4η Άσκηση στα Λειτουργικά Συστήματα

6ο Εξάμηνο, Ακαδημαϊκή περίοδος 2020 – 2021

Ονοματεπώνυμο	Αριθμός Μητρώου
Στεφανάκης Γεώργιος	el18436
Τζανακάκης Αλέξανδρος	el18431

1.1 Κλήσεις συστήματος και μηχανισμοί του ΛΣ για τη διαχείριση της εικονικής μνήμης

Στην παρούσα άσκηση μας ζητήθηκε να συμπληρώσουμε με τις κατάλληλες εντολές τον σκελετό του προγράμματος `mmap.c` ώστε να εκτελούνται σωστά τα ερωτήματα της εκφώνησης. Η τελική έξοδος του προγράμματος φαίνεται παρακάτω.

```
oslaba35@os-node1:~/exc-4/mmap$ ./mmap

Step 1: Print the virtual address space map of this process [12217].


Virtual Memory Map of process [12217]:
00400000-00403000 r-xp 00000000 00:21 20454631
/home/oslab/oslaba35/exc-4/mmap/mmap
00602000-00603000 rw-p 00002000 00:21 20454631
/home/oslab/oslaba35/exc-4/mmap/mmap
00e7c000-00e9d000 rw-p 00000000 00:00 0                                  [heap]
7f1c151dc000-7f1c1537d000 r-xp 00000000 08:01 6032227                    /lib/x86_64-
linux-gnu/libc-2.19.so
7f1c1537d000-7f1c1557d000 ---p 001a1000 08:01 6032227                    /lib/x86_64-
linux-gnu/libc-2.19.so
7f1c1557d000-7f1c15581000 r--p 001a1000 08:01 6032227                    /lib/x86_64-
linux-gnu/libc-2.19.so
7f1c15581000-7f1c15583000 rw-p 001a5000 08:01 6032227                    /lib/x86_64-
linux-gnu/libc-2.19.so
7f1c15583000-7f1c15587000 rw-p 00000000 00:00 0
7f1c15587000-7f1c155a8000 r-xp 00000000 08:01 6032224                    /lib/x86_64-
linux-gnu/ld-2.19.so
7f1c1579a000-7f1c1579d000 rw-p 00000000 00:00 0
7f1c157a2000-7f1c157a7000 rw-p 00000000 00:00 0
7f1c157a7000-7f1c157a8000 r--p 00020000 08:01 6032224                    /lib/x86_64-
linux-gnu/ld-2.19.so
7f1c157a8000-7f1c157a9000 rw-p 00021000 08:01 6032224                    /lib/x86_64-
linux-gnu/ld-2.19.so
7f1c157a9000-7f1c157aa000 rw-p 00000000 00:00 0
7ffd940b7000-7ffd940d8000 rw-p 00000000 00:00 0                          [stack]
7ffd94169000-7ffd9416c000 r--p 00000000 00:00 0                          [vvar]
7ffd9416c000-7ffd9416e000 r-xp 00000000 00:00 0                          [vdso]
ffffffffff600000-ffffffffff601000 r-xp 00000000 00:00 0                  [vsyscall]
-----------------------------------------------------------


Step 2: Use mmap(2) to allocate a private buffer of size equal to 1 page and print the VM
map again.


Virtual Memory Map of process [12217]:
00400000-00403000 r-xp 00000000 00:21 20454631
/home/oslab/oslaba35/exc-4/mmap/mmap
```

```
00602000-00603000 rw-p 00002000 00:21 20454631
/home/oslab/oslaba35/exc-4/mmap/mmap
00e7c000-00e9d000 rw-p 00000000 00:00 0                                  [heap]
7f1c151dc000-7f1c1537d000 r-xp 00000000 08:01 6032227                    /lib/x86_64-
linux-gnu/libc-2.19.so
7f1c1537d000-7f1c1557d000 ---p 001a1000 08:01 6032227                    /lib/x86_64-
linux-gnu/libc-2.19.so
7f1c1557d000-7f1c15581000 r--p 001a1000 08:01 6032227                    /lib/x86_64-
linux-gnu/libc-2.19.so
7f1c15581000-7f1c15583000 rw-p 001a5000 08:01 6032227                    /lib/x86_64-
linux-gnu/libc-2.19.so
7f1c15583000-7f1c15587000 rw-p 00000000 00:00 0
7f1c15587000-7f1c155a8000 r-xp 00000000 08:01 6032224                    /lib/x86_64-
linux-gnu/ld-2.19.so
7f1c1579a000-7f1c1579d000 rw-p 00000000 00:00 0
7f1c157a1000-7f1c157a7000 rw-p 00000000 00:00 0
7f1c157a7000-7f1c157a8000 r--p 00020000 08:01 6032224                    /lib/x86_64-
linux-gnu/ld-2.19.so
7f1c157a8000-7f1c157a9000 rw-p 00021000 08:01 6032224                    /lib/x86_64-
linux-gnu/ld-2.19.so
7f1c157a9000-7f1c157aa000 rw-p 00000000 00:00 0
7ffd940b7000-7ffd940d8000 rw-p 00000000 00:00 0                          [stack]
7ffd94169000-7ffd9416c000 r--p 00000000 00:00 0                          [vvar]
7ffd9416c000-7ffd9416e000 r-xp 00000000 00:00 0                          [vdso]
ffffffffff600000-ffffffffff601000 r-xp 00000000 00:00 0                  [vsyscall]
-----------------------------------------------------------
```

