FSO
PRÀCTICA 1

Antonio Torres Cabero
Eduard Bel
Cristina Izquierdo Lozano

Índex

Enunciat resumit

L'objectiu és llegir fitxers de log i guardar les dades a una base de dades per tal de facilitar l'anàlisi de seguretat. Per a dur a terme això, es farà un programa en python3 tenint en compte diverses codificacion, parts binàries i compressions dels fitxers.

Input de l'usuari (en tkinter): fitxer de log a tractar i fitxer BD on guardar el resultat
Interfície gràfica: consultes de l'usuari
Condicions:

- fitxer DB existeix: nous registres s'afegeixen sense tenir en compte la possibilitat de que ja hi fossin.
- fitxer DB no existeix: es crea un de nou "_ddmmaa"

Output: missatge amb el path, nom del fitxer i els permisos d'accés

Importació

Formats:

1. 7 camps → mes dia hora nom-maquina dimoni[PID]: missatge
2. 6 camps → mes dia hora nom-maquina proces: missatge

Comprovacions: la línia ha de tenir aquest format i cada camp ha de tenir també el format esperat per a poder ser inserit a la BD. Si no, es descarta i s'avisa amb el stderr.

Simplificacions:

- BD no relacional, sense cap camp clau ni cap restricció. Podria arribar a haver-hi registres duplicats.
- Per inserir els missatges a la BD alguns caràcters poden trencar la sintaxi SQL → tractar-los com a URLs.

Filtratge de registres

Funcions de botons:

- **Mes** → filtrar els registres de la BD que el seu mes sigui el que l'usuari hagi escrit al camp *Cerca*.
    - *Comprovacions*: format correcte.
    - *Resultats:*
        - Listbox corresponent per tal que es mostrin a la GUI. El format usat serà exactament el del fitxer de log.
        - Actualitzar la *línia d'estat* mostrant quants elements s'han trobat que compleixin el criteri del query.
- **Tots** → mostrar tots els registres de la BD
- **Exporta** → mostrar tots els registres pel stdout amb el format inicial dels logs.

Exemple de comanda sobre la BD: sqlcmd="SELECT camp2,camp3 FROM taula WHERE camp1 LIKE ‘%buscat%’"

Pseudocodi de les funcions

buscaMes()

```
funcio buscaMes()
mes_par := llegir();
mostra := executar("SELECCIONAR esdeveniments on mes := mes_par);
   j := 0;
   per cada i en mostra fer
       listbox := insertar(i);
       j := j + 1;
   fi per
   estat := escriure("Elements filtrats: " + string(j));
 fi funcio
```

buscaData()

```
funcio buscaData()
dia_par := llegir();
mostra := executar("SELECCIONAR esdeveniments on dia := dia_par);
   j := 0;
   per cada i en mostra fer
       listbox := insertar(i);
       j := j + 1;
   fi per
   estat := escriure("Elements filtrats: " + string(j));
 fi funcio
```

buscaMaquina()

```
funcio buscaMaquina()
   maquina_par := llegir();
   mostra := executar("SELECCIONAR esdeveniments on maquina :=
maquina_par);
   j := 0;
   per cada i en mostra fer
       listbox := insertar(i);
       j := j + 1;
   fi per
```

```
   estat := escriure("Elements filtrats: " + string(j));
 fi funcio
```

buscaProces()

```
funcio buscaProces()
   proces_par := llegir();
   mostra := executar("SELECCIONAR esdeveniments on proces :=
proces_par);
   j := 0;
   per cada i en mostra fer
       listbox := insertar(i);
       j := j + 1;
   fi per
   estat := escriure("Elements filtrats: " + string(j));
 fi funcio
```

neteja()

```
funcio neteja()
   listbox := eliminar(inici, fi);
   estat := escriure("Elements filtrats: "+"0");
 fi funcio
```

buscaPID()

```
funcio buscaPID()
   pid_par := llegir();
   mostra := executar("SELECCIONAR esdeveniments on pid := pid_par);
   j := 0;
   per cada i en mostra fer
       listbox := insertar(i);
       j := j + 1;
   fi per
   estat := escriure("Elements filtrats: " + string(j));
 fi funcio
```

buscaTots()

```
funcio buscaTots()
   mostra := executar("SELECCIONAR esdeveniments");
```

```
   j := 0;
   per cada i en mostra fer
       listbox := insertar(i);
       j := j + 1;
   fi per
   estat := escriure("Elements filtrats: "+string(j))
 fi funcio
```

acabar()

```
funcio acabar()
   tancaGUI()
 fi funcio
```

exportaFiltrats()

```
funcio exportaFiltrats()
   per_exportar := llegir_listbox('inici', 'fi');
   fitxer_base := obrir(fitxer_bd, "afegir");
   per cada i en per_exportar
       fitxer_base := escriure(string(i));
   fi per
   fitxer_base := tanca();
   messagebox := mostrar_info(titol="Atenció", missatge="Logs exportats
correctament al fitxer de DB");
 fi funcio
```

comprovaLogs()

```
funcio comprovaLogs(fitxer_logs, cursor)
   si fitxer_logs esta comprimit fer
       f := obrir_zip(fitxer_logs,'llegir binari');
   fsi
   sino fer
       f := obrir(fitxer_logs, "llegir binari");
   fsino
   per cada i en f
       correcte := Cert;
       tipus_codif := detectar_codificacio(i);
       i := decodificar(tipus_codif);
       llista := separar_camps(llista, " ");
```

```
        mes := llista[0];
        dia := llista[1];
        temps := llista[2];
        temps := separar_camps(temps, ":");
        hora := temps[0];
        minuts := temps[1];
        segons := temps[2];
        correcte = comprova_data(mes, dia, hora, minuts, segons)
        si correcte fer
            nom_maquina := llista[3];
            proces := llista[4];
            te_pid := buscar(proces, "[");
            si te_pid>=0 fer
                j := 5;
                proces := separar_camps(proces, "[");
                pid := "[" + proces[1];
                proces := proces[0];
             fsi
            sino fer
                j := 4;
                pid := ""  $descartar pid
             fsino
             missatge := "";
            mentre j < longitud(llista) fer
                missatge := missatge + " " + llista[j];
                j := j + 1;
             fmentre
            sql_insert := ''' INSERTAR A esdeveniments (mes, dia, hora,
maquina, proces, pid, missatge) ''';
            llista_mom := [mes, dia, llista[2], nom_maq, proces, pid,
missatge];
            executar(sql_insert, llista_mom);
         fsi
        sino fer
            escriu("Error de format);
        fsino
    f := tancar();
 fi funcio
```

comprova_data()

```
funcio comprova_data(m, d, hora, minuto, sec):
   correcte := Cert;
   mesos := ['Gen', 'Ene', 'Jan', 'Feb', 'Mar', 'Abr', 'Apr', 'Mai',
'May', 'Jun', 'Jul', 'Ago', 'Aug', 'Set', 'Sep', 'Oct', 'Nov', 'Dec',
'Dic', 'gen', 'ene', 'jan', 'feb', 'mar', 'abr', 'apr', 'mai', 'may',
'jun', 'jul', 'ago', 'aug', 'set', 'sep', 'oct', 'nov', 'dec', 'dic'];

   si (minuto < 0 o minuto >= 60) fer
       retorna Fals;
   fsi
   si (sec < 0 o sec >= 60) fer
       retorna Fals;
   fsi
   si (hora < 0 o hora >= 24) fer
       retorna Fals
   fsi
   si (longitud(m) != 3) fer
       retorna Fals;
   fsi
   si (d <= 0 o d >= 31) fer
       retorna Fals;
   fsi
   per cada i en mesos fer
       si m == i fer
           retorna correcte;
        fsi
    fper
   retorna Fals
```

Funcions i mètodes

Descodificació

Aquesta llibreria, juntament amb el mètode decode(), l'hem utilitzat per a la descodificació del fitxer logs. Exemple d'ús:

«chardet.detect Python Example». Consulta 11 març 2020.
https://www.programcreek.com/python/example/91538/chardet.detect.

Llibreria chardet

«chardet — chardet 3.0.4 documentation». Consulta 11 març de 2020.
https://chardet.readthedocs.io/en/latest/index.html.

Definició: Detecció automàtica de la codificació dels caràcters a Python.

Funcionament: pren una seqüència d'octets en una codificació de caràcters desconeguda i intentar determinar la codificació per a que es pugui llegir el text. És com agafar un codi quan no tenim la clau de desxifrat.

Ús: detect() → Detecta quina codificació té la cadena de caràcters (string).

Mètode decode()

«Python String decode() Method - Tutorialspoint». Consulta 11 març 2020.
https://www.tutorialspoint.com/python/string_decode.htm.

Definició: descodifica la cadena (string) mitjançant el codec registrat per a la codificació. Per defecte descodifica segons la codificació predeterminada dels string.

Funcionament: rep per paràmetre el tipus de codificació i el mode de maneig d'errors, per defecte es 'strict', és a dir, que retorna el UnicodeError.

Ús: decode() → descodifica segons el tipus de codificació detectada.

