

Practico

lunes, 15 de julio de 2024 01:21

6. Se sabe que la orden `equal archivo1 archivo2`, que NO usa su entrada, envía a su salida las palabras del `archivo1` que aparecen en el `archivo2`.

También, la orden `ls dir`, que NO usa su entrada, envía a su salida los nombres de los archivos que hay en el `dir` especificado. Por ejemplo, si hacemos `ls \` se obtiene como salida, (solo) los nombres de los archivos que están en el directorio raíz.

Sabiendo esto, escriba órdenes (puede ser más de una), que envíe a su salida los nombres de los archivos comunes del directorio actual (i.e. el `Dir` donde se está "parado") y de su directorio padre.

En su respuesta, explique con un comentario, que hace cada orden escrita.

Equal archivo 1 archivo 2 = hola
Ls dir
Archvohola1
Archivo2
Ls ..
Holaupa
Progra

Archivo 1=
Holacomoes

Archivo 2=
asdklhola

Ls dir > temp.txt
Ls .. > temp2.txt
Equal temp temp2
Archvohola1
Archivo2
Holaupa

7. Una APP llamada "command.exe", es capaz de ejecutar un comando del Shell, enviando las palabras de su entrada, como parámetros de ese comando.

Por ejemplo: Asumiendo que la entrada de `command.exe` tiene las palabras "uagrm.edu.bo" "google.com" y escribimos:

command chrome
↓
esta orden, genera y ejecuta esta otra.
chrome uagrm.edu.bo google.com

Otra APP llamada "cl.exe", que no usa parámetros, es capaz de enviar a su salida la cantidad de líneas que hay en su entrada.

Escriba UNA (sola) orden, cuya salida sea la cantidad total de líneas que suman todos los archivos del directorio actual. Explique su solución.

Si no explica la solución, su respuesta no será evaluada.

Tome en cuenta que "`dir /B /A-D`", envía a su salida (solo) los nombres de los archivos del directorio actual. También la orden "`type archivo1 archivo2 ... archivoN`", envía a su salida los contenidos de los archivos especificados como parámetros.

Dir /b/a-d | command type | cl.exe



8. La orden **diffLines** **archivo1** **archivo2**, envía a su salida las líneas que NO aparecen simultáneamente en ambos archivos especificados.

El comando **killAll** es capaz de “matar” a todos los procesos, cuyos nombres de ejecutables (archivos .exe) están anotados en su entrada.

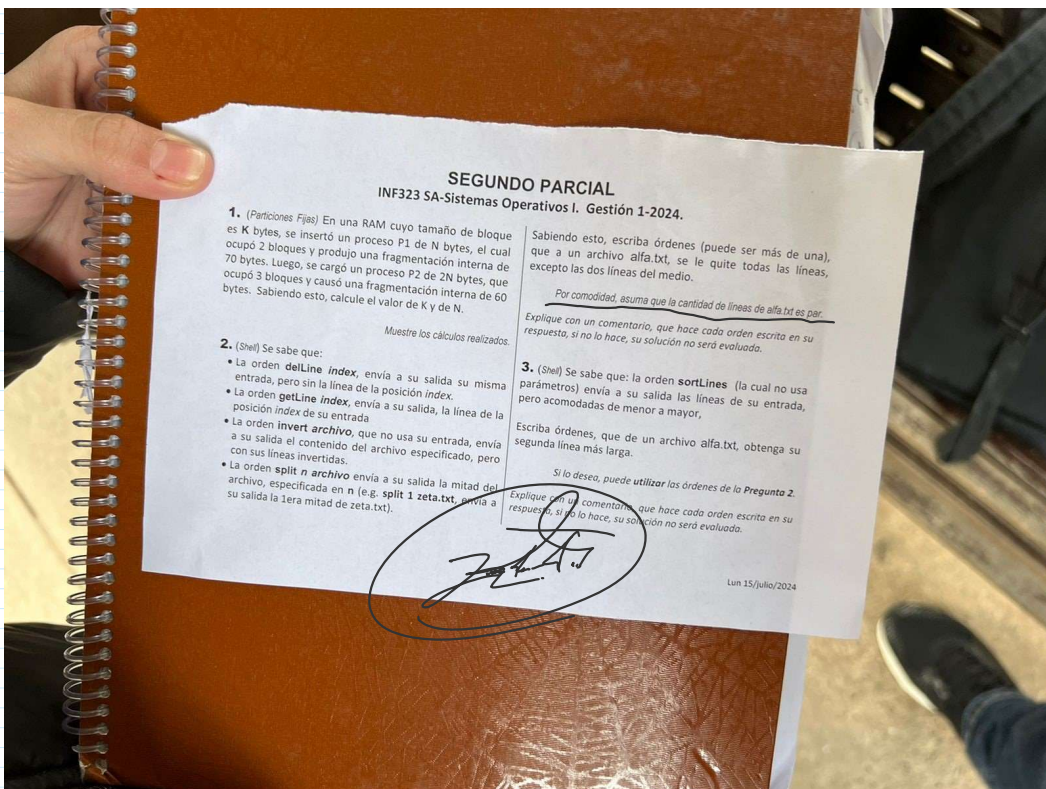
El comando **taskList**, envía a su salida los nombres de los procesos que están corriendo actualmente en el sistema.

La orden **dir *.exe /A-D**, envía a su salida (solo) los nombres, con extensión .exe, de los archivos que están en el directorio actual (i.e. el **dir** donde se está “parado”).

Sabiendo esto, escriba órdenes Shell (puede ser más de una), las cuales maten a todos los procesos, cuyos archivos ejecutables (archivos .exe) NO están en el directorio actual.

No se olvide de explicar, con un comentario, que hace cada una de sus órdenes.

```
Dir *.exe /A-D > temp.txt
Tasklist > procesos.txt
DiffLines temp.txt procesos.txt > prograkill.txt
Killall < progra.txt
```



```
sortLines alfa.txt > temp.txt
Invert temp.txt
Getline 2 < temp.txt > alfa.txt
```

2