Step 3: Find and print the physical address of the buffer in main memory. What do you see?

```
VA[0x7f1c157a2000] is not mapped; no physical memory allocated.
```

Step 4: Initialize your buffer with zeros and repeat Step 3. What happened?

```
42dd7000
```

Step 5: Use mmap(2) to read and print file.txt. Print the new mapping information that has been created.

```
Hello everyone!

Virtual Memory Map of process [12217]:
00400000-00403000 r-xp 00000000 00:21 20454631
/home/oslab/oslaba35/exc-4/mmap/mmap
00602000-00603000 rw-p 00002000 00:21 20454631
/home/oslab/oslaba35/exc-4/mmap/mmap
00e7c000-00e9d000 rw-p 00000000 00:00 0                                  [heap]
7f1c151dc000-7f1c1537d000 r-xp 00000000 08:01 6032227                    /lib/x86_64-
linux-gnu/libc-2.19.so
7f1c1537d000-7f1c1557d000 ---p 001a1000 08:01 6032227                    /lib/x86_64-
linux-gnu/libc-2.19.so
7f1c1557d000-7f1c15581000 r--p 001a1000 08:01 6032227                    /lib/x86_64-
linux-gnu/libc-2.19.so
7f1c15581000-7f1c15583000 rw-p 001a5000 08:01 6032227                    /lib/x86_64-
linux-gnu/libc-2.19.so
7f1c15583000-7f1c15587000 rw-p 00000000 00:00 0
7f1c15587000-7f1c155a8000 r-xp 00000000 08:01 6032224                    /lib/x86_64-
linux-gnu/ld-2.19.so
7f1c1579a000-7f1c1579d000 rw-p 00000000 00:00 0
7f1c157a0000-7f1c157a1000 rw-p 00000000 00:00 0
```

```
7f1c157a1000-7f1c157a2000 r--s 00000000 00:21 20455228
/home/oslab/oslaba35/exc-4/mmap/file.txt
7f1c157a2000-7f1c157a7000 rw-p 00000000 00:00 0
7f1c157a7000-7f1c157a8000 r--p 00020000 08:01 6032224                    /lib/x86_64-
linux-gnu/ld-2.19.so
7f1c157a8000-7f1c157a9000 rw-p 00021000 08:01 6032224                    /lib/x86_64-
linux-gnu/ld-2.19.so
7f1c157a9000-7f1c157aa000 rw-p 00000000 00:00 0
7ffd940b7000-7ffd940d8000 rw-p 00000000 00:00 0                          [stack]
7ffd94169000-7ffd9416c000 r--p 00000000 00:00 0                          [vvar]
7ffd9416c000-7ffd9416e000 r-xp 00000000 00:00 0                          [vdso]
ffffffffff600000-ffffffffff601000 r-xp 00000000 00:00 0                  [vsyscall]
-----------------------------------------------------------

7f1c157a1000-7f1c157a2000 r--s 00000000 00:21 20455228
/home/oslab/oslaba35/exc-4/mmap/file.txt
```