Joc de proves

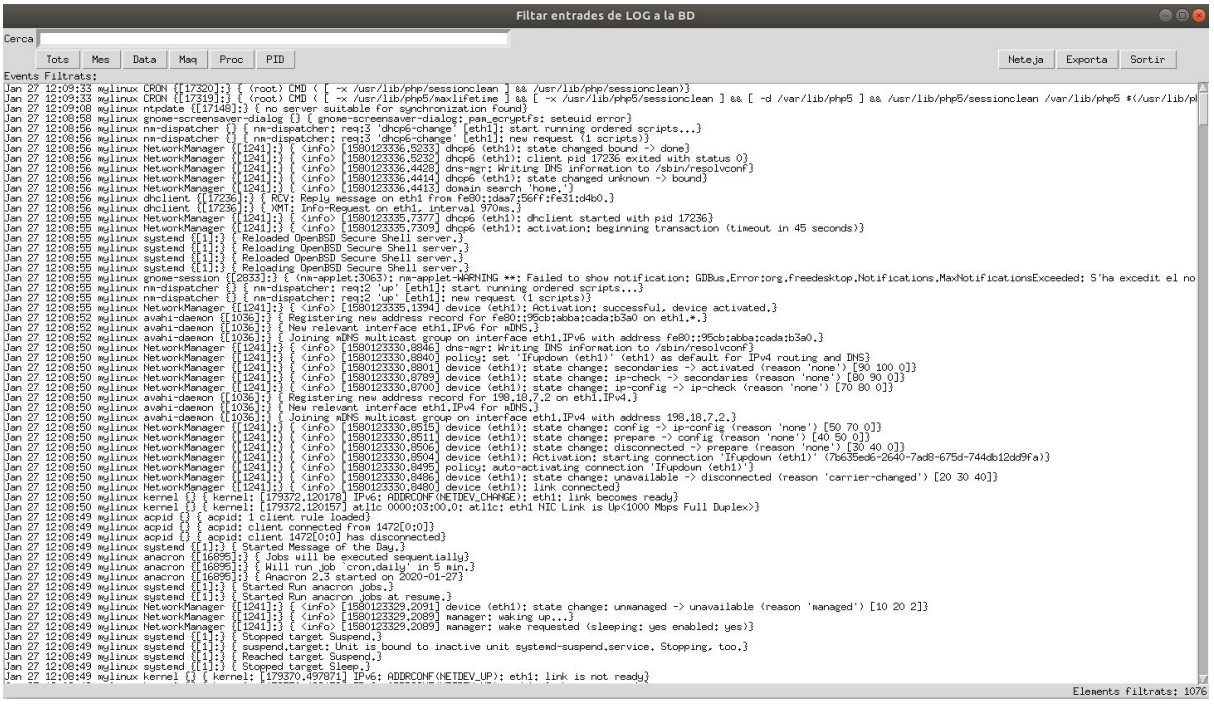
Descripció de les proves fetes, quins fitxers s'han obtingut i explicació de la seva casuística.

Bàsica

Yes Tot correcte	Ens farà escollir un fitxer de logs, si el fitxer és correcte podem utilitzar el programa
Yes Fitxer logs incorrecte	El programa es tancarà.
No Tot correcte	Escollirem el fitxer de logs i després ens demanarà la base de dades.
No Fitxers incorrectes	El programa es tancarà.

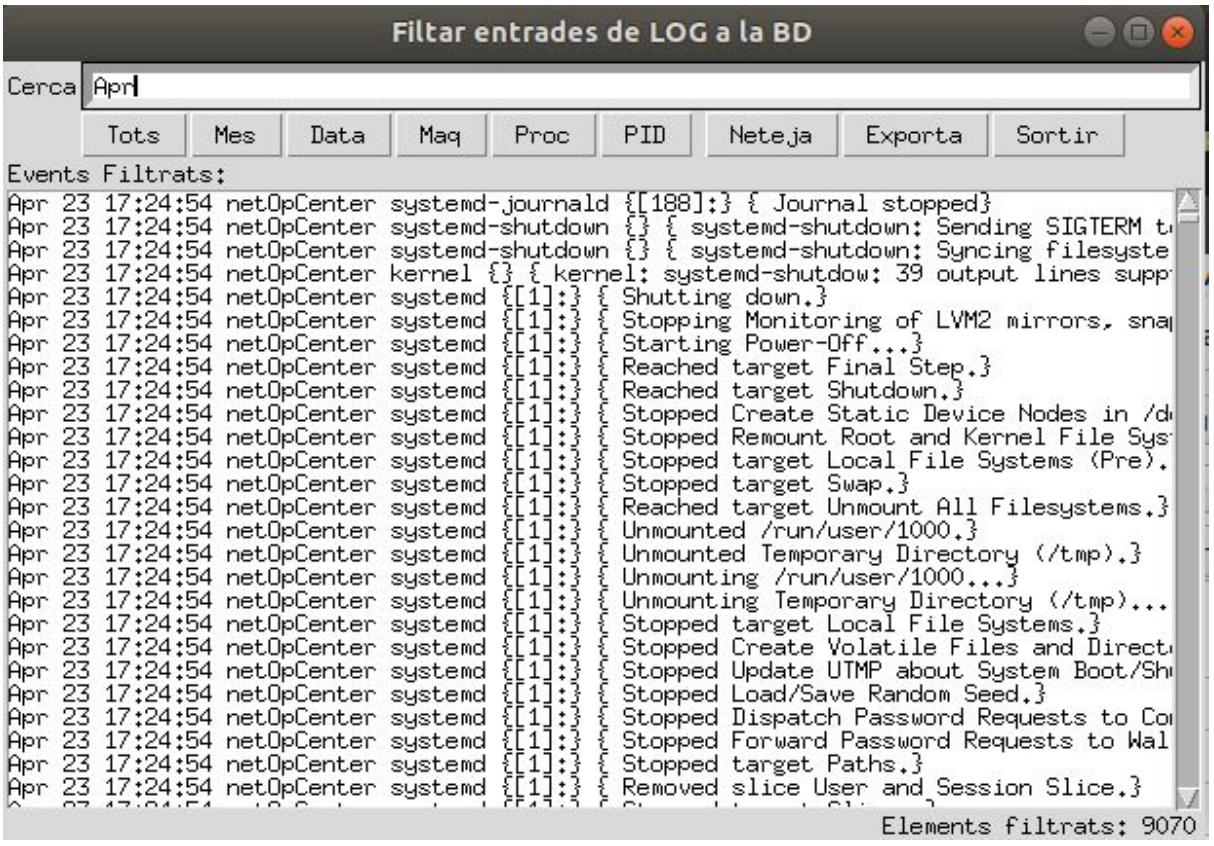
Escollim el fitxer syslog.1 per tal de fer una prova del funcionament bàsic del programa.

