

Εργαστήριο Λειτουργικών Συστημάτων
Ακ. Έτος 2012-13
Άσκηση 4

Η εργασία να εκπονηθεί με την έκδοση minix 3.1.x

1^ο Πρόβλημα

Αναπτύξτε μια νέα κλήση του συστήματος (system call) στα πλαίσια του διαχειριστή διεργασιών (process manager) που να επιστρέφει τα εξής στοιχεία:

1. Το πλήθος των οπών μνήμης.
2. Το μέσο μέγεθος οπής.

Υλοποιείτε ένα απλό πρόγραμμα που χρησιμοποιεί την νέα κλήση του συστήματος που αναπτύξατε και να εμφανίζει τα ακόλουθα στοιχεία ανά 1 δευτερόλεπτο:

Πλήθος οπών <tab> Μέσο μέγεθος σε mb

Τα αρχεία που θα παραδώσετε θα πρέπει να είναι καλά δομημένα (με την χρήση tab, κλπ.) και καλά σχολιασμένα.

2^ο Πρόβλημα

Στην έκδοση 3.1.x του MINIX ο μηχανισμός ανάθεσης μνήμης που χρησιμοποιεί ο διαχειριστής διεργασιών (process manager) ακολουθεί την πολιτική first fit: ανατρέχει την λίστα των οπών, κάθε φορά από την αρχή της λίστας, προσπαθώντας να εντοπίσει την πρώτη οπή που είναι αρκετά μεγάλη για να χωρέσει το μέγεθος της μνήμης που πρέπει να ανατεθεί.

- Τροποποιήστε τον μηχανισμό ανάθεσης μνήμης έτσι ώστε σε κάθε ανάθεση μνήμης να εμφανίζεται στην οθόνη το μέγεθος της οπής που εντοπίστηκε και ο υπόλοιπος διαθέσιμος χώρος στην οπή, μετά την ανάθεση
- Τροποποιήστε τον μηχανισμό ανάθεσης μνήμης έτσι ώστε να ακολουθεί την πολιτική best fit: ανατρέχει την λίστα των οπών και εντοπίζει την οπή που αφήνει το μικρότερο κενό χώρο όταν δεσμευθεί το μέγεθος της μνήμης που έχει ζητηθεί. Σε κάθε ανάθεση μνήμης θα πρέπει να εμφανίζεται στην οθόνη το μέγεθος της οπής που εντοπίστηκε και ο υπόλοιπος διαθέσιμος χώρος στην οπή, μετά την ανάθεση.
- Τροποποιήστε τον μηχανισμό ανάθεσης μνήμης ξανά έτσι ώστε να ακολουθεί την πολιτική worst fit: ανατρέχει την λίστα των οπών και εντοπίζει την οπή που αφήνει τον περισσότερο κενό χώρο όταν δεσμευθεί το μέγεθος της μνήμης που έχει ζητηθεί. Σε κάθε ανάθεση μνήμης θα πρέπει να εμφανίζεται στην οθόνη το μέγεθος της οπής που εντοπίστηκε και ο υπόλοιπος διαθέσιμος χώρος στην οπή, μετά την ανάθεση.

Για να μελετήσετε τη συμπεριφορά των τροποποιήσεών σας, εκτελέστε οποιοδήποτε πρόγραμμα στο Minix δίνοντας στο τέλος το σύμβολο & (π.χ. nano& - εκκινεί το πρόγραμμα nano παράλληλα με την κονσόλα, αν δώσετε κατόπιν την εντολή ps θα το δείτε σαν διεργασία).

3ο Πρόβλημα

Με χρήση της κλήσης του συστήματος και του προγράμματος που αναπτύξατε στο 1^ο πρόβλημα μελετήστε την συμπεριφορά του συστήματος όταν χρησιμοποιείται η κάθε μια πολιτική εντοπισμού οπών. Πιο συγκεκριμένα, θα πρέπει να δείξετε ότι ο αλγόριθμός σας επιλέγει οπές οι οποίες δεν θα επιλέγονταν με τον αλγόριθμο first-fit. Χρησιμοποιήστε τον κατώθι απλό τρόπο ελέγχου:

Δημιουργήστε 2 απλά προγράμματα, programA και programB, με τις main συναρτήσεις

```
programA: main() { while (1) {} }  
programB: main() { int arr[100];  
               printf("Hello World\n");  
               while (1){ } }
```

Κατόπιν εκκινήστε διεργασίες με την κατώθι σειρά:

```
% programB &  
% programB &  
% programB &  
% programA &  
% programB &
```

Ελέγξτε ότι το programB πιάνει περισσότερη μνήμη από το programA, αν όχι, δώστε του περισσότερη μνήμη με την εντολή chmem ή προσθέτοντας κάποιο μεγάλο array. Κατόπιν τερματίστε την 4^η διεργασία (programA) και αμέσως μετά την 2^η. Θα πρέπει έτσι να μείνει μια λίστα οπών με πρώτη μια οπή μεγέθους programB και κατόπιν μια οπή μεγέθους program A. Μπορείτε να βρείτε τα process ids με την εντολή

```
% ps -a
```

και να τις τερματίσετε με τις εντολές

```
% kill -9 [process_id]
```

Τώρα εκκινήστε μια νέα διεργασία programA (% programA &). Με τον first fit αλγόριθμο, η διεργασία θα πρέπει να εισαχθεί στην οπή που αφήνει η τερματισμένη διεργασία programB, ενώ με τον best-fit, σε αυτή που αφήνει η τερματισμένη programA. Υπάρχει μια μικρή περίπτωση να εμφανισθεί σπατάλη μνήμης κατά τον τερματισμό της programA, οπότε η νέα programA να μην μπορεί να χωρέσει ακριβώς στην οπή, οπότε μπορείτε να δοκιμάσετε και κάποια τρίτη διεργασία με διαφορετικά μεγέθη μνήμης.