Step 6: Use mmap(2) to allocate a shared buffer of size equal to 1 page. Initialize the buffer and print the new mapping information that has been created.

```
Virtual Memory Map of process [12217]:
00400000-00403000 r-xp 00000000 00:21 20454631
/home/oslab/oslaba35/exc-4/mmap/mmap
00602000-00603000 rw-p 00002000 00:21 20454631
/home/oslab/oslaba35/exc-4/mmap/mmap
00e7c000-00e9d000 rw-p 00000000 00:00 0                                  [heap]
7f1c151dc000-7f1c1537d000 r-xp 00000000 08:01 6032227                    /lib/x86_64-
linux-gnu/libc-2.19.so
7f1c1537d000-7f1c1557d000 ---p 001a1000 08:01 6032227                    /lib/x86_64-
linux-gnu/libc-2.19.so
7f1c1557d000-7f1c15581000 r--p 001a1000 08:01 6032227                    /lib/x86_64-
linux-gnu/libc-2.19.so
7f1c15581000-7f1c15583000 rw-p 001a5000 08:01 6032227                    /lib/x86_64-
linux-gnu/libc-2.19.so
7f1c15583000-7f1c15587000 rw-p 00000000 00:00 0
7f1c15587000-7f1c155a8000 r-xp 00000000 08:01 6032224                    /lib/x86_64-
linux-gnu/ld-2.19.so
7f1c1579a000-7f1c1579d000 rw-p 00000000 00:00 0
7f1c1579f000-7f1c157a0000 rw-p 00000000 00:00 0
7f1c157a0000-7f1c157a1000 rw-s 00000000 00:04 2307531                    /dev/zero
(deleted)
7f1c157a1000-7f1c157a2000 r--s 00000000 00:21 20455228
/home/oslab/oslaba35/exc-4/mmap/file.txt
7f1c157a2000-7f1c157a7000 rw-p 00000000 00:00 0
7f1c157a7000-7f1c157a8000 r--p 00020000 08:01 6032224                    /lib/x86_64-
linux-gnu/ld-2.19.so
7f1c157a8000-7f1c157a9000 rw-p 00021000 08:01 6032224                    /lib/x86_64-
linux-gnu/ld-2.19.so
7f1c157a9000-7f1c157aa000 rw-p 00000000 00:00 0
7ffd940b7000-7ffd940d8000 rw-p 00000000 00:00 0                          [stack]
7ffd94169000-7ffd9416c000 r--p 00000000 00:00 0                          [vvar]
7ffd9416c000-7ffd9416e000 r-xp 00000000 00:00 0                          [vdso]
ffffffffff600000-ffffffffff601000 r-xp 00000000 00:00 0                  [vsyscall]
-----------------------------------------------------------

7f1c157a0000-7f1c157a1000 rw-s 00000000 00:04 2307531                    /dev/zero
(deleted)
72cf000
```

Step 7: Print parent's and child's map.

```
Parent

Virtual Memory Map of process [12217]:
00400000-00403000 r-xp 00000000 00:21 20454631
/home/oslab/oslaba35/exc-4/mmap/mmap
00602000-00603000 rw-p 00002000 00:21 20454631
/home/oslab/oslaba35/exc-4/mmap/mmap
00e7c000-00e9d000 rw-p 00000000 00:00 0                                  [heap]
7f1c151dc000-7f1c1537d000 r-xp 00000000 08:01 6032227                    /lib/x86_64-
linux-gnu/libc-2.19.so
7f1c1537d000-7f1c1557d000 ---p 001a1000 08:01 6032227                    /lib/x86_64-
linux-gnu/libc-2.19.so
7f1c1557d000-7f1c15581000 r--p 001a1000 08:01 6032227                    /lib/x86_64-
linux-gnu/libc-2.19.so
7f1c15581000-7f1c15583000 rw-p 001a5000 08:01 6032227                    /lib/x86_64-
linux-gnu/libc-2.19.so
7f1c15583000-7f1c15587000 rw-p 00000000 00:00 0
7f1c15587000-7f1c155a8000 r-xp 00000000 08:01 6032224                    /lib/x86_64-
linux-gnu/ld-2.19.so
7f1c1579a000-7f1c1579d000 rw-p 00000000 00:00 0
7f1c1579f000-7f1c157a0000 rw-p 00000000 00:00 0
7f1c157a0000-7f1c157a1000 rw-s 00000000 00:04 2307531                    /dev/zero
(deleted)
7f1c157a1000-7f1c157a2000 r--s 00000000 00:21 20455228
/home/oslab/oslaba35/exc-4/mmap/file.txt
7f1c157a2000-7f1c157a7000 rw-p 00000000 00:00 0
7f1c157a7000-7f1c157a8000 r--p 00020000 08:01 6032224                    /lib/x86_64-
linux-gnu/ld-2.19.so
7f1c157a8000-7f1c157a9000 rw-p 00021000 08:01 6032224                    /lib/x86_64-
linux-gnu/ld-2.19.so
7f1c157a9000-7f1c157aa000 rw-p 00000000 00:00 0
7ffd940b7000-7ffd940d8000 rw-p 00000000 00:00 0                          [stack]
7ffd94169000-7ffd9416c000 r--p 00000000 00:00 0                          [vvar]
7ffd9416c000-7ffd9416e000 r-xp 00000000 00:00 0                          [vdso]
ffffffffff600000-ffffffffff601000 r-xp 00000000 00:00 0                  [vsyscall]
---------------------------------------------------------

Child

Virtual Memory Map of process [12218]:
00400000-00403000 r-xp 00000000 00:21 20454631
/home/oslab/oslaba35/exc-4/mmap/mmap
00602000-00603000 rw-p 00002000 00:21 20454631
/home/oslab/oslaba35/exc-4/mmap/mmap
00e7c000-00e9d000 rw-p 00000000 00:00 0                                  [heap]
7f1c151dc000-7f1c1537d000 r-xp 00000000 08:01 6032227                    /lib/x86_64-
linux-gnu/libc-2.19.so
7f1c1537d000-7f1c1557d000 ---p 001a1000 08:01 6032227                    /lib/x86_64-
linux-gnu/libc-2.19.so
7f1c1557d000-7f1c15581000 r--p 001a1000 08:01 6032227                    /lib/x86_64-
linux-gnu/libc-2.19.so
7f1c15581000-7f1c15583000 rw-p 001a5000 08:01 6032227                    /lib/x86_64-
linux-gnu/libc-2.19.so
7f1c15583000-7f1c15587000 rw-p 00000000 00:00 0
7f1c15587000-7f1c155a8000 r-xp 00000000 08:01 6032224                    /lib/x86_64-
linux-gnu/ld-2.19.so
7f1c1579a000-7f1c1579d000 rw-p 00000000 00:00 0
7f1c1579f000-7f1c157a0000 rw-p 00000000 00:00 0
7f1c157a0000-7f1c157a1000 rw-s 00000000 00:04 2307531                    /dev/zero
(deleted)
```

```
7f1c157a1000-7f1c157a2000 r--s 00000000 00:21 20455228
/home/oslab/oslaba35/exc-4/mmap/file.txt
7f1c157a2000-7f1c157a7000 rw-p 00000000 00:00 0
7f1c157a7000-7f1c157a8000 r--p 00020000 08:01 6032224                    /lib/x86_64-
linux-gnu/ld-2.19.so
7f1c157a8000-7f1c157a9000 rw-p 00021000 08:01 6032224                    /lib/x86_64-
linux-gnu/ld-2.19.so
7f1c157a9000-7f1c157aa000 rw-p 00000000 00:00 0
7ffd940b7000-7ffd940d8000 rw-p 00000000 00:00 0                          [stack]
7ffd94169000-7ffd9416c000 r--p 00000000 00:00 0                          [vvar]
7ffd9416c000-7ffd9416e000 r-xp 00000000 00:00 0                          [vdso]
ffffffffff600000-ffffffffff601000 r-xp 00000000 00:00 0                  [vsyscall]
-----------------------------------------------------------
```