1. Botó “Tots”
   Mostra els missatges amb el format correcte i printa per pantalla els erronis.

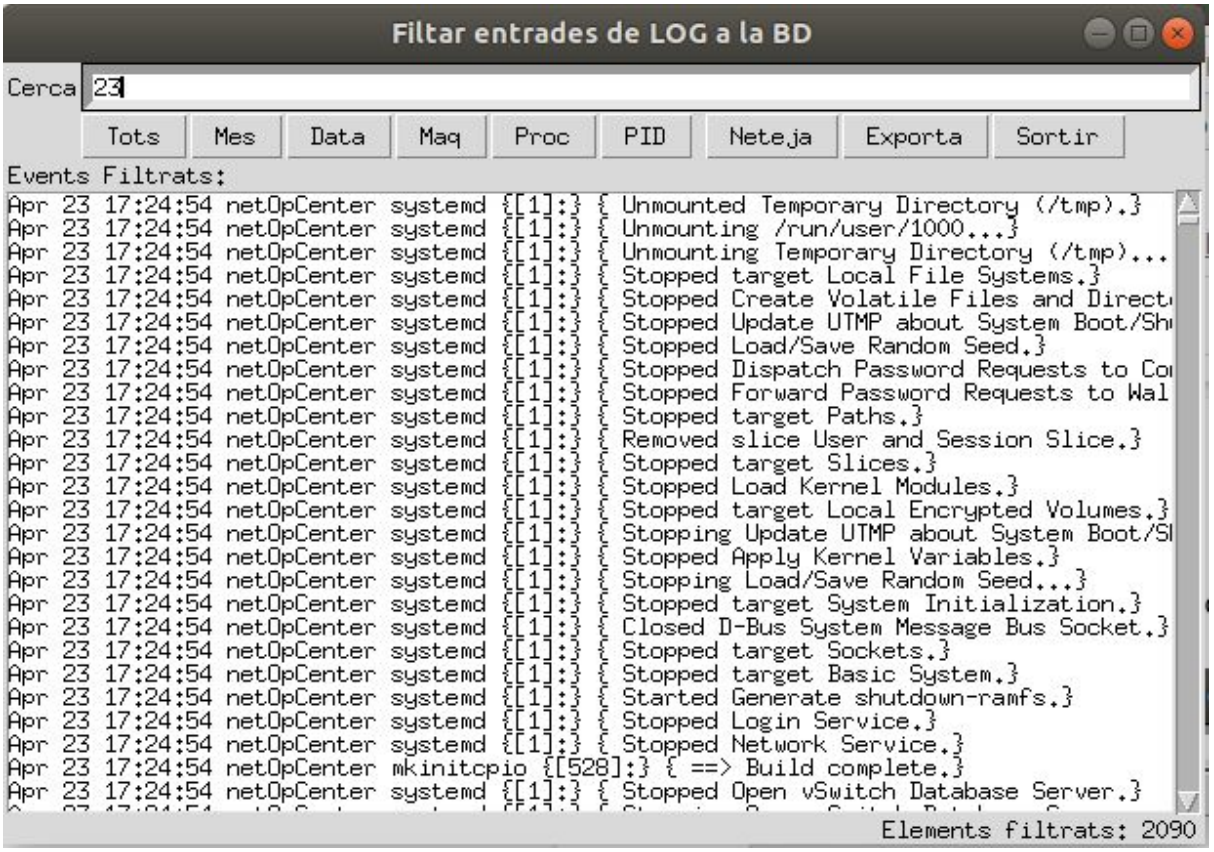


2. Botó “Mes”

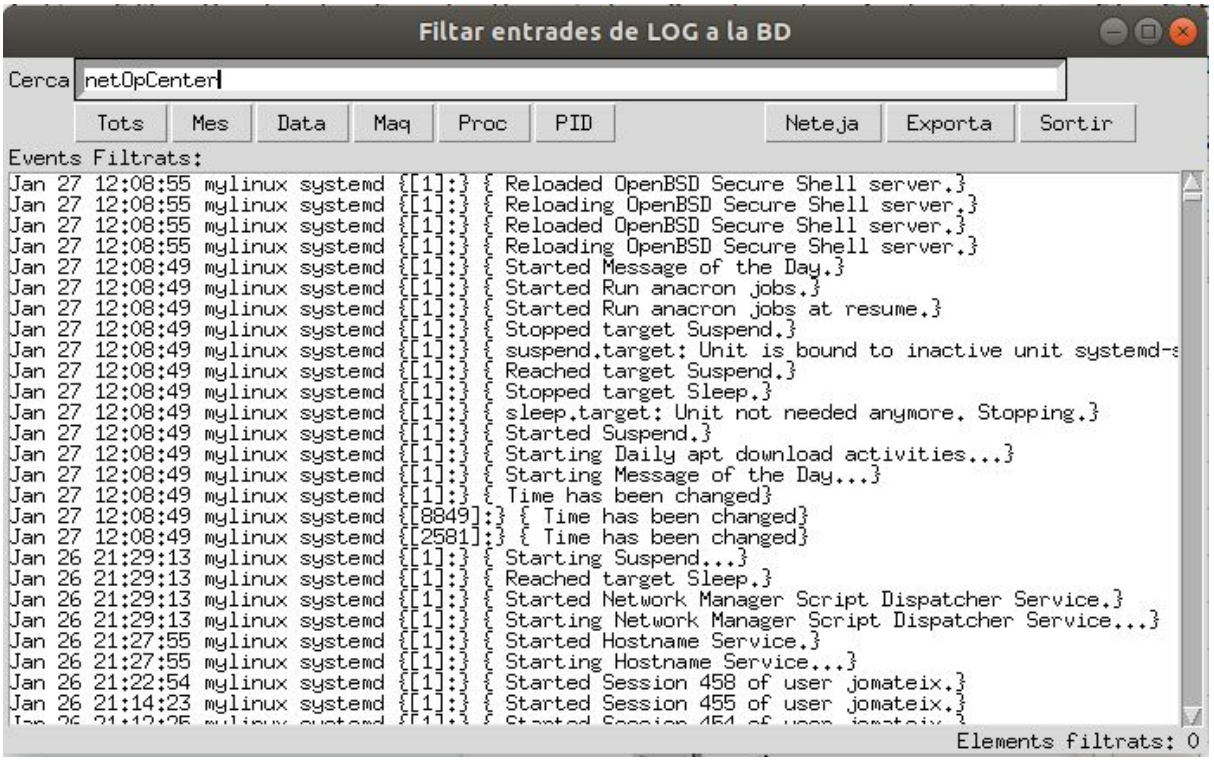
Per a demostrar aquest cas hem fet servir els fitxers de log comprimits.



