

ΕΙΣΑΓΩΓΗ ΣΤΟΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ (2006-07)

Άσκηση 5

Στα συστήματα Unix, οι συνδέσεις των διαφόρων χρηστών καταγράφονται με αρκετή λεπτομέρεια. Με την εντολή “last” μπορούμε να δούμε πληροφορίες για τις συνδέσεις των χρηστών, από τις πιο πρόσφατες προς τις παλιότερες. Με ένα “man last” μπορείτε να δείτε τους διάφορους τρόπους χρήσης της εντολής αυτής. Συνδεθείτε και σε ένα σύστημα Unix και εκτελέστε την εντολή “last -a”. Για παράδειγμα, τα πρώτα αποτελέσματα της εντολής αυτής σ’ ένα σταθμό εργασίας Sun του Τμήματος (π.χ. τον pegasus) τη στιγμή που γράφονται αυτές οι γραμμές είναι τα εξής:

```
% last -a
std04241 pts/10      Wed Dec 13 19:24   still logged in   atheds1-128103.otenet.gr
std04241 pts/10      Wed Dec 13 19:17 - 19:22 (00:04)   atheds1-128103.otenet.gr
takis    pts/7       Wed Dec 13 19:01   still logged in   kronos.di.uoa.gr
std03147 pts/5       Wed Dec 13 18:42   still logged in   atheds1-157624.otenet.gr
std06168 pts/2       Wed Dec 13 18:34 - 18:43 (00:09)   dyn126.kif6.nas.panafonet.gr
std05097 pts/11      Wed Dec 13 18:22   still logged in   ppp192-156.adsl.forthnet.gr
std03147 pts/12      Wed Dec 13 17:56   still logged in   atheds1-157624.otenet.gr
std05097 pts/11      Wed Dec 13 17:54 - 18:17 (00:23)   ppp38-102.adsl.forthnet.gr
stud1308 pts/11      Wed Dec 13 17:50 - 17:54 (00:03)   alefantos.di.uoa.gr
std06025 pts/1       Wed Dec 13 17:45   still logged in   atheds1-235760.otenet.gr
.....
```

Παρατηρήστε ότι στα αποτελέσματα της εντολής “last -a” έχουμε μία γραμμή για κάθε σύνδεση χρήστη, στην οποία περιέχονται το όνομα του χρήστη, το τερματικό σύνδεσής του, η ημερομηνία και ώρα σύνδεσής του, η ώρα αποσύνδεσής του και ο συνολικός χρόνος που ήταν συνδεδεμένος (ή η φράση “still logged in”, αν δεν έχει αποσυνδεθεί ακόμα) και, τέλος, η πληροφορία για το από πού συνδέθηκε ο χρήστης αυτός.

Στην άσκηση αυτή, θα σας ζητηθεί να γράψετε ένα πρόγραμμα C που να επεξεργάζεται με κάποιο τρόπο τα αποτελέσματα της εντολής “last”, αλλά για να μην περιπλέξουμε άσκοπα το πρόβλημα, θα εξαιρέσουμε από την επεξεργασία κάποιες γραμμές. Συγκεκριμένα, για την άσκηση θα μας ενδιαφέρουν αποτελέσματα από την εντολή “last -a | grep pts/ | grep ’(’”. Συγκρίνετε τα αποτελέσματα αυτής της εντολής με αυτά μίας απλής “last -a” και θα δείτε τις διαφορές.

Γράψτε, λοιπόν, ένα πρόγραμμα C (έστω ότι το εκτελέσιμο που θα δημιουργηθεί ονομάζεται “lastout”), το οποίο όταν καλείται χωρίς ορίσματα στην γραμμή εντολής να διαβάσει από την πρότυπη είσοδο (stdin) δεδομένα όπως αυτά που παράγει η εντολή “last -a | grep pts/ | grep ’(’” και να δημιουργεί ένα δυαδικό δέντρο με τα εξής χαρακτηριστικά:

- Σε κάθε κόμβο του δυαδικού δέντρου να είναι καταχωρημένες οι πληροφορίες για τις συνδέσεις ενός χρήστη, με βάση τα δεδομένα εισόδου, δηλαδή το όνομά του, το πλήθος των συνδέσεών του, ο συνολικός χρόνος για όλες τις συνδέσεις του και μία λίστα με πληροφορίες για τις συνδέσεις του.
- Το δέντρο να είναι ταξινομημένο με βάση το όνομα χρήστη (αλφαβητική σειρά).
- Το πλήθος των συνδέσεων του χρήστη ταυτίζεται με το πλήθος των γραμμών στην είσοδο που τον αφορούν.
- Ο συνολικός χρόνος σύνδεσης κάθε χρήστη να προκύπτει από το άθροισμα των χρόνων σύνδεσης που βρίσκονται μεταξύ των παρενθέσεων στις γραμμές από την είσοδο που αφορούν αυτόν

τον χρήστη. Αναφερόμαστε στην προτελευταία στήλη της εισόδου, όπου μεταξύ των παρενθέσεων ο χρόνος σύνδεσης δίνεται σαν hh:mm, δηλαδή hh ώρες και mm λεπτά, ή ακόμα και σαν d+hh:mm, όπου εδώ το d είναι πλήθος ημερών (όχι κατ' ανάγκη μονοψήφιο).

- Κάθε κόμβος της λίστας των συνδέσεων του χρήστη να αντιστοιχεί σε μία γραμμή από την είσοδο που αφορά αυτόν τον χρήστη. Στον κόμβο να καταχωρούνται η ημερομηνία και ώρα που συνδέθηκε ο χρήστης, πόσο χρόνο ήταν συνδεδεμένος, σε λεπτά, και από πού συνδέθηκε.

Αφού το πρόγραμμα “lastout” δημιουργήσει το δυαδικό δέντρο, να εκτυπώσει στην πρότυπη έξοδο (stdout) τις πληροφορίες που έχουν καταχωρηθεί σ' αυτό, με αλφαβητική σειρά των ονομάτων χρηστών, φυσικά, αλλά και διάφορα στατιστικά στοιχεία, όπως αυτά που φαίνονται στην ενδεικτική εκτέλεση που ακολουθεί.

```
% last -a | grep pts/ | grep '(' | head > lastout-pegasus.txt
% cat lastout-pegasus.txt
std04241 pts/10 Wed Dec 13 19:17 - 19:22 (00:04) atheds1-128103.otenet.gr
std06168 pts/2 Wed Dec 13 18:34 - 18:43 (00:09) dyn126.kif6.nas.panafonet.gr
std05097 pts/11 Wed Dec 13 17:54 - 18:17 (00:23) ppp38-102.adsl.forthnet.gr
stud1308 pts/11 Wed Dec 13 17:50 - 17:54 (00:03) alefantos.di.uoa.gr
std05041 pts/10 Wed Dec 13 17:37 - 18:43 (01:06) ppp37-148dynamic.athens.acn.gr
std06207 pts/6 Wed Dec 13 17:24 - 17:26 (00:01) ip-83-212-217-55.adsl.aueb.gr
std06207 pts/6 Wed Dec 13 17:24 - 17:24 (00:00) ip-83-212-217-55.adsl.aueb.gr
std06088 pts/1 Wed Dec 13 17:18 - 17:41 (00:22) atheds1-82021.otenet.gr
std05049 pts/2 Wed Dec 13 17:08 - 18:23 (01:14) 87.239.238.54
std05041 pts/6 Wed Dec 13 17:02 - 17:11 (00:09) ppp35-7dynamic.athens.acn.gr
% ./lastout < lastout-pegasus.txt
std04241 - 1 connection(s) - 4 minute(s) total connection time
Dec 13 19:17 - 4 - atheds1-128103.otenet.gr
std05041 - 2 connection(s) - 75 minute(s) total connection time
Dec 13 17:37 - 66 - ppp37-148dynamic.athens.acn.gr
Dec 13 17:02 - 9 - ppp35-7dynamic.athens.acn.gr
std05049 - 1 connection(s) - 74 minute(s) total connection time
Dec 13 17:08 - 74 - 87.239.238.54
std05097 - 1 connection(s) - 23 minute(s) total connection time
Dec 13 17:54 - 23 - ppp38-102.adsl.forthnet.gr
std06088 - 1 connection(s) - 22 minute(s) total connection time
Dec 13 17:18 - 22 - atheds1-82021.otenet.gr
std06168 - 1 connection(s) - 9 minute(s) total connection time
Dec 13 18:34 - 9 - dyn126.kif6.nas.panafonet.gr
std06207 - 2 connection(s) - 1 minute(s) total connection time
Dec 13 17:24 - 1 - ip-83-212-217-55.adsl.aueb.gr
Dec 13 17:24 - 0 - ip-83-212-217-55.adsl.aueb.gr
stud1308 - 1 connection(s) - 3 minute(s) total connection time
Dec 13 17:50 - 3 - alefantos.di.uoa.gr

Total number of users is: 8
Total number of connections is: 10
Average: 1.25 connections/user
%
```

