

Εργαστήριο Λειτουργικών Συστημάτων

Σειμένης Σπύρος	5070
Καλλιβωκάς Δημήτριος	4993

Άσκηση 4

Οι παρακάτω αλλαγές έγιναν σε έκδοση minix 3.1.3 διότι είναι η τελευταία λειτουργική έκδοση στην οποία ο pm είναι ενοποιημένος. Τα παρακάτω εφαρμόζονται σχεδόν πανομοιότυπα σε εκδόσεις 3.1.7 και μετά στις οποίες ο pm έχει χωριστεί σε pm και vm με μικρές αλλαγές. Συγκεκριμένα ο vm διατηρεί την λειτουργικότητα και την δυνατότητα να λαμβάνει πληροφορίες από τον kernel για τον πίνακα των οπών αλλά ζητείται η αλλαγή να γίνει στον pm. Η μόνη περαιτέρω αλλαγή θα ήταν η υλοποίηση ενός taskcall ώστε μέσω system call στον pm να λαμβάνεται ο πίνακας από τον vm.

Πρόβλημα 1

1. Τροποποιήσεις Minix

Λίστα τροποποιηθέντων αρχείων Minix

```
/usr/src/servers/pm/proto.h  
/usr/src/servers/pm/table.c  
/usr/src/servers/pm/alloc.c
```

Τροποποιήσεις αρχείων Minix

```
/usr/src/servers/pm/proto.h
```

Line	Code
10	/* alloc.c */
11	_PROTOTYPE(phys_clicks alloc_mem, (phys_clicks clicks));
12	_PROTOTYPE(void free_mem, (phys_clicks base, phys_clicks clicks));
13	_PROTOTYPE(void mem_init, (struct memory *chunks, phys_clicks *free));
14	_PROTOTYPE(void mem_info, (void));
15	#if ENABLE_SWAP

```
/usr/src/servers/pm/table.c
```

Line	Code
10	no_sys, /* 66 = unused */
11	no_sys, /* 67 = unused */
12	no_sys, /* 68 = unused */
13	no_sys, /* 69 = unused */
14	mem_info, /* 70 = unused */
15	do_sigaction, /* 71 = sigaction */

/usr/src/servers/pm/alloc.c (στο τέλος του αρχείου)

```
void mem_info()
{
    int i,s;
    register struct hole *hp, *prev_ptr;

    prev_ptr = NIL_HOLE;
    hp = hole_head;

    i=0;
    s=0;

    while (hp != NIL_HOLE && hp->h_base < swap_base) {
        i++;
        s+=hp->h_len;

        //proairetiko. gia debugging to proshtetw k kano pali compile
        //dioti ta sholia den epitrepontai
        //printf("Hole %d with size %d clicks\n", I, hp->h_len);

        prev_ptr = hp;
        hp = hp->h_next;
    }
    s /=i;

    mp->mp_reply.m3_i1=i;
    mp->mp_reply.m3_i2=s/1024;
}
```

2. Κώδικες δοκιμαστικών προγραμμάτων test.c

```
#include <stdio.h>
#include <lib.h>
#include <unistd.h>

int main()
{
    message m;

    while(1) {
        _syscall(MM, 70, &m);

        printf("Holes: %d\tAverage Size: %d MB\n", m.m3_i1, m.m3_i2);

        sleep(1);
    }
}
```

3. Screenshot εκτέλεσης προγραμμάτων

test program

```
# ./test
Hole 1 with size 1 clicks
Hole 2 with size 2 clicks
Hole 3 with size 3 clicks
Hole 4 with size 5 clicks
Hole 5 with size 19 clicks
Hole 6 with size 14 clicks
Hole 7 with size 8 clicks
Hole 8 with size 11 clicks
Hole 9 with size 1 clicks
Hole 10 with size 1687 clicks
Hole 11 with size 55785 clicks
Holes: 11      Average Size: 5 MB
# _
```

Πρόβλημα 2

First - fit

1. Τροποποιήσεις Minix

Λίστα τροποποιηθέντων αρχείων Minix

/usr/src/servers/pm/alloc.c

Τροποποιήσεις αρχείων Minix

/usr/src/servers/pm/alloc.c

Line	Code
82	<code>printf("*---Hole requested %d clicks\n", clicks);</code>
83	<code>printf("-*---Hole found with size %d clicks\n", hp->h_len);</code>
84	<code>hp->h_len -= clicks; /* ditto */</code>
85	<code>printf("--*-Remaining clicks %d\n", hp->h_len);</code>

Screenshot first - fit

```
# ./programA
----Hole requested 44 clicks
----Hole found with size 1687 clicks
----Remaining clicks 1643
----Hole requested 34 clicks
----Hole found with size 36 clicks
----Remaining clicks 2
_
```

