Primera PED

Introducción

En este trabajo he implementado, siguiendo el enunciado, los scripts Demonio.sh y Fausto.sh. La finalidad de este programa es manejar procesos y listas de procesos de manera automatizada y periódica.

Implementación

Demonio.sh

Este script es responsable de controlar el ciclo de vida de los procesos y las listas asociadas. Tiene un bucle principal que se ejecuta continuamente hasta que se activa el “Apocalipsis”.

Dentro del bucle, verifica la presencia del archivo “Apocalipsis”. Si se detecta, realiza las siguientes acciones:

- Elimina todos los procesos de las listas asociadas: procesos, procesos_servicio y procesos_periodicos.
- Borra archivos y directorios no esenciales según los criterios del enunciado. No se eliminan los ficheros “test” para posibilitar la lectura de los resultados una vez haya concluido el “Apocalipsis”.
- Elimina el Demonio y todos los procesos relacionados con él, incluidos los asociados al proceso Fausto.

Si no se detecta la presencia del fichero “Apocalipsis”, continúa ejecutando el cuerpo del bucle y verifica y revive los procesos que figuran en la lista “procesos_servicio” si no están en ejecución, lo que quiere decir que no ha sido Fausto quien los ha eliminado.

Por último, verifica los tiempos de los procesos periódicos y lanza sus comandos de ser necesario con la función lanzamiento_periodico.

Para evitar el acceso simultáneo a las listas por parte de los dos scripts, Demonio.sh hace uso de la opción -x del comando flock utilizando el fichero SanPedro como llave.

Fausto.sh

Este script proporciona una interfaz para manejar los procesos. Si se ejecuta sin argumentos, verifica la existencia del Demonio, y si no está en ejecución, lo inicia. Implementa los siguientes comandos:

- **run** -> ejecuta un comando una sola vez.
- **Run-service** -> ejecuta un comando como un servicio.
- **Run-periodic** -> ejecuta un comando de manera periódica.
- **list** -> lista los procesos clasificados por: procesos, procesos servicio y procesos periódicos.

- **help** -> muestra la sintaxis y opciones disponibles.
- **stop** -> detiene un proceso proporcionando su PID
- **end** -> activa el “Apocalipsis”, desencadenando una serie de acciones controladas por el Demonio.

Al trabajar con listas, se realizan verificaciones y operaciones seguras, utilizando bloqueos para evitar conflictos con el Demonio.

Ejecución de ejemplo

```
**********************************************************
1) Debería de haberse creado el proceso Demonio
la salida esperada es algo así
systemd——systemd——Demonio.sh——sleep
donde Demonio.sh no debe ser hijo de bash sino de systemd o similar
**********************************************************

systemd——Demonio.sh——sleep

**********************************************************
2) Lanzo algunos comandos y compruebo que se han creado
Debería de haber un proceso normal
'sleep 10; echo hola > test1.txt'
Un proceso servicio 'yes > /dev/null'
y dos periódicos, el normal y el lento
Comparamos los procesos teóricamente lanzados y los que
realmente existen
**********************************************************

Procesos lanzados según Fausto:
./Fausto.sh list
***** Procesos normales *****
50612 sleep 10; echo hola >> test1.txt
***** Procesos servicio *****
50628 yes > /dev/null
***** Procesos periodicos *****
0 5 pid echo hola_periodico >> test2.txt
0 5 pid echo hola_periodico_lento >> test3.txt; sleep 20

Procesos existentes:
ps -l
F S   UID     PID    PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
0 S  1000    2241    2238  0  80   0 -  4290 sigsus pts/0    00:00:05 zsh
0 S  1000   50585    2241  0  80   0 -  2339 do_wai pts/0    00:00:00 Ejercicio1.sh
0 S  1000   50604       1  0  80   0 -  2339 do_wai pts/0    00:00:00 Demonio.sh
0 S  1000   50608   50604  0  80   0 -  1932 hrtime pts/0    00:00:00 sleep
0 S  1000   50612       1  0  80   0 -  2339 do_wai pts/0    00:00:00 bash
0 S  1000   50615   50612  0  80   0 -  1932 hrtime pts/0    00:00:00 sleep
```

```
0 S  1000   50628      1  0  80   0 -  2339 do_wai pts/0    00:00:00 bash
0 R  1000   50631   50628 99  80   0 -  1932 -      pts/0    00:00:00 yes
0 R  1000   50651   50585  0  80   0 -  3401 -      pts/0    00:00:00 ps
```

****
3) Elimino manualmente el proceso yes sin avisar a Fausto.
El Demonio debería detectarlo y reiniciar el proceso:
****

pkill yes
Terminado

./Fausto.sh list
***** Procesos normales *****
50612 sleep 10; echo hola >> test1.txt
***** Procesos servicio *****
50676 yes > /dev/null
***** Procesos periodicos *****
3 5 50686 echo hola_periodico >> test2.txt
3 5 50695 echo hola_periodico_lento >> test3.txt; sleep 20