3. Botó Data

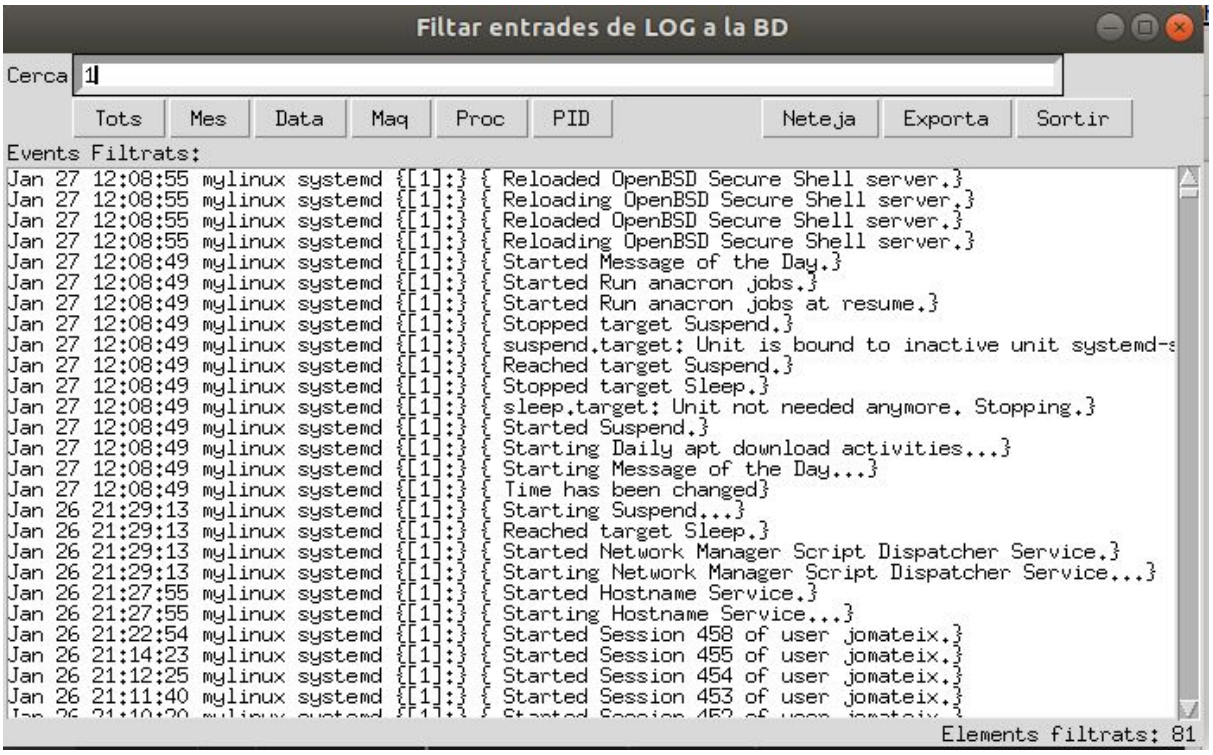


4. Botó Maq

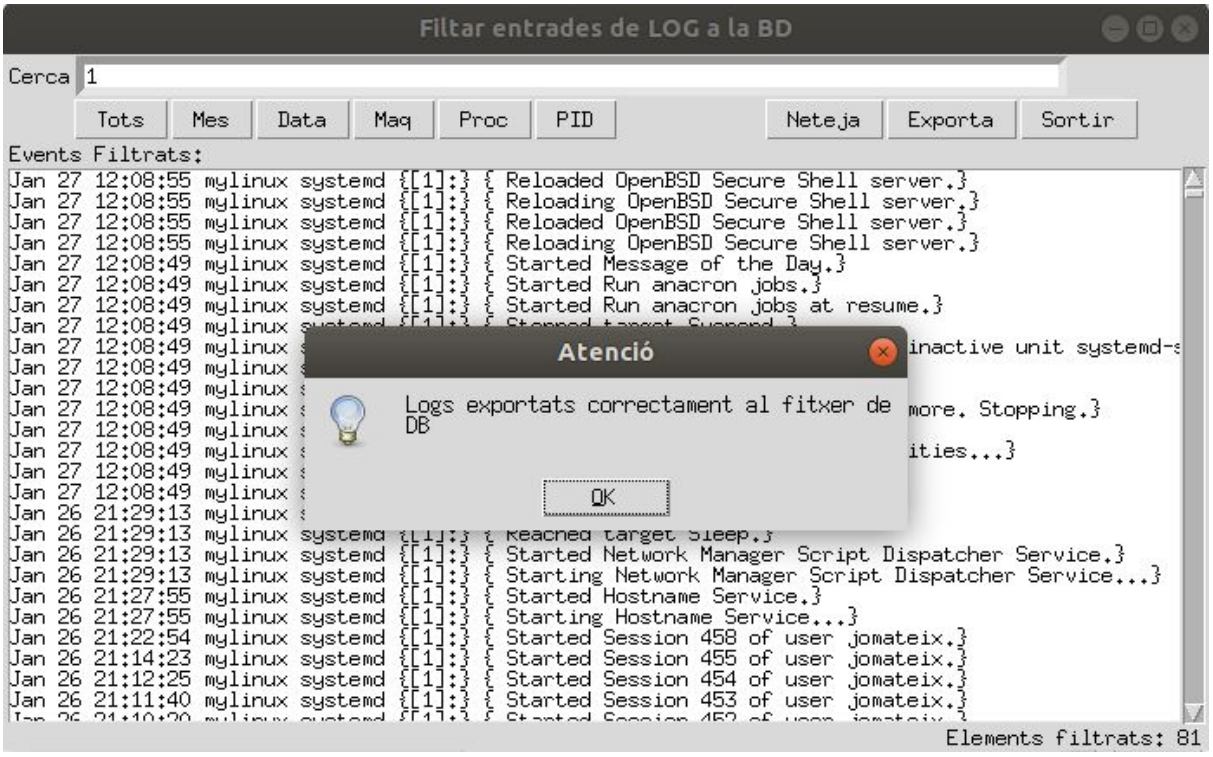


5. Botó Proc

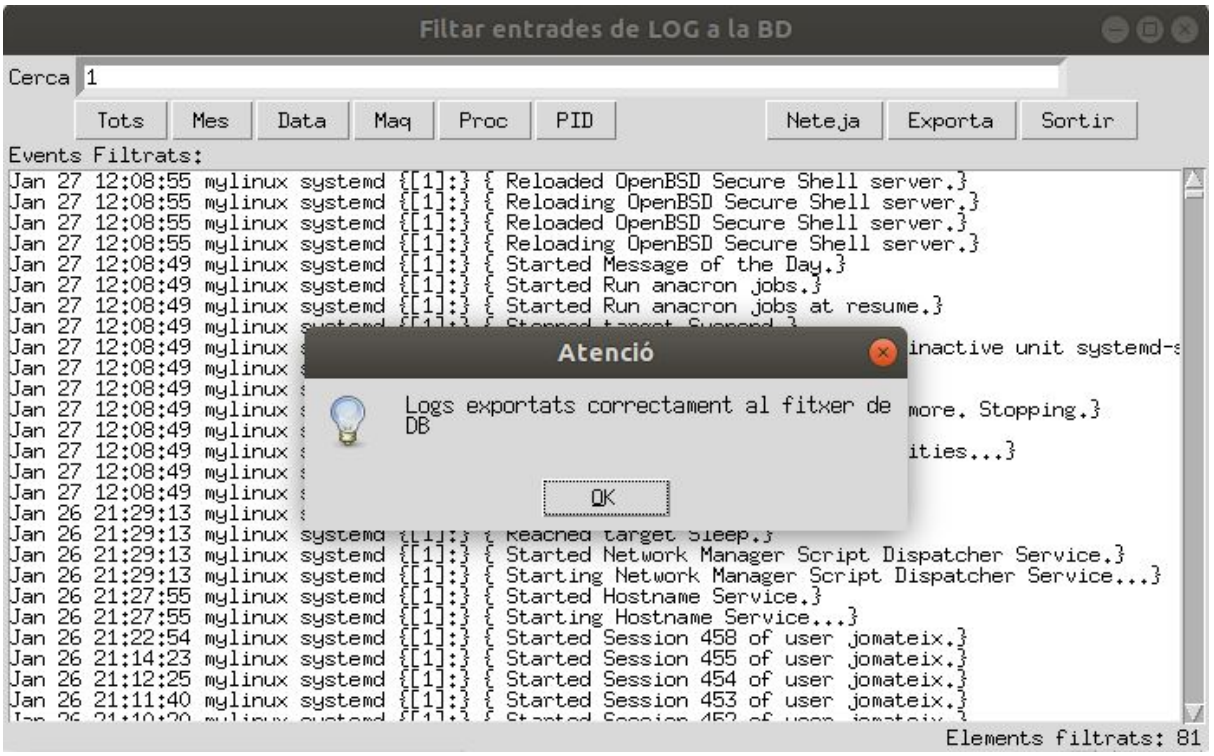
6. Boto PID

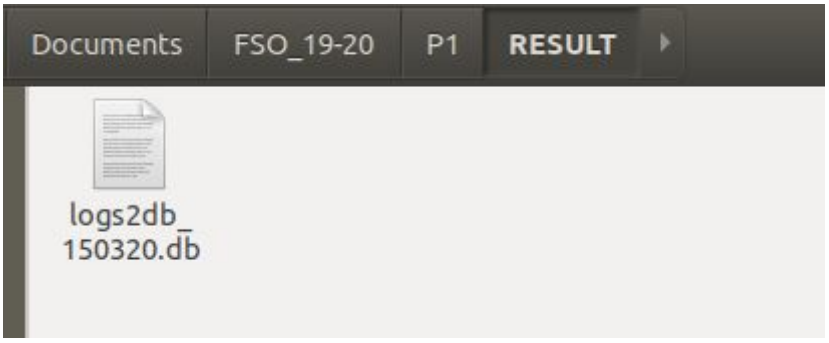


7. Botó Neteja

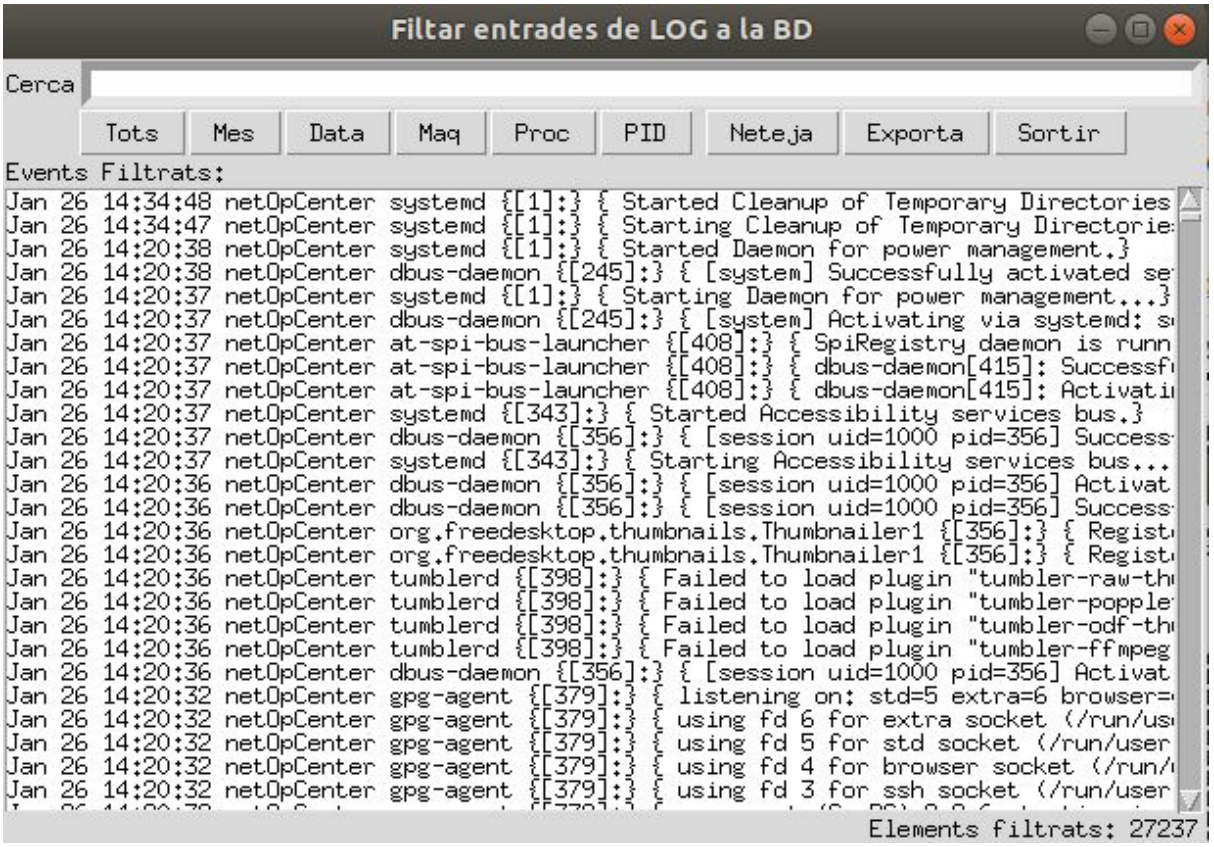


8. Botó Exporta


Filtar entrades de LOG a la BD
Cerca
1
Tots
Mes
Data
Maq
Proc
PID
Neteja
Exporta
Sortir
Events Filtrats:
Jan 27 12:08:55 mylinux systemd {[1]:} { Reloaded OpenBSD Secure Shell server.}
Jan 27 12:08:55 mylinux systemd {[1]:} { Reloading OpenBSD Secure Shell server.}
Jan 27 12:08:55 mylinux systemd {[1]:} { Reloaded OpenBSD Secure Shell server.}
Jan 27 12:08:55 mylinux systemd {[1]:} { Reloading OpenBSD Secure Shell server.}