Best - fit

1. Τροποποιήσεις Minix

Λίστα τροποποιηθέντων αρχείων Minix

/usr/src/servers/pm/alloc.c

Τροποποιήσεις αρχείων Minix

/usr/src/servers/pm/alloc.c (changed alloc_mem)

```
PUBLIC phys_clicks alloc_mem(clicks)
phys_clicks clicks;          /* amount of memory requested */
{
/* Allocate a block of memory from the free list using first fit. The block
 * consists of a sequence of contiguous bytes, whose length in clicks is
 * given by 'clicks'. A pointer to the block is returned. The block is
 * always on a click boundary. This procedure is called when memory is
 * needed for FORK or EXEC. Swap other processes out if needed.*/
register struct hole *hp, *prev_ptr, *shp, *sprev_ptr;
phys_clicks old_base;
int min;
int found;

do {
    prev_ptr = NIL_HOLE;
    hp = hole_head;
    shp = NIL_HOLE;
    sprev_ptr = NIL_HOLE;
    min = 99999999;
    found=0;

    while (hp != NIL_HOLE && hp->h_base < swap_base) {
        if (hp->h_len >= clicks && hp->h_len-clicks < min) {
            found=1;
            sprev_ptr = prev_ptr;
            shp = hp;
            min = hp->h_len-clicks;
        }
        prev_ptr = hp;
        hp = hp->h_next;
    }
    if(found==1){
        /* We found a hole that is the best. Use it. */
        old_base = shp->h_base; /* remember where it started */
        shp->h_base += clicks; /* bite a piece off */
        printf("----Hole requested %d clicks\n", clicks);
        printf("----Hole found with size %d clicks\n", shp->h_len);
        shp->h_len -= clicks; /* ditto */
        printf("---Remaining clicks %d\n", shp->h_len);

        /* Remember new high watermark of used memory. */
        if(shp->h_base > high_watermark)
            high_watermark = shp->h_base;

        /* Delete the hole if used up completely. */
        if (shp->h_len == 0) del_slot(sprev_ptr, shp);

        /* Return the start address of the acquired block. */
        return(old_base);
    }
} while (swap_out()); /* try to swap some other process out */
return(NO_MEM);
}
```

Screenshot best - fit

```
# ./programΑ
----Hole requested 44 clicks
----Hole found with size 44 clicks
----Remaining clicks 0
----Hole requested 34 clicks
----Hole found with size 44 clicks
----Remaining clicks 10
_
```

Worst - fit

1. Τροποποιήσεις Minix

Λίστα τροποποιηθέντων αρχείων Minix

/usr/src/servers/pm/alloc.c

Τροποποιήσεις αρχείων Minix

/usr/src/servers/pm/alloc.c

```
PUBLIC phys_clicks alloc_mem(clicks)
phys_clicks clicks;          /* amount of memory requested */
{
/* Allocate a block of memory from the free list using first fit. The block
 * consists of a sequence of contiguous bytes, whose length in clicks is
 * given by 'clicks'. A pointer to the block is returned. The block is
 * always on a click boundary. This procedure is called when memory is
 * needed for FORK or EXEC. Swap other processes out if needed.
 */
register struct hole *hp, *prev_ptr, *shp, *sprev_ptr;
phys_clicks old_base;
int min;
int found;

do {
    prev_ptr = NIL_HOLE;
    hp = hole_head;
    shp = NIL_HOLE;
    sprev_ptr = NIL_HOLE;
    max = 0;
    found=0;

    while (hp != NIL_HOLE && hp->h_base < swap_base) {
        if (hp->h_len >= clicks && hp->h_len-clicks > max ) {
            found=1;
            sprev_ptr = prev_ptr;
            shp = hp;
            max = hp->h_len-clicks;
        }
        prev_ptr = hp;
        hp = hp->h_next;
    }
    if(found==1) {
        /* We found a hole that is big enough. Use it. */
        old_base = shp->h_base; /* remember where it started */
        shp->h_base += clicks; /* bite a piece off */
        printf("*---Hole requested %d clicks\n", clicks);
        printf("*---Hole found with size %d clicks\n", shp->h_len);
        shp->h_len -= clicks; /* ditto */
    }
}
```

```

printf("--*-Remaining clicks %d\n", shp->h_len);

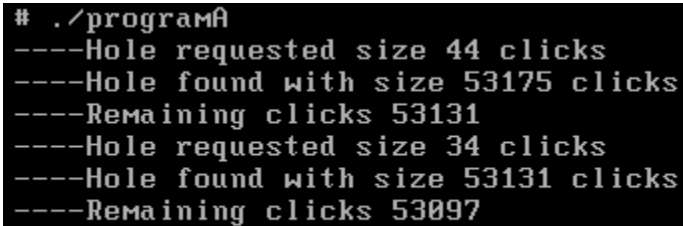
/* Remember new high watermark of used memory. */
if(shp->h_base > high_watermark)
high_watermark = shp->h_base;

/* Delete the hole if used up completely. */
if (shp->h_len == 0) del_slot(sprev_ptr, shp);

/* Return the start address of the acquired block. */
return(old_base);
}
} while (swap_out());          /* try to swap some other process out */
return(NO_MEM);
}

