llamado RejectedData en rechazados si hay archivos con errores en sus registros:**
 - 1) Ir a \$GRUPO/BACKUPS/pruebas/aceptados_err_campos.
 - 2) Copiar algun archivo y llevarlo a novedades.
 - 3) Confirmar que el archivo se movió al directorio de procesados.
 - 4) Confirmar que se creó un archivo RejectedData en rechazados que indica cuáles registros se rechazaron y porque.

Aclaración: Hay tres motivos por los cuales un registro se puede rechazar.

- **Formato inválido:** un campo que tiene que tener números tiene caracteres o viceversa.
- **Falta campo:** Algunos de los 15 campos falta.
- Hay más de 15 campos en el registro.

Por último se puede revisar en el archivo log que todo esto se realizó bien.

6) stop:

El stop tiene como función detener al proceso antes de que alcance los 10000 ciclos. Se tiene que ejecutar al igual que el proceso y el start en foreground y sin proceso hijo, si no hay proceso corriendo entonces informa de esto y no hace nada.

Para ejecutarlo tipear en la terminal que corre al proceso:

./stop.sh