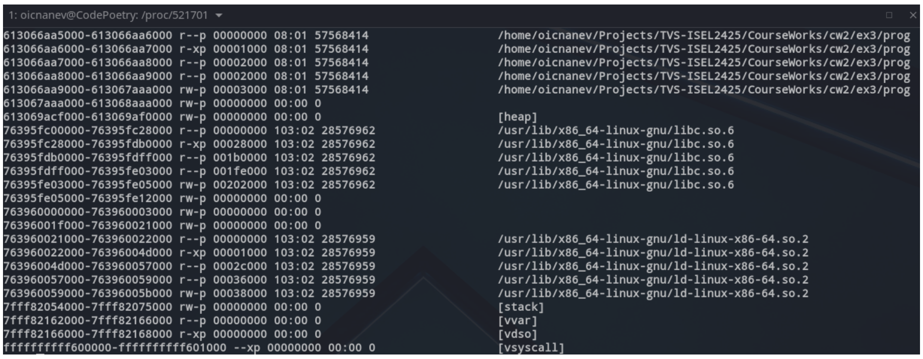
Explicação:

Mapeamento original - /proc/[pid]/maps



```
613066aa5000-613066aa6000 r--p 00000000 08:01 57568414                   /home/oicnanev/Projects/TVS-ISEL2425/CourseWorks/cw2/ex3/prog
613066aa6000-613066aa7000 r-xp 00001000 08:01 57568414                   /home/oicnanev/Projects/TVS-ISEL2425/CourseWorks/cw2/ex3/prog
613066aa7000-613066aa8000 r--p 00002000 08:01 57568414                   /home/oicnanev/Projects/TVS-ISEL2425/CourseWorks/cw2/ex3/prog
613066aa8000-613066aa9000 r--p 00002000 08:01 57568414                   /home/oicnanev/Projects/TVS-ISEL2425/CourseWorks/cw2/ex3/prog
613066aa9000-613067aaa000 rw-p 00003000 08:01 57568414                   /home/oicnanev/Projects/TVS-ISEL2425/CourseWorks/cw2/ex3/prog
613067aaa000-613068aaa000 rw-p 00000000 00:00 0
613069acf000-613069af0000 rw-p 00000000 00:00 0                          [heap]
76395fc00000-76395fc28000 r--p 00000000 103:02 28576962                  /usr/lib/x86_64-linux-gnu/libc.so.6
76395fc28000-76395fdb0000 r-xp 00028000 103:02 28576962                  /usr/lib/x86_64-linux-gnu/libc.so.6
76395fdb0000-76395fdff000 r--p 001b0000 103:02 28576962                  /usr/lib/x86_64-linux-gnu/libc.so.6
76395fdff000-76395fe03000 r--p 001fe000 103:02 28576962                  /usr/lib/x86_64-linux-gnu/libc.so.6
76395fe03000-76395fe05000 rw-p 00202000 103:02 28576962                  /usr/lib/x86_64-linux-gnu/libc.so.6
76395fe05000-76395fe12000 rw-p 00000000 00:00 0
763960000000-763960003000 rw-p 00000000 00:00 0
76396001f000-763960021000 rw-p 00000000 00:00 0
763960021000-763960022000 r--p 00000000 103:02 28576959                  /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
763960022000-76396004d000 r-xp 00001000 103:02 28576959                  /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
76396004d000-763960057000 r--p 0002c000 103:02 28576959                  /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
763960057000-763960059000 r--p 00036000 103:02 28576959                  /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
763960059000-76396005b000 rw-p 00038000 103:02 28576959                  /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
7fff82054000-7fff82075000 rw-p 00000000 00:00 0                          [stack]
7fff82162000-7fff82166000 r--p 00000000 00:00 0                          [vvar]
7fff82166000-7fff82168000 r-xp 00000000 00:00 0                          [vdso]
ffffffffff600000-ffffffffff601000 --xp 00000000 00:00 0                  [vsyscall]
```