```

Screenshot worst – fit



```

# ./programA
----Hole requested size 44 clicks
----Hole found with size 53175 clicks
----Remaining clicks 53131
----Hole requested size 34 clicks
----Hole found with size 53131 clicks
----Remaining clicks 53097
_

```

Πρόβλημα 3

1. Σχόλια για τη συμπεριφορά των αλγορίθμων

Εκτελώντας το πρόγραμμα test αλλά και με κάθε εκτέλεση προγράμματος A ή B δεσμεύονται holes τόσο για το πρόγραμμα όσο και για την νέα διεργασία του shell έτσι δεν είναι ξεκάθαρο ποιο hole πήρε τελικά το συγκεκριμένο hole παρα μόνο με μετρήσεις του μεγέθους των holes ξεχωριστά. Επίσης όταν χρησιμοποιείται διαφορετικός αλγόριθμος allocation η αρχική κατάσταση του συστήματος(πρώτο screenshot σε κάθε περίπτωση) διαφέρει αφού η συγκεκριμένη πολιτική διανέμει την μνήμη κατα το init. Επομένως τα screenshot δεν είναι άμεσα συγκρίσιμα μεταξύ τους.

First – fit

Η κατάσταση των holes μετά το boot χρησιμοποιώντας τον first-fit αλγόριθμο. Χαρακτηριστικό του first fit αλγορίθμου ανάθεσης μνήμης είναι ότι αφήνει κατα μέσο όρο τρύπες μεσαίου μεγέθους και δεν αξιοποιεί την μνήμη με τον βέλτιστρο τρόπο.

```
# ./test
Hole 1 with size 1 clicks
Hole 2 with size 2 clicks
Hole 3 with size 3 clicks
Hole 4 with size 5 clicks
Hole 5 with size 19 clicks
Hole 6 with size 14 clicks
Hole 7 with size 8 clicks
Hole 8 with size 11 clicks
Hole 9 with size 1 clicks
Hole 10 with size 1687 clicks
Hole 11 with size 55785 clicks
```

Η κατάσταση των holes μετά απο την εκκίνηση των προγραμμάτων programB και programA όπως ζητείται. Φαίνεται ότι όλες οι διεργασίες κατέλαβαν χώρο απο το μόνο επαρκώς μεγάλο hole το 10.

```
# ./test
Hole 1 with size 1 clicks
Hole 2 with size 2 clicks
Hole 3 with size 3 clicks
Hole 4 with size 5 clicks
Hole 5 with size 19 clicks
Hole 6 with size 14 clicks
Hole 7 with size 8 clicks
Hole 8 with size 11 clicks
Hole 9 with size 44 clicks
Hole 10 with size 1469 clicks
Hole 11 with size 55785 clicks
—
```

Η κατάσταση των holes αφού τερματίστηκαν 2 διεργασίες (μία A και μια B) και εκτελέστηκε μια νέα A. Όπως φαίνεται απελευθερώνοντας χώρο 2 διεργασιών επιλέγεται για την νέα διεργασία το πρώτο hole που την χωράει.

```
# ./test
Hole 1 with size 1 clicks
Hole 2 with size 2 clicks
Hole 3 with size 3 clicks
Hole 4 with size 5 clicks
Hole 5 with size 19 clicks
Hole 6 with size 14 clicks
Hole 7 with size 8 clicks
Hole 8 with size 11 clicks
Hole 9 with size 78 clicks
Hole 10 with size 1469 clicks
Hole 11 with size 55785 clicks
—
```

Best– fit

Η κατάσταση των holes μετά το boot χρησιμοποιώντας τον best-fit αλγόριθμο. Χαρακτηριστικό του best fit αλγορίθμου ανάθεσης μνήμης είναι ότι αφήνει κατα μέσο όρο τρύπες μικρού μεγέθους που κατα πάσα πιθανότητα δεν μπορούν να αξιοποιηθούν αλλά βελτιστοποιεί την χρήση μνήμης Αφήνοντας λιγότερο κατακερματισμένα τα holes.

```
# ./test
Hole 1 with size 44 clicks
Hole 2 with size 2 clicks
Hole 3 with size 9 clicks
Hole 4 with size 2 clicks
Hole 5 with size 3 clicks
Hole 6 with size 8 clicks
Hole 7 with size 2 clicks
Hole 8 with size 21 clicks
Hole 9 with size 1643 clicks
Hole 10 with size 55802 clicks
-
```