Το πρόγραμμα “lastout”, εκτός από το να διαβάζει τα δεδομένα του από την πρότυπη είσοδο, να έχει και τη δυνατότητα όταν καλείται στην γραμμή εντολής με την επιλογή -itf <txtfile>, να

διαβάζει το αρχείο <txtfile> αντί για την πρότυπη είσοδο, αλλά, κατά τα άλλα, να συμπεριφέρεται με τον ίδιο τρόπο.

Αν το πρόγραμμα “lastout” κληθεί με την επιλογή `-obf <binfile>`, να δημιουργεί το δυαδικό αρχείο <binfile>, στο οποίο να καταχωρεί κατά σειρά εγγραφές των 16 bytes, μία για κάθε χρήστη που εμφανίστηκαν συνδέσεις του στα δεδομένα εισόδου, με αλφαβητική σειρά των ονομάτων χρηστών. Κάθε εγγραφή περιέχει το όνομα χρήστη, που καταχωρείται στα πρώτα 10 bytes από τα 16 της εγγραφής, το πλήθος των συνδέσεων του χρήστη, που καταχωρείται σαν δυαδικός αριθμός στα επόμενα 2 bytes και το συνολικό χρόνο σύνδεσης του χρήστη, που καταχωρείται σαν δυαδικός αριθμός στα τελευταία 4 bytes από τα 16 της εγγραφής. Τα bytes που δεν θα χρησιμοποιηθούν στο τμήμα της εγγραφής που καταχωρείται το όνομα του χρήστη να γεμίσουν με ‘\0’. Στην περίπτωση που έχει δοθεί η επιλογή `-obf`, να μην εκτυπώνονται οι πληροφορίες που είναι καταχωρημένες στο δυαδικό δέντρο και τα στατιστικά στοιχεία που είδαμε προηγουμένως, δηλαδή όταν είχε δοθεί μόνο η επιλογή `-itf` η καμία επιλογή. Δηλαδή:¹

```
% ./lastout -itf lastout-pegasus.txt -obf lastout-pegasus1.bin
% od -tu2c lastout-pegasus1.bin
0000000 29556 25648 13362 13361 00000 00001 00000 00004
          s  t  d  0  4  2  4  1  \0  \0  \0 001  \0  \0  \0 004
0000020 29556 25648 13616 13361 00000 00002 00000 00075
          s  t  d  0  5  0  4  1  \0  \0  \0 002  \0  \0  \0  K
0000040 29556 25648 13616 13369 00000 00001 00000 00074
          s  t  d  0  5  0  4  9  \0  \0  \0 001  \0  \0  \0  J
0000060 29556 25648 13616 14647 00000 00001 00000 00023
          s  t  d  0  5  0  9  7  \0  \0  \0 001  \0  \0  \0 027
0000100 29556 25648 13872 14392 00000 00001 00000 00022
          s  t  d  0  6  0  8  8  \0  \0  \0 001  \0  \0  \0 026
0000120 29556 25648 13873 13880 00000 00001 00000 00009
          s  t  d  0  6  1  6  8  \0  \0  \0 001  \0  \0  \0  \t
0000140 29556 25648 13874 12343 00000 00002 00000 00001
          s  t  d  0  6  2  0  7  \0  \0  \0 002  \0  \0  \0 001
0000160 29556 30052 12595 12344 00000 00001 00000 00003
          s  t  u  d  1  3  0  8  \0  \0  \0 001  \0  \0  \0 003
0000200
%
```