****
4) Elimino el proceso usando Fausto.
El Demonio NO debe reiniciar el proceso:
****

./Fausto.sh stop 50676
./Fausto.sh list
***** Procesos normales *****
50612 sleep 10; echo hola >> test1.txt
***** Procesos servicio *****
***** Procesos periodicos *****
6 5 50814 echo hola_periodico >> test2.txt
6 5 50823 echo hola_periodico_lento >> test3.txt; sleep 20
ps -l
```
F S   UID     PID    PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
0 S  1000    2241    2238  0  80   0 -  4290 sigsus pts/0    00:00:05 zsh
0 S  1000   50585    2241  0  80   0 -  2339 do_wai pts/0    00:00:00 Ejercicio1.sh
0 S  1000   50604       1  0  80   0 -  2483 do_wai pts/0    00:00:00 Demonio.sh
0 S  1000   50612       1  0  80   0 -  2339 do_wai pts/0    00:00:00 bash
0 S  1000   50615   50612  0  80   0 -  1932 hrtime pts/0    00:00:00 sleep
1 S  1000   50695   50604  0  80   0 -  2450 do_wai pts/0    00:00:00 Demonio.sh
0 S  1000   50698   50695  0  80   0 -  1932 hrtime pts/0    00:00:00 sleep
1 S  1000   50823   50604  0  80   0 -  2450 do_wai pts/0    00:00:00 Demonio.sh
0 S  1000   50825   50604  0  80   0 -  1932 hrtime pts/0    00:00:00 sleep
0 S  1000   50826   50823  0  80   0 -  1932 hrtime pts/0    00:00:00 sleep
0 R  1000   50828   50585  0  80   0 -  3401 -      pts/0    00:00:00 ps
```

```
************************************************************
5) Error de sintaxis provocado para que Fausto nos avise:
************************************************************

./Fausto.sh asdf
Error, orden 'asdf' no reconocido, consulte las ordenes disponibles con ./Fausto.sh help

************************************************************
6) Siguiendo la sugerencia anterior veríamos la ayuda:
************************************************************

./Fausto.sh help
Sintaxis:
 ./Fausto.sh run comando
 ./Fausto.sh run-service comando
 ./Fausto.sh run-periodic T comando
 ./Fausto.sh list
 ./Fausto.sh help
 ./Fausto.sh stop PID
 ./Fausto.sh end

************************************************************
7) Terminamos la ejecución y vemos los mensajes enviados
por los procesos lanzados en los ficheros test 1, 2 y 3
************************************************************

Terminado
hola
hola_periodico
hola_periodico
hola_periodico_lento
hola_periodico_lento

************************************************************
8) Comprobamos que no hay procesos sin terminar.
Esperamos que sólo salgan bash, Ejercicio1.sh y ps
************************************************************

  PID TTY          TIME CMD
 2241 pts/0    00:00:05 zsh
50585 pts/0    00:00:00 Ejercicio1.sh
50922 pts/0    00:00:00 ps
```

```
************************************************************
9) Comprobamos que no hay ficerhos basura.
Solo deben quedar Fausto.sh, Demonio.sh y la Biblia.txt
************************************************************

Biblia.txt  Demonio.sh  Fausto.sh

************************************************************
10) Comprobamos que no hay bloqueos pendientes
no debería de salir nada:
************************************************************


************************************************************
11) Finalmente mostramos la Biblia
************************************************************

15:20:46: ----------Genesis----------
15:20:46: El demonio ha sido creado.
15:20:46: El proceso 50612 sleep 10; echo hola >> test1.txt ha nacido
15:20:46: El proceso 50628 yes > /dev/null ha nacido
15:20:46: El proceso echo hola_periodico >> test2.txt ha nacido
15:20:46: El proceso echo hola_periodico_lento >> test3.txt; sleep 20 ha nacido
15:20:47: El proceso 50628 resucita con pid 50676
15:20:47: El proceso pid se reencarna con pid 50686
15:20:47: El proceso pid se reencarna con pid 50695
15:20:49: El proceso 50676 ha terminado
15:20:52: El proceso 50686 se reencarna con pid 50814
15:20:52: El proceso 50695 se reencarna con pid 50823
15:20:56: ----------Apocalipsis----------
15:20:56: El proceso sleep 10; echo hola >> test1.txt ha terminado
15:20:56: El proceso echo hola_periodico >> test2.txt ha terminado
15:20:56: El proceso echo hola_periodico_lento >> test3.txt; sleep 20 ha terminado
```

Consideraciones y mejoras

El programa ha sido diseñado para manejar procesos y listas de manera eficiente. Se utiliza un sistema de bloqueos para garantizar la integridad de las listas y prevenir condiciones de carrera. Sin embargo, podría mejorarse implementando un manejo más robusto de errores y una gestión más detallada de los procesos. La estructura del código podría optimizarse mediante la reducción de bloques repetitivos y la reutilización de funciones.