Η κατάσταση των holes μετά απο την εκκίνηση των προγραμμάτων programB και programA όπως ζητείται. Φαίνεται ότι όλες οι διεργασίες κατέλαβαν χώρο απο το μόνο επαρκώς μεγάλο hole το 9 όπως και πριν.

```
# ./test
Hole 1 with size 44 clicks
Hole 2 with size 2 clicks
Hole 3 with size 8 clicks
Hole 4 with size 2 clicks
Hole 5 with size 3 clicks
Hole 6 with size 8 clicks
Hole 7 with size 2 clicks
Hole 8 with size 21 clicks
Hole 9 with size 1469 clicks
Hole 10 with size 55802 clicks
-
```

Η κατάσταση των holes αφού τερματίστηκαν 2 διεργασίες (μία A και μια B) και εκτελέστηκε μια νέα A. Όπως φαίνεται απελευθερώνοντας χώρο 2 διεργασιών επιλέγεται για την νέα διεργασία όχι το πρώτο hole που την χωράει.(το 1 με 44 clicks) αλλά αυτό που απελευθερώθηκε απο την προηγούμενη διεργασία A.

```
# ./test
Hole 1 with size 44 clicks
Hole 2 with size 2 clicks
Hole 3 with size 8 clicks
Hole 4 with size 2 clicks
Hole 5 with size 3 clicks
Hole 6 with size 8 clicks
Hole 7 with size 2 clicks
Hole 8 with size 21 clicks
Hole 9 with size 34 clicks
Hole 10 with size 2 clicks
Hole 11 with size 1504 clicks
Hole 12 with size 55802 clicks
-
```


Worst-fit

Η κατάσταση των holes μετά το boot χρησιμοποιώντας τον worst-fit αλγόριθμο. Χαρακτηριστικό του worst fit αλγορίθμου ανάθεσης μνήμης είναι ότι αφήνει πολλές τρύπες αυξάνοντας το fragmentation αλλά δεν αφήνει πολλές αναξιοποίητες τρύπες μικρού μεγέθους.

```
Hole 3 with size 128
Hole 4 with size 128
Hole 5 with size 128
Hole 6 with size 286
Hole 7 with size 202
Hole 8 with size 250
Hole 9 with size 250
Hole 10 with size 217
Hole 11 with size 173
Hole 12 with size 129
Hole 13 with size 129
Hole 14 with size 44
Hole 15 with size 88
Hole 16 with size 44
Hole 17 with size 16
Hole 18 with size 44
Hole 19 with size 1854
Hole 20 with size 3
Hole 21 with size 8
Hole 22 with size 3
Hole 23 with size 3
Hole 24 with size 10
Hole 25 with size 44
Hole 26 with size 53096
```

Η κατάσταση των holes μετά απο την εκκίνηση των προγραμμάτων programB και programA όπως ζητείται. Σε αυτήν την περίπτωση όλες οι διεργασίες παίρνουν χώρο απο το μεγαλύτερο hole(worst fit) το τελευταίο.

```
Hole 8 with size 250
Hole 9 with size 250
Hole 10 with size 217
Hole 11 with size 173
Hole 12 with size 129
Hole 13 with size 129
Hole 14 with size 44
Hole 15 with size 88
Hole 16 with size 44
Hole 17 with size 16
Hole 18 with size 44
Hole 19 with size 1854
Hole 20 with size 3
Hole 21 with size 8
Hole 22 with size 3
Hole 23 with size 3
Hole 24 with size 10
Hole 25 with size 44
Hole 26 with size 44
Hole 27 with size 44
Hole 28 with size 44
Hole 29 with size 44
Hole 30 with size 44
Hole 31 with size 52701
```

—

Η κατάσταση των holes αφού τερματίστηκαν 2 διεργασίες (μία A και μια B) και εκτελέστηκε μια νέα A. Ο χώρος που απελευθερώνεται απο τις δύο διεργασίες δεν επηρεάζει την επιλογή του αλγορίθμου καθώς θα επιλεγεί ξανά το τελευταίο hole, αφού αυτό θα αφήσει τον περισσότερο χώρο μετά την ανάθεση.

```
Hole 7 with size 202
Hole 8 with size 250
Hole 9 with size 250
Hole 10 with size 217
Hole 11 with size 173
Hole 12 with size 129
Hole 13 with size 129
Hole 14 with size 44
Hole 15 with size 88
Hole 16 with size 44
Hole 17 with size 16
Hole 18 with size 44
Hole 19 with size 1854
Hole 20 with size 3
Hole 21 with size 8
Hole 22 with size 3
Hole 23 with size 3
Hole 24 with size 10
Hole 25 with size 44
Hole 26 with size 122
Hole 27 with size 125
Hole 28 with size 44
Hole 29 with size 44
Hole 30 with size 52620
```