Το πρόγραμμα “lastout” να έχει τη δυνατότητα να κληθεί και με την επιλογή `-ibf <binfile>`, οπότε σ’ αυτή την περίπτωση να διαβάζει το δυαδικό αρχείο <binfile> που περιέχει εγγραφές χρηστών (όνομα χρήστη, πλήθος συνδέσεων και συνολικός χρόνος σύνδεσης). Τέτοια αρχεία μπορούν να δημιουργηθούν με την επιλογή `-obf` που περιγράφηκε προηγουμένως. Κατά την ανάγνωση του αρχείου, το πρόγραμμά σας να εισάγει τις πληροφορίες των εγγραφών που διαβάζει σ’ ένα ταξινομημένο δυαδικό δέντρο με διάταξη ταξινόμησης σύμφωνα με το συνολικό χρόνο σύνδεσης. Στο τέλος να εκτυπώνει στην πρότυπη έξοδο τις πληροφορίες που έχουν καταχωρηθεί στους κόμβους του δέντρου, με αύξουσα σειρά των συνολικών χρόνων σύνδεσης, φυσικά. Παράδειγμα:

```
% ./lastout -ibf lastout-pegasus1.bin
std06207 - 2 connection(s) - 1 minute(s) (0+00:01)
stud1308 - 1 connection(s) - 3 minute(s) (0+00:03)
std04241 - 1 connection(s) - 4 minute(s) (0+00:04)
std06168 - 1 connection(s) - 9 minute(s) (0+00:09)
```

¹Με την εντολή “od” μπορούμε να δούμε το “εσωτερικό” ενός δυαδικού αρχείου. Δώστε “man od” για περισσότερες πληροφορίες.

```
std06088 - 1 connection(s) - 22 minute(s) (0+00:22)
std05097 - 1 connection(s) - 23 minute(s) (0+00:23)
std05049 - 1 connection(s) - 74 minute(s) (0+01:14)
std05041 - 2 connection(s) - 75 minute(s) (0+01:15)
%
```

Οι επιλογές `-ibf` και `-obf` μπορούν να συνδυασθούν σε μία εκτέλεση, οπότε έτσι μπορεί να διαβαστεί ένα δυαδικό αρχείο με αλφαβητική σειρά εγγραφών ως προς τα ονόματα χρηστών και να γραφεί ένα άλλο με σειρά συνολικών χρόνων σύνδεσης. Στην περίπτωση αυτή, να μην εκτυπώνεται στην πρότυπη έξοδο κάτι. Δηλαδή:

```
% ./lastout -ibf lastout-pegasus1.bin -obf lastout-pegasus2.bin
% od -tu2c lastout-pegasus2.bin
0000000 29556 25648 13874 12343 00000 00002 00000 00001
          s t d 0 6 2 0 7 \0 \0 \0 002 \0 \0 \0 001
0000020 29556 30052 12595 12344 00000 00001 00000 00003
          s t u d 1 3 0 8 \0 \0 \0 001 \0 \0 \0 003
0000040 29556 25648 13362 13361 00000 00001 00000 00004
          s t d 0 4 2 4 1 \0 \0 \0 001 \0 \0 \0 004
0000060 29556 25648 13873 13880 00000 00001 00000 00009
          s t d 0 6 1 6 8 \0 \0 \0 001 \0 \0 \0 \t
0000100 29556 25648 13872 14392 00000 00001 00000 00022
          s t d 0 6 0 8 8 \0 \0 \0 001 \0 \0 \0 026
0000120 29556 25648 13616 14647 00000 00001 00000 00023
          s t d 0 5 0 9 7 \0 \0 \0 001 \0 \0 \0 027
0000140 29556 25648 13616 13369 00000 00001 00000 00074
          s t d 0 5 0 4 9 \0 \0 \0 001 \0 \0 \0 J
0000160 29556 25648 13616 13361 00000 00002 00000 00075
          s t d 0 5 0 4 1 \0 \0 \0 002 \0 \0 \0 K
0000200
%
```

Οι επιλογές `-itf` και `-ibf` είναι αποκλειστικές μεταξύ τους. Δηλαδή δεν μπορούν να υπάρχουν και οι δύο στην γραμμή εντολής. Δεν είναι απαραίτητο να το απαγορεύσετε αυτό. Αρκεί, αν υπάρχουν πολλές εμφανίσεις των επιλογών αυτών στην γραμμή εντολής, να λαμβάνεται υπόψη μόνο η τελευταία.

