Proiect ITBI MyLast

Echipa Machine Head

Budău Ștefan, Constandache Vlad-Ciprian

Introducere în myLast (Descrierea temei)

2. **MyLast** (1–2 studenți) – Scrieți un program (script) shell care emuleaza comportamentul comenzilor `last/lastb` folosind in acest sens continutul fisierului `/var/log/auth.log[1-4]` (Atentie: nu `/var/log/wtmp` sau `/var/log/btmp` care sunt fisiere binare). Implementarea trebuie sa suporte minimal comportamentul comenzilor `last/lastb` indus de folosirea flag-urilor `-n, -p, -s, -t`. Puteti implementa doua comenzi diferite, la fel ca in cazul `last/lastb`, sau puteti implementa o singura comanda care sa emuleze comportamentul celor doua.
Obs: Comanda `less` stie sa decomprime fisiere de tip gzip, adica puteti rula comanda `less /var/log/auth.log.3.gz` si sa obtineti rezultatul anticipat, dar alte comenzi, de pilda `grep` sau `more`, nu stiu.

Comanda "last" afișează informații despre ultimii utilizatori conectați. Este destul de convenabil și la îndemână atunci când trebuie să urmărim activitățile de conectare sau să investigăm un security breach.

Ultima comandă va lua, în mod implicit, fișierul de sistem /var/log/wtmp ca sursă de date pentru a genera rapoarte.

wtmp este un fișier binar pe sistemele de operare *nix care păstrează un istoric al tuturor activităților de conectare și deconectare.

Soluție adoptată:

Emularea comenzii last + parsarea basic a comenzii cu un input minimal pentru flagurile -n -s -p

-t. Astfel, am scris o comandă care afișeaza operațiile înregistrate în fișierele auth.log, când au fost deschise (pornite) și când au fost închise (oprite).

Formatul inputului este: ./machinelast [-n <nr>][-s <data>][-p <data>][-t <data>] (path la fisier)

Formatul outputului este: ora / data / user+mașină / nume+id comanda / deschisă sau închisă.

```
stefan@stefan-Virtualnavaspatiala:~/Desktop$ ./machinelast.sh -p 2025-01-08 -n 6 ../../../var/log/auth.log.1

11:48:26 2025-01-08 stefan-Virtualnavaspatiala gdm-launch-environment]: a fost deschis
11:48:26 2025-01-08 stefan-Virtualnavaspatiala (systemd): a fost deschis
11:54:17 2025-01-08 stefan-Virtualnavaspatiala gdm-password]: a fost deschis
11:54:17 2025-01-08 stefan-Virtualnavaspatiala (systemd): a fost deschis
11:54:21 2025-01-08 stefan-Virtualnavaspatiala gdm-launch-environment]: a fost inchis
11:55:01 2025-01-08 stefan-Virtualnavaspatiala CRON[2877]: a fost deschis
```

În cazul unui input greșit, este amintit userului formatul corect, sau este menționat unde este greșeala.

```
stefan@stefan-Virtualnavaspatiala:~/Desktop$ ./machinelast.sh -p 2025-01 -n 6 ../../../var/log/auth.log.1
Format invalid -p: 2025-01
```

```
stefan@stefan-Virtualnavaspatiala:~/Desktop$ ./machinelast.sh -p 2025-01 -q 6 ../../../var/log/auth.log.1
Format: ./machinelast.sh [-n <numar_linii>] [-s <yyyy-mm-dd>] [-t <yyyy-mm-dd>] [-p <yyyy-mm-dd>] <path_la_f
ile>
stefan@stefan-Virtualnavaspatiala:~/Desktop$
```

Probleme întâlnite:

1. Unele fișiere erau tratate ca binary de grep și refuza să le citească => am rezolvat obligând grep să trateze ca text fisierul prin flagul -a.

```
else
    grep -a -E "session opened|session closed" "$file"
fi
```

2. În loc de user ne apărea ”root” la început, nu știam motivul până am văzut ca am scris greșit numărul argumentului la printf.

```
2025-01-08T13:52:27.298394+02:00 sudo:        crashed
root(uid=0) 2025-01-08T13:55:02.074389+02:00 CRON[5984]:  still running
2025-01-08T13:55:02.081035+02:00 CRON[5984]:  crashed
root(uid=0) 2025-01-08T14:05:02.668058+02:00 CRON[6071]:  still running
2025-01-08T14:05:02.674488+02:00 CRON[6071]:  crashed
(base) vlad-ciprian@vlad-ciprian-VirtualBox:~/Desktop$ nano my_last.sh
(base) vlad-ciprian@vlad-ciprian-VirtualBox:~/Desktop$
```

3. Am mai întâlnit multe erori de output care erau din cauza acoladelor puse greșit => soluție: am deschis ochii mai tare când scriam codul.

Codul:

```
  GNU nano 7.2                                machinelast.sh
#!/bin/bash

# valorile default pt flaguri
LIMITA_LINII=0
DATA_SINCE=""
DATA_FINAL=""
DATA_EXACTA=""

# functie pt usage (input format)
usage() {
    echo "Format: $0 [-n <numar_linii>] [-s <yyyy-mm-dd>] [-t <yyyy-mm-dd>] [-p <yyyy-mm-dd>] <path_la_file>
    exit 1
}

# parseaza optiuni folosind getopts
while getopts ":n:s:t:p:" opt; do
    case $opt in
        n) LIMITA_LINII=$OPTARG ;;
        s) DATA_SINCE=$OPTARG ;;
        t) DATA_FINAL=$OPTARG ;;
        p) DATA_EXACTA=$OPTARG ;;
        *) usage ;;
    esac
done
```

Am declarat flagurile opționale la început

Și le-am citit valorile prin acest *while*

```
# shifteaza optiunile parsate ca sa fie log fileul ultimul argument
shift $((OPTIND - 1))

# verifica daca este log file dat
if [[ $# -ne 1 ]]; then
    usage
fi

LOG_FILE=$1

# daca avem data pt -s transform in unix timestamp
if [[ -n $DATA_SINCE ]]; then
    DATA_SINCE_TS=$(date -d "$DATA_SINCE" +%s 2>/dev/null)
    if [[ $? -ne 0 ]]; then
        echo "Format invalid -s: $DATA_SINCE"
        exit 1
    fi
else
    DATA_SINCE_TS=0
fi

# daca avem data pt -t transform in unix timestamp (ts) iar
if [[ -n $DATA_FINAL ]]; then
    DATA_FINAL_TS=$(date -d "$DATA_FINAL" +%s 2>/dev/null)
    if [[ $? -ne 0 ]]; then
        echo "Format invalid -t: $DATA_FINAL"
        exit 1
    fi
else
    DATA_FINAL_TS=0
fi

# daca avem data pt -p data, verifica daca e valida
if [[ -n $DATA_EXACTA ]]; then
    DATA_EXACTA_VALID=$(date -d "$DATA_EXACTA" +%Y-%m-%d 2>/dev/null)
    if [[ $? -ne 0 ]]; then
        echo "Format invalid -p: $DATA_EXACTA"
        exit 1
    fi
else
    DATA_EXACTA_VALID=""
fi

# functie pt a procesa un log file
process_file() {
    local file=$1
    if [[ $file == *.gz ]]; then
        zgrep -a -E "session opened|session closed" "$file"
    else
        grep -a -E "session opened|session closed" "$file"
    fi
}

# functie pt a parsa si formata log lines
parse_log() {
    awk -v data_since_ts="$DATA_SINCE_TS" \
        -v data_final_ts="$DATA_FINAL_TS" \
        -v data_exacta="$DATA_EXACTA_VALID" '
        {
            split($1, datetime, "T")  # ptc is data si ora is despartite de T in log file
            log_date = datetime[1]
            log_time_full = datetime[2]  # partea cu ora dar tot cu secunde si numere multe

            split(log_time_full, time_parts, "\\.")  # despartite de "."
            log_time = time_parts[1]

            # convertul la unix timestamp pt comparatii
            cmd = "date -d \"" log_date "T" log_time "\" +%s"
            cmd | getline timestamp
```

După ce am verificat si file-ul, am transformat datele primite in timestamps unix pentru a le putea compara mai încolo.

În cazul flagului -p, nu am mai convertit pentru că mai încolo doar verificăm dacă coincid, nu comparam mai mare/mai mic.

Am folosit zgrep pentru fișiere gzip.

Aici am parsat si transformat datele din fișier.

```
            close(cmd)

            # aici is conditiile de afisare
            if (data_exacta != "") {
                if (log_date == data_exacta) {
                    if ($0 ~ /session opened/) {
                        printf "%-8s %-10s %-12s %s a fost deschis\n", log_time, log_date, $2, $3;
                    } else if ($0 ~ /session closed/) {
                        printf "%-8s %-10s %-12s %s a fost inchis\n", log_time, log_date, $2, $3;
                    }
                }
            } else if ((data_since_ts == 0 || timestamp >= data_since_ts) &&
                       (data_final_ts == 0 || timestamp <= data_final_ts)) {
                if ($0 ~ /session opened/) {
                    printf "%-8s %-10s %-12s %s a fost deschis\n", log_time, log_date, $2, $3;
                } else if ($0 ~ /session closed/) {
                    printf "%-8s %-10s %-12s %s a fost inchis\n", log_time, log_date, $2, $3;
                }
            }
        }
    '
}

# procesarea log file si afisarea cu line limit (optional)
if [[ $LIMITA_LINII -gt 0 ]]; then
    process_file "$LOG_FILE" | parse_log | head -n "$LIMITA_LINII"
else
    process_file "$LOG_FILE" | parse_log
fi

^G Help        ^O Write Out   ^W Where Is    ^K Cut         ^T Execute     ^C Location    M-U Undo
^X Exit        ^R Read File   ^\ Replace     ^U Paste       ^J Justify     ^/ Go To Line  M-E Redo
```

La final avem o parte pentru flagul -n, unde afișăm doar numarul dat de linii.