Περιγράψτε τα συμπεράσματα σας σε μια συνοπτική αναφορά, χρησιμοποιώντας το template στο τέλος της εκφώνησης. Στην αναφορά θα πρέπει να **δείξετε με κατάλληλα screenshots τον τρόπο με τον οποίο συμπεριφέρεται ο first-fit και κατόπιν ο best-fit και ο worst-fit αλγόριθμος, κάνοντας χρήση της υλοποίησης που κάνατε στο πρόβλημα 1.**

Παράδοση Άσκησης

Για κάθε άσκηση θα πρέπει να υποβάλλετε μια λίστα με όλα τα τροποποιημένα αρχεία συστήματος και τις αλλαγές σε αυτά, σε ένα ενιαίο αρχείο Word / PDF. Θα πρέπει επίσης να φαίνεται καθαρά το πλήρες path και ο πηγαίος κώδικας

τυχόν αρχείων που έχετε δημιουργήσει εκ νέου. Για την υποβολή θα πρέπει υποχρεωτικά να χρησιμοποιηθεί το template που βρίσκεται παρακάτω. Τέλος, θα πρέπει να υποβάλλετε

- Να ονομάσετε το συμπιεσμένο αρχείο σε [AM]_ex4.[zip|rar] (π.χ. 3345_ex4.rar ή 3345_ex4.zip)
- Το συμπιεσμένο αρχείο θα πρέπει να υποβληθεί ηλεκτρονικά στην ιστοσελίδα <http://diogenis.ceid.upatras.gr/~akomninos/oslab>

ΟΛΑ ΤΑ ΟΝΟΜΑΤΑ ΤΩΝ ΑΡΧΕΙΩΝ ΚΑΘΩΣ ΚΑΙ ΤΑ ΠΕΡΙΕΧΟΜΕΝΑ ΠΡΕΠΕΙ ΝΑ ΑΠΟΤΕΛΟΥΝΤΑΙ ΜΟΝΟ ΑΠΟ ΛΑΤΙΝΙΚΟΥΣ ΧΑΡΑΚΤΗΡΕΣ.

Σημειώνεται ότι η εργασία είναι ομαδική με επιθυμητό μέγεθος ομάδας το πολύ 3 άτομα.

Φόρμα υποβολής 4ης Άσκησης

Ονόματα φοιτητών
ΑΜ φοιτητών

Άσκηση 4

Πρόβλημα 1

1. Τροποποιήσεις Minix

Λίστα τροποποιηθέντων αρχείων Minix

/usr/src/somedir/file1.c
/usr/src/somedir2/file2.c

....

Τροποποιήσεις αρχείων Minix

/usr/src/somedir/file1.c

Line	Code
50	if (var1>0) {
51	var2=var1 + 56;
...	...

/usr/src/somedir2/file2.c

....

2. Κώδικας δοκιμαστικού προγράμματος

test.c

//test.c dokimastiko programma
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
...

3. Screenshot εκτέλεσης δοκιμαστικού προγράμματος

[σχετικά screenshots τα οποία δείχνουν το output από τα δοκιμαστικά προγράμματα]

Πρόβλημα 2

First – fit

1. Τροποποιήσεις Minix

Λίστα τροποποιηθέντων αρχείων Minix

/usr/src/somedir/file1.c
/usr/src/somedir2/file2.c

....

Τροποποιήσεις αρχείων Minix

/usr/src/somedir/file1.c

Line	Code
50	if (var1>0) {
51	var2=var1 + 56;
...	...

/usr/src/somedir2/file2.c

....

Screenshot first – fit

[σχετικό screenshot το οποίο δείχνει τα στοιχεία της οπής στην οποία έγινε ανάθεση]

Best - fit

1. Τροποποιήσεις Minix

Λίστα τροποποιηθέντων αρχείων Minix

/usr/src/somedir/file1.c
/usr/src/somedir2/file2.c

....

Τροποποιήσεις αρχείων Minix

/usr/src/somedir/file1.c

Line	Code
50	if (var1>0) {
51	var2=var1 + 56;
...	...

/usr/src/somedir2/file2.c

....

Screenshot best – fit

[σχετικό screenshot το οποίο δείχνει τα στοιχεία της οπής στην οποία έγινε

ανάθεση]

Worst – fit

1. Τροποποιήσεις Minix

Λίστα τροποποιηθέντων αρχείων Minix

/usr/src/somedir/file1.c
/usr/src/somedir2/file2.c

....

Τροποποιήσεις αρχείων Minix

/usr/src/somedir/file1.c

Line	Code
50	if (var1>0) {
51	var2=var1 + 56;
...	...

/usr/src/somedir2/file2.c

....

Screenshot worst – fit

[σχετικό screenshot το οποίο δείχνει τα στοιχεία της οπής στην οποία έγινε ανάθεση]

Πρόβλημα 3

1. Σχόλια για τη συμπεριφορά των αλγορίθμων

[μέχρι 500 λέξεις, εξαιρουμένων των screenshots για κάθε αλγόριθμο]