Jan 27 12:08:49 mylinux systemd {[1]:} { Started Message of the Day.}
Jan 27 12:08:49 mylinux systemd {[1]:} { Started Run anacron jobs.}
Jan 27 12:08:49 mylinux systemd {[1]:} { Started Run anacron jobs at resume.}
Atenció
Logs exportats correctament al fitxer de DB
OK
Jan 26 21:29:13 mylinux systemd {[1]:} { Reached target Sleep.}
Jan 26 21:29:13 mylinux systemd {[1]:} { Started Network Manager Script Dispatcher Service.}
Jan 26 21:29:13 mylinux systemd {[1]:} { Starting Network Manager Script Dispatcher Service...}
Jan 26 21:27:55 mylinux systemd {[1]:} { Started Hostname Service.}
Jan 26 21:27:55 mylinux systemd {[1]:} { Starting Hostname Service...}
Jan 26 21:22:54 mylinux systemd {[1]:} { Started Session 458 of user jomateix.}
Jan 26 21:14:23 mylinux systemd {[1]:} { Started Session 455 of user jomateix.}
Jan 26 21:12:25 mylinux systemd {[1]:} { Started Session 454 of user jomateix.}
Jan 26 21:11:40 mylinux systemd {[1]:} { Started Session 453 of user jomateix.}
Elements filtrats: 81


Documents
FSO_19-20
P1
RESULT
logs2db_
150320.db

Comprimida



Triga més en carregar però el funcionament acaba sent el mateix.

Errònia

```
Fitxer de logs: /home/cris/Documents/FSO_19-20/P1/Moodle/logs_p1.2/errors      errors
Permisos:
-rw-------


Fitxer de BD: /home/cris/Documents/FSO_19-20/P1/Moodle/dbs_p1.2/messages.db    messages.db
Permisos:
-rw-------
Error de format: Jan 31 14:19:40 mylinux kernel: no error       KERNEL supported cpus:


Error de format: Jan 31 25:19:40 mylinux kernel: error1 KERNEL supported cpus:


Error de format: Jan 31 14:69:40 mylinux kernel: error2 KERNEL supported cpus:


Error de format: Jan 31 25:19:69 mylinux kernel: error3 KERNEL supported cpus:


Error de format: Jan 31 25:19:40 mylinux kernel  error4 KERNEL supported cpus:


Error de format: Jan 32 14:19:40 mylinux kernel: error5 KERNEL supported cpus:


Error de format: Feb 31 25:19:40 mylinux kernel: error6 KERNEL supported cpus:
```

En el cas de provar amb el fitxer de logs amb errors obtenim aquesta sortida.