Step 8: Find the physical address of the private heap buffer (main) for both the parent and the child.

```
Parent
42dd7000
Child
42dd7000
```

Step 9: Write to the private buffer from the child and repeat step 8. What happened?

```
Parent
42dd7000
Child
693d000
```

Step 10: Write to the shared heap buffer (main) from child and get the physical address for both the parent and the child. What happened?

```
Parent
72cf000
Child
72cf000
```

Step 11: Disable writing on the shared buffer for the child. Verify through the maps for the parent and the child.

Parent

```
Virtual Memory Map of process [12217]:
00400000-00403000 r-xp 00000000 00:21 20454631
/home/oslab/oslaba35/exc-4/mmap/mmap
00602000-00603000 rw-p 00002000 00:21 20454631
/home/oslab/oslaba35/exc-4/mmap/mmap
00e7c000-00e9d000 rw-p 00000000 00:00 0                                  [heap]
7f1c151dc000-7f1c1537d000 r-xp 00000000 08:01 6032227                    /lib/x86_64-
linux-gnu/libc-2.19.so
7f1c1537d000-7f1c1557d000 ---p 001a1000 08:01 6032227                    /lib/x86_64-
linux-gnu/libc-2.19.so
7f1c1557d000-7f1c15581000 r--p 001a1000 08:01 6032227                    /lib/x86_64-
linux-gnu/libc-2.19.so
7f1c15581000-7f1c15583000 rw-p 001a5000 08:01 6032227                    /lib/x86_64-
linux-gnu/libc-2.19.so
7f1c15583000-7f1c15587000 rw-p 00000000 00:00 0
7f1c15587000-7f1c155a8000 r-xp 00000000 08:01 6032224                    /lib/x86_64-
linux-gnu/ld-2.19.so
7f1c1579a000-7f1c1579d000 rw-p 00000000 00:00 0
```

```
7f1c1579f000-7f1c157a0000 rw-p 00000000 00:00 0
7f1c157a0000-7f1c157a1000 rw-s 00000000 00:04 2307531                    /dev/zero
(deleted)
7f1c157a1000-7f1c157a2000 r--s 00000000 00:21 20455228
/home/oslab/oslaba35/exc-4/mmap/file.txt
7f1c157a2000-7f1c157a7000 rw-p 00000000 00:00 0
7f1c157a7000-7f1c157a8000 r--p 00020000 08:01 6032224                    /lib/x86_64-
linux-gnu/ld-2.19.so
7f1c157a8000-7f1c157a9000 rw-p 00021000 08:01 6032224                    /lib/x86_64-
linux-gnu/ld-2.19.so
7f1c157a9000-7f1c157aa000 rw-p 00000000 00:00 0
7ffd940b7000-7ffd940d8000 rw-p 00000000 00:00 0                          [stack]
7ffd94169000-7ffd9416c000 r--p 00000000 00:00 0                          [vvar]
7ffd9416c000-7ffd9416e000 r-xp 00000000 00:00 0                          [vdso]
ffffffffff600000-ffffffffff601000 r-xp 00000000 00:00 0                  [vsyscall]
-----------------------------------------------------

Child

Virtual Memory Map of process [12218]:
00400000-00403000 r-xp 00000000 00:21 20454631
/home/oslab/oslaba35/exc-4/mmap/mmap
00602000-00603000 rw-p 00002000 00:21 20454631
/home/oslab/oslaba35/exc-4/mmap/mmap
00e7c000-00e9d000 rw-p 00000000 00:00 0                                  [heap]
7f1c151dc000-7f1c1537d000 r-xp 00000000 08:01 6032227                    /lib/x86_64-
linux-gnu/libc-2.19.so
7f1c1537d000-7f1c1557d000 ---p 001a1000 08:01 6032227                    /lib/x86_64-
linux-gnu/libc-2.19.so
7f1c1557d000-7f1c15581000 r--p 001a1000 08:01 6032227                    /lib/x86_64-
linux-gnu/libc-2.19.so
7f1c15581000-7f1c15583000 rw-p 001a5000 08:01 6032227                    /lib/x86_64-
linux-gnu/libc-2.19.so
7f1c15583000-7f1c15587000 rw-p 00000000 00:00 0
7f1c15587000-7f1c155a8000 r-xp 00000000 08:01 6032224                    /lib/x86_64-
linux-gnu/ld-2.19.so
7f1c1579a000-7f1c1579d000 rw-p 00000000 00:00 0
7f1c1579f000-7f1c157a0000 rw-p 00000000 00:00 0
7f1c157a0000-7f1c157a1000 r--s 00000000 00:04 2307531                    /dev/zero
(deleted)
7f1c157a1000-7f1c157a2000 r--s 00000000 00:21 20455228
/home/oslab/oslaba35/exc-4/mmap/file.txt
7f1c157a2000-7f1c157a7000 rw-p 00000000 00:00 0
7f1c157a7000-7f1c157a8000 r--p 00020000 08:01 6032224                    /lib/x86_64-
linux-gnu/ld-2.19.so
7f1c157a8000-7f1c157a9000 rw-p 00021000 08:01 6032224                    /lib/x86_64-
linux-gnu/ld-2.19.so
7f1c157a9000-7f1c157aa000 rw-p 00000000 00:00 0
7ffd940b7000-7ffd940d8000 rw-p 00000000 00:00 0                          [stack]
7ffd94169000-7ffd9416c000 r--p 00000000 00:00 0                          [vvar]
7ffd9416c000-7ffd9416e000 r-xp 00000000 00:00 0                          [vdso]
ffffffffff600000-ffffffffff601000 r-xp 00000000 00:00 0                  [vsyscall]
-----------------------------------------------------

oslaba35@os-node1:~/exc-4/mmap$
```

Αξιοσημείωτες παρατηρήσεις για κάθε βήμα:

2) Παρατηρούμε ότι στον χάρτη που τυπώσαμε σε αυτό το βήμα, στην δέκατη γραμμή έχει συγχωνευτεί μία σελίδα την οποία δεσμεύσαμε μέσω της απεικόνισης που κατασκευάσαμε. Το λειτουργικό σύστημα επέκτεινε την απεικόνιση εκείνης της γραμμής κατά μία σελίδα από την αρχή της επειδή σηματοδοτήσαμε την νέα απεικόνιση ως private για τη συγκεκριμένη διεργασία.

3) Όπως περιμέναμε, από τη στιγμή που δεν έχουμε αρχικοποιήσει τον buffer η δεσμευμένη εικονική μνήμη δεν αντιστοιχεί ακόμα σε κάποιο πεδίο διευθύνσεων στην πραγματική μνήμη. Αυτό συμβαίνει καθώς εφαρμόζεται η τακτική Demand Paging, δηλαδή μόνο όταν μια διεργασία χρειαστεί να αποθηκεύσει δεδομένα σε μία σελίδα δημιουργείται το αντίστοιχο πλαίσιο στη φυσική μνήμη. Από τη στιγμή που δεν έχουμε χρησιμοποιήσει ακόμα τη μνήμη που δεσμεύσαμε δεν έχει γίνει η αντιστοίχιση.

4) Αφού λοιπόν αρχικοποιήσουμε τον buffer παρατηρούμε ότι πλέον υπάρχει αντιστοίχιση της απεικόνισής μας σε πραγματική διεύθυνση φυσικής μνήμης.

5) Αφού χρησιμοποιήσαμε την κλήση συστήματος `open()` για να ανοίξουμε το αρχείο κάναμε ένα shared mapping μεγέθους όσο και το μέγεθος του αρχείου με χρήση των κατάλληλων flag για ανάγνωση αρχείου. Τα περιεχόμενα του buffer είναι και οι χαρακτήρες που περιέχονται στο αρχείο, πράγμα που γίνεται αντιληπτό με την εκτύπωσή τους. Στον χάρτη απεικονίσεων παρατηρούμε ότι έχει δημιουργηθεί ένα νέο record που περιγράφει αυτή την απεικόνιση.

6) Σε αυτό το βήμα το entry του νέου shared buffer που κατασκευάσαμε εμφανίζεται αμέσως στον χάρτη απεικονίσεων καθώς αμέσως μετά την δημιουργία του τον αρχικοποιούμε.

7) Σε αυτό το σημείο δημιουργούμε μία ακόμη διεργασία. Η διεργασία γονέας εκτελεί τη ρουτίνα `parent()` ενώ η διεργασία παιδί εκτελεί τη ρουτίνα `child()`. Επειδή η μία διεργασία είναι αντίγραφο της άλλης, οι χάρτες απεικονίσεών τους οφείλουν να είναι πανομοιότυποι, πράγμα που επαληθεύεται από την έξοδο.

8) Αμέσως μετά το fork, η διεύθυνση στην οποία απεικονίζεται ο private buffer είναι η ίδια και στις δύο διεργασίες. Ακολουθείται λογική Copy on Write, δηλαδή, ενώ αρχικά οι δείκτες δείχνουν στην ίδια διεύθυνση, όταν μία από τις δύο διεργασίες προσπαθήσει να γράψει στον private buffer, μόνο τότε το λειτουργικό αντιγράφει τα περιεχόμενα της μνήμης σε διαφορετικό πεδίο διευθύνσεων ώστε να αποφεύγονται οι άσκοπες αντιγραφές (πχ. όταν έχουμε read only δεδομένα).

9) Αφού γράψουμε στον private buffer από το παιδί, βλέπουμε ότι η διεύθυνσή του μεταφέρεται σε άλλη διεύθυνση φυσικής μνήμης για το λόγο που εξηγήσαμε παραπάνω.

10) Προφανώς σε αυτή την περίπτωση ο shared buffer είναι στις ίδιες θέσεις μνήμης και στις δύο διεργασίες καθώς είναι διαμοιραζόμενος.

11) Αφού αλλάξουμε τα δικαιώματα του buffer στη διεργασία παιδί, παρατηρούμε ότι οι χάρτες των δύο διεργασιών διαφέρουν ως προς τα δικαιώματα αυτά. Συγκεκριμένα, το παιδί έχει μόνο δικαίωμα ανάγνωσης ενώ ο πατέρας συνεχίζει να έχει δικαιώματα ανάγνωσης και εγγραφής.

12) Τέλος, αποδεσμεύουμε όλες τις απεικονίσεις και κλείνουμε το αρχείο `file.txt`.

1.2 Παράλληλος υπολογισμός Mandelbrot με διεργασίες αντί για νήματα

Σε αυτό το ερώτημα ζητήθηκε να επαναλάβουμε το πρόβλημα της εκτύπωσης του Mandelbrot Set χρησιμοποιώντας αυτή τη φορά διεργασίες αντί για νήματα. Καθώς οι διεργασίες εν γένει δεν μοιράζονται κοινά δεδομένα, έγινε χρήση διαμοιραζόμενου buffer μέσω απεικόνισης το οποίο περιέχει την κυκλική δομή από σημαφόρους που εξηγήθηκε στην προηγούμενη αναφορά έτσι ώστε να επιτευχθεί ο συγχρονισμός της εκτύπωσης κάθε γραμμής. Επιπλέον, ζητήθηκε και μία υλοποίηση χωρίς συγχρονισμό κατά την οποία κάθε διεργασία κάνει τους υπολογισμούς των χρωμάτων που της αντιστοιχεί και τοποθετεί το αποτέλεσμα σε έναν διαμοιραζόμενο buffer μεγέθους όσοι και οι χαρακτήρες προς εκτύπωση. Έτσι καθίσταται αδιάφορη η προτεραιότητα με την οποία κάθε διεργασία τυπώνει μία γραμμή ενώ στο τέλος όλων των υπολογισμών γίνεται απλή εκτύπωση του buffer.

1) Η υλοποίηση με νήματα περιμένουμε να έχει καλύτερη επίδοση από την υλοποίηση με διεργασίες. Ο κύριος λόγος είναι επειδή κατά τη δημιουργία μίας διεργασίας γίνεται ολική αντιγραφή όλων των δεδομένων (program counter, registers, virtual memory κ.λπ.) και αυτό είναι αρκετά χρονοβόρο. Επιπλέον, γνωρίζουμε ότι η κλήση `mmap()` είναι αρκετά βαριά από μόνης της. Με νήματα καταφέρνουμε να παρακάμψουμε αυτή την αντιγραφή, παρόλα αυτά δεν παρατηρούμε μεγάλη διαφορά στο χρόνο εκτέλεσης των δύο εκδοχών του προγράμματος λόγω της απλότητάς του. Σε πιο περίπλοκα προγράμματα θα ήταν περισσότερο αισθητή η διαφορά στην απόδοση.

2) Αυτή τη φορά κατασκευάσαμε ένα shared buffer μεγέθους `x_chars * y_chars` ο οποίο προορίζεται να αποθηκεύσει τη χρωματική τιμή κάθε χαρακτήρα που επιθυμούμε να τυπώσουμε. Αφού δημιουργηθούν οι διεργασίες παιδία, κάνουν τους υπολογισμούς των γραμμών και ανανεώσουν τον buffer, η διεργασία γονέας τυπώνει τον buffer χρησιμοποιώντας τη γνωστή βοηθητική συνάρτηση `output_mandel_line()`. Ο συγχρονισμός εδώ επιτυγχάνεται απλά με τη χρήση πολλών wait() που είναι σε πλήθος όσες και οι διεργασίες παιδία που αναλαμβάνουν τον υπολογισμό. Σημαντικό είναι να τονίσουμε πως η περιοχή διαμοιραζόμενης μνήμης που δεσμεύουμε είναι μονοδιάστατη σε αντίθεση με τον δισδιάστατο πίνακα που θέλουμε να τυπώσουμε, πράγμα που το αντιμετωπίσαμε «σειριοποιώντας» τις γραμμές του πίνακα σε μία ($1^η$ γραμμή με indexes από 0 μέχρι x_chars – 1, $2^η$ γραμμή από x_chars μέχρι 2 * x_chars – 1 κ.λπ.).

Εάν ο buffer μας είχε μέγεθος NPROCS * x_chars αυτό θα σήμαινε ότι μπορεί κάθε διεργασία να αναλάβει NPROCS γραμμές και συνεπώς το πρόγραμμα θα ήταν πιο περίπλοκο. Με αυτό τον τρόπο, αφού μια διεργασία υπολογίσει και τυπώσει μία από τις γραμμές που της αντιστοιχούν θα κάνει raise(SIGSTOP) ώστε να αναστείλει τον εαυτό της. Όταν πλέον όλες οι διεργασίες έχουν ανασταλεί τότε ο buffer θα έχει γεμίσει με μία συνεχόμενη N-άδα γραμμών οι οποίες είναι έτοιμες για εκτύπωση. Η γονεϊκή διεργασία είναι υπεύθυνη να περιμένει τα παιδιά της να σταματήσουν και ύστερα να τυπώσει τα περιεχόμενα του buffer. Έπειτα, πρέπει να στείλει σήμα SIGCONT στα παιδία ώστε να υπολογιστούν οι επόμενες N γραμμές. Αυτή η διαδικασία θα επαναληφθεί NPROCS / y_chars φορές.

Πηγαίος Κώδικας

Μαζί με την παρούσα αναφορά επισυνάπτουμε και τους φακέλους `mmap`, `sync-mmap` στους οποίους συμπεριλαμβάνονται όλα τα αρχεία που μας δόθηκαν από την εκφώνηση της άσκησης, μεταρρυθμισμένα ώστε να λειτουργούν με τον επιθυμητό τρόπο. Τα αρχεία πηγαίου κώδικα τα οποία επεξεργαστήκαμε ονομάζονται `mmap.c`, `mandel-fork.c` και `mandel-buffer.c`. Επιπλέον έχουμε τροποποιήσει το αρχείο `Makefile` του δεύτερου ερωτήματος ώστε να παράγει και το εκτελέσιμο για την εκδοχή του προβλήματος Mandelbrot με χρήση buffer.