Figure 1: mapeamento

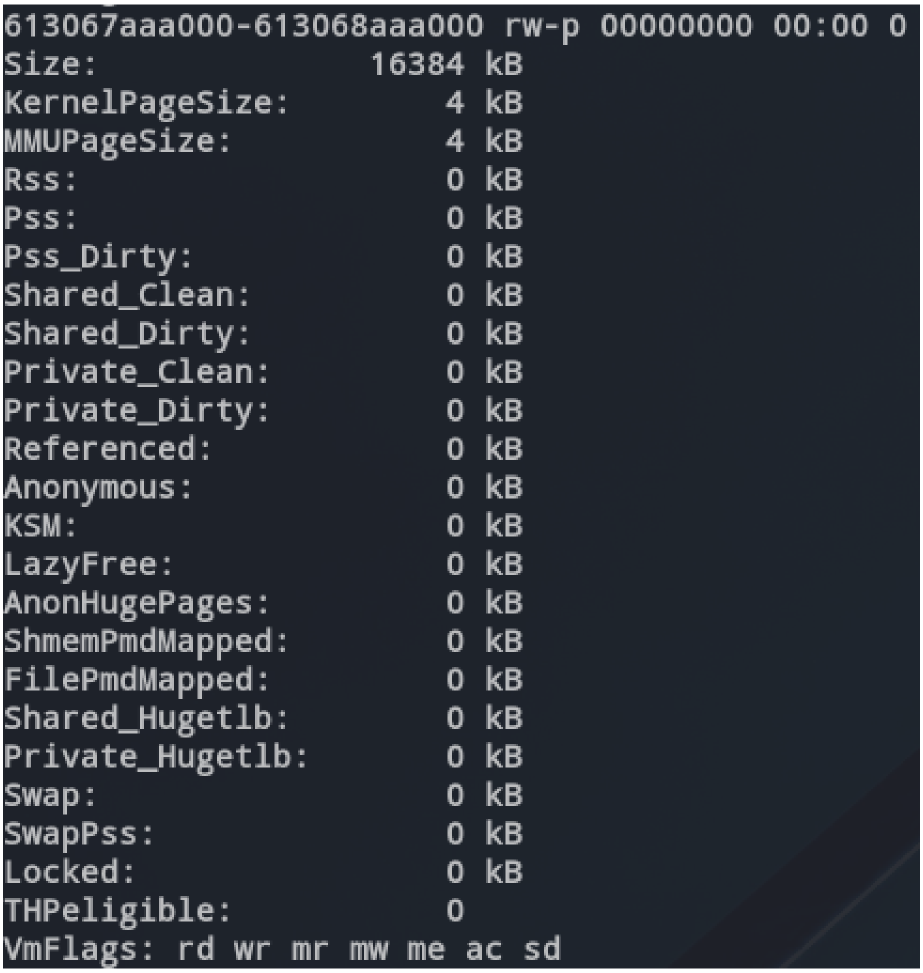
prog

- HEADER
  - r–p (read-only)
  - 0x6130 66AA 5000 - 0x6130 66AA 6000
- .text
  - r-xp (read-only and executable)
  - 0x6130 66AA 6000 - 0x6130 66AA 7000
- .rodata
  - r–p (read-only)
  - 0x6130 66AA 7000 - 0x6130 66AA 8000
- .got (?) - (armazena endereços de variáveis globais e funções usadas)
  - r–p (read-only)
  - 0x6130 66AA 8000 - 0x6130 66AA 9000
- .data
  - rw-p (read and write)
  - 0x6130 66AA 9000 0x6130 67AA A000
- .bss
  - rw-p (read and write)
  - 0x6130 67AA A000 - 0x6130 68AA A000
- heap
  - rw-p (read and write)
  - 0x6130 69AC F000 - 6130 69AF 0000

1. a. Increase the resident set (Rss) by about 3MB in the .bss region:

```
for (int i = 0; i < DATA_3MB; i++) {
    info[i] = 1;
};
```

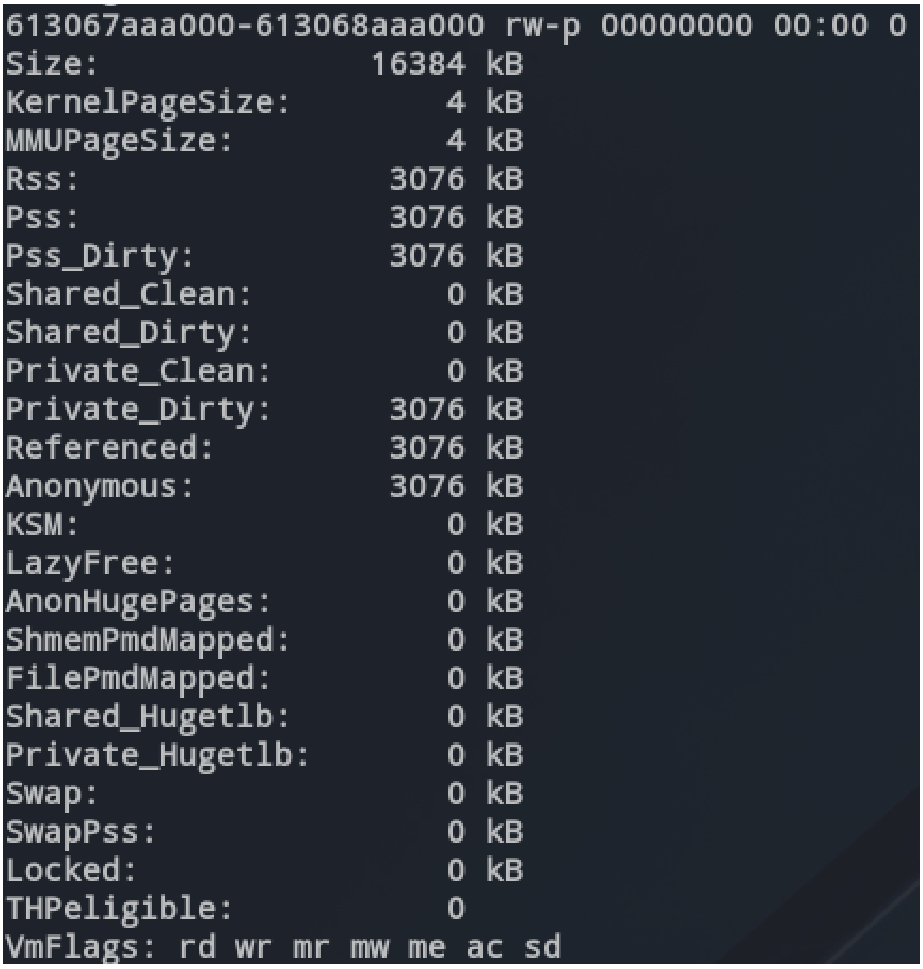
Isto inicializa 3MB da `info` *array* na seção `.bss`, aumentando o RSS (resident



```
613067aaa000-613068aaa000 rw-p 00000000 00:00 0
Size:              16384 kB
KernelPageSize:        4 kB
MMUPageSize:           4 kB
Rss:                   0 kB
Pss:                   0 kB
Pss_Dirty:             0 kB
Shared_Clean:          0 kB
Shared_Dirty:          0 kB
Private_Clean:         0 kB
Private_Dirty:         0 kB
Referenced:            0 kB
Anonymous:             0 kB
KSM:                   0 kB
LazyFree:              0 kB
AnonHugePages:         0 kB
ShmemPmdMapped:        0 kB
FilePmdMapped:         0 kB
Shared_Hugetlb:        0 kB
Private_Hugetlb:       0 kB
Swap:                  0 kB
SwapPss:               0 kB
Locked:                0 kB
THPeligible:           0
VmFlags: rd wr mr mw me ac sd
```

Figure 2: Antes

set size).



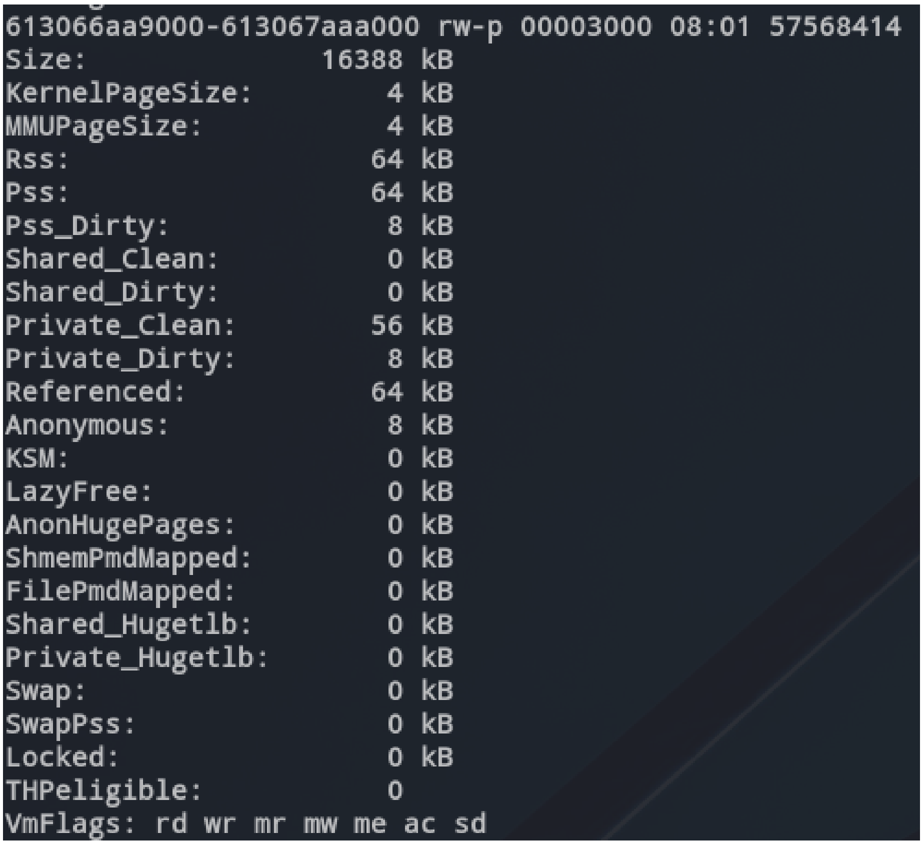
```
613067aaa000-613068aaa000 rw-p 00000000 00:00 0
Size:                  16384 kB
KernelPageSize:            4 kB
MMUPageSize:               4 kB
Rss:                    3076 kB
Pss:                    3076 kB
Pss_Dirty:              3076 kB
Shared_Clean:              0 kB
Shared_Dirty:              0 kB
Private_Clean:             0 kB
Private_Dirty:          3076 kB
Referenced:             3076 kB
Anonymous:              3076 kB
KSM:                       0 kB
LazyFree:                  0 kB
AnonHugePages:             0 kB
ShmemPmdMapped:            0 kB
FilePmdMapped:             0 kB
Shared_Hugetlb:            0 kB
Private_Hugetlb:           0 kB
Swap:                      0 kB
SwapPss:                   0 kB
Locked:                    0 kB
THPeligible:               0
VmFlags: rd wr mr mw me ac sd
```

Figure 3: Depois

2. b. Access 256 bytes of initialized data (.data) with maximum impact in Private Clean pages:

