

Εργαστήριο Λειτουργικών Συστημάτων

Εαρινό Εξάμηνο 2010-2011

4^η Άσκηση

2ο Πρόβλημα

Μελέτη συμπεριφοράς συστήματος (3.1.2a)

Στις παρακάτω εικόνες βλέπουμε την εκτέλεση της εντολής του πρώτου ερωτήματος της άσκησης (hinfo) για 17 δευτερόλεπτα. Κάθε δευτερόλεπτο, αυτή η εντολή κάνει την κλήση συστήματος που επιστρέφει κάθε φορά τον συνολικό αριθμό των οπών μνήμης καθώς και το μέγιστο, ελάχιστο και μέσο μέγεθος οπής (σε KB). Αυτές είναι αντίστοιχα οι τέσσερις στήλες του πίνακα που εκτυπώνεται.

Παρατηρούμε πως όλες οι γραμμές περιέχουν τις ίδιες τιμές. Αυτό οφείλεται στο ότι δεν δημιουργούνται καινούριες διεργασίες εκείνη την χρονική περίοδο (των 17 δευτερολέπτων) και έτσι δεν «γεμίζουν» οι σπές (ούτε και αδειάζουν φυσικά).

Εάν καταφέρουμε και τρέξουμε κάποιες καινούριες διεργασίες (ενώ εκτελείται η εντολή) θα δούμε σίγουρα την διαφορά.

- $M\epsilon$ first fit allocation

```
If you do not have sufficient memory to run X Windows, standard MINIX 3
supports multiple virtual terminals. Just use ALT+F1, F2, F3 and F4 to
navigate among them.
```

To get rid of this message, edit /etc/motd.

```
# hinfo
```

[illegible]

Εδώ βλέπουμε τα στατιστικά των οπών μνήμης όταν χρησιμοποιείται ο αλγόριθμος first fit για την δέσμευση μνήμης (memory allocation), ο οποίος ανατρέπει την λίστα των οπών και όταν βρει κάποια που να είναι αρκετά μεγάλη για να χωρέσει το ποσό

μνήμης που θέλουμε να δεσμεύσουμε (για την εκτέλεση μιας διεργασίας) την χρησιμοποιεί και μετά την βγάζει από την λίστα των ελεύθερων οπών.

- Με best fit allocation

```
If you do not have sufficient memory to run X Windows, standard MINIX 3
supports multiple virtual terminals. Just use ALT+F1, F2, F3 and F4 to
navigate among them.

To get rid of this message, edit /etc/motd.

# hinfo

204      48625    0      238
204      48625    0      238
204      48625    0      238
204      48625    0      238
204      48625    0      238
204      48625    0      238
204      48625    0      238
204      48625    0      238
204      48625    0      238
204      48625    0      238
204      48625    0      238
204      48625    0      238
204      48625    0      238
204      48625    0      238
204      48625    0      238
204      48625    0      238
204      48625    0      238
```

Σε αυτήν την περίπτωση χρησιμοποιούμε τον αλγόριθμο best fit. Με αυτόν τον τρόπο όχι μόνο βρίσκουμε μια οπή για την δέσμευση, αλλά αυτήν που αφήνει τον λιγότερο ελεύθερο χώρο.

Με μια πρώτη σκέψη θα λέγαμε πως η σπατάλη μνήμης (σχεδόν) ελαχιστοποιείται, δηλαδή έχουμε οικονομία στο υλικό (μνήμη) αλλά έχουμε μεγαλύτερη σπατάλη χρόνου, αφού θα πρέπει να ανατρέξουμε ολόκληρη την λίστα των οπών. Όμως μένουν ακρησιμοποιήτες κάποιες μικρές οπές, με αποτέλεσμα η λίστα να γεμίζει πολλές και μικρές ελεύθερες οπές οι οποίες δεν θα μπορούν να εξυπηρετήσουν μελλοντικές αιτήσεις μνήμης. Αυτή η κατάσταση ονομάζεται κατακερματισμός (fragmentation) και είναι «άβολη» όχι μόνο για τον συγκεκριμένο αλγόριθμο αλλά και γενικά για όλες τις στρατηγικές διαχείρισης μνήμης. Όσον αφορά τον χρόνο, ισχύει αυτό που είπαμε.

Επίσης παρατηρούμε ότι το μέγιστο μέγεθος οπής είναι ελάχιστα πιο μικρό στον best fit. Αυτό οφείλεται στο ότι αφήνονται μικρότερα κενά (οπές), όπως προαναφέραμε.