Επίσης, το πρόγραμμα “lastout” που θα γράψετε να παρέχει τη δυνατότητα στις επιλογές `-itf`, `-ibf` και `-obf` να μπορεί να δοθεί σαν όνομα αρχείου το ειδικό όνομα `-` (ένα σκέτο πλην), οπότε τότε, στις περιπτώσεις των επιλογών `-itf` και `-ibf`, το πρόγραμμα να διαβάζει από την πρότυπη είσοδο (stdin), και στην περίπτωση της επιλογής `-obf`, το πρόγραμμα να γράφει στην πρότυπη έξοδο (stdout). Με αυτόν τον τρόπο, μπορούμε να κάνουμε πολύ ενδιαφέρουσες εκτελέσεις, όπως:

```
% ls -l ~iphw/samples/lastout
total 28436
-rw-r--r-- 1 iphw other 9665847 Dec 13 21:28 lastout-gaia.txt
-rw-r--r-- 1 iphw other 2070805 Dec 13 21:26 lastout-heavy.txt
-rw-r--r-- 1 iphw other 48169 Dec 13 21:23 lastout-iokasti.txt
-rw-r--r-- 1 iphw other 2508258 Dec 13 21:27 lastout-kronos.txt
-rw-r--r-- 1 iphw other 186863 Dec 13 21:22 lastout-laios.txt
-rw-r--r-- 1 iphw other 10563 Dec 13 21:24 lastout-linux07.txt
-rw-r--r-- 1 iphw other 849 Dec 13 19:26 lastout-pegasus.txt
-rw-r--r-- 1 iphw other 21860 Dec 13 21:26 lastout-zeus.txt
% cat ~iphw/samples/lastout/*.txt | ./lastout -itf - -obf - | ./lastout -ibf -
.....
```

Το πρόγραμμα που θα γράψετε θα πρέπει να είναι δομημένο σε ένα σύνολο από **τουλάχιστον τρία πηγαία αρχεία C** (με κατάληξη `.c`) και **τουλάχιστον δύο αρχεία επικεφαλίδας** (με κατάληξη `.h`). Για να παραδώσετε τη δουλειά σας, θα πρέπει να κάνετε τα εξής:

- Τοποθετήστε όλα τα αρχεία (πηγαία και αρχεία επικεφαλίδας) μέσα σ' ένα κατάλογο που θα δημιουργήσετε, έστω με όνομα `hw5`, στους σταθμούς εργασίας Suns του Τμήματος. Επίσης, τοποθετήστε στον κατάλογο αυτό και ένα αρχείο με όνομα `README`, στο οποίο να δίνετε οδηγίες για τη μεταγλώττιση των αρχείων και την κατασκευή του τελικού εκτελέσιμου.
- Όντας στον κατάλογο που περιέχει τον κατάλογο `hw5`, δημιουργήστε ένα “επιπεδοποιημένο” `tar` αρχείο (έστω με όνομα `lastout.tar`) που περιέχει τον κατάλογο `hw5` και όλα του τα περιεχόμενα. Αυτό γίνεται με την εντολή `“tar cvf lastout.tar hw5”`.²
- Συμπιέστε το αρχείο `lastout.tar`, ώστε να δημιουργηθεί το αρχείο `lastout.tar.gz`. Αυτό γίνεται με την εντολή `“gzip lastout.tar”`.³
- Το αρχείο `lastout.tar.gz` είναι που θα πρέπει να υποβάλετε, με διαδικασία που θα ανακοινωθεί σύντομα.

Σημείωση: Η άσκηση αυτή μπορεί να παραδοθεί και από ομάδες των δύο ατόμων. Στην περίπτωση αυτή, θα παραδοθεί μόνο από το ένα μέλος της ομάδας, αλλά μέσα στο αρχείο `README` θα αναφέρονται σαφώς τα στοιχεία των δύο μελών. Ο στόχος της διαδικασίας αυτής είναι να ενισχυθεί η ιδέα της **ισότιμης** συνεργασίας σε μία ομάδα για την επίτευξη ενός στόχου.

² Αν θέλετε να ανακτήσετε την δενδρική δομή που έχει φυλαχθεί σ' ένα “επιπεδοποιημένο” `tar` αρχείο `file.tar`, αυτό μπορεί να γίνει με την εντολή `“tar xvf file.tar”`.

³ Αν θέλετε να αποσυμπιέσετε ένα αρχείο `file.gz` που έχει συμπιεσθεί με την εντολή `gzip`, αυτό μπορεί να γίνει με την εντολή `“gzip -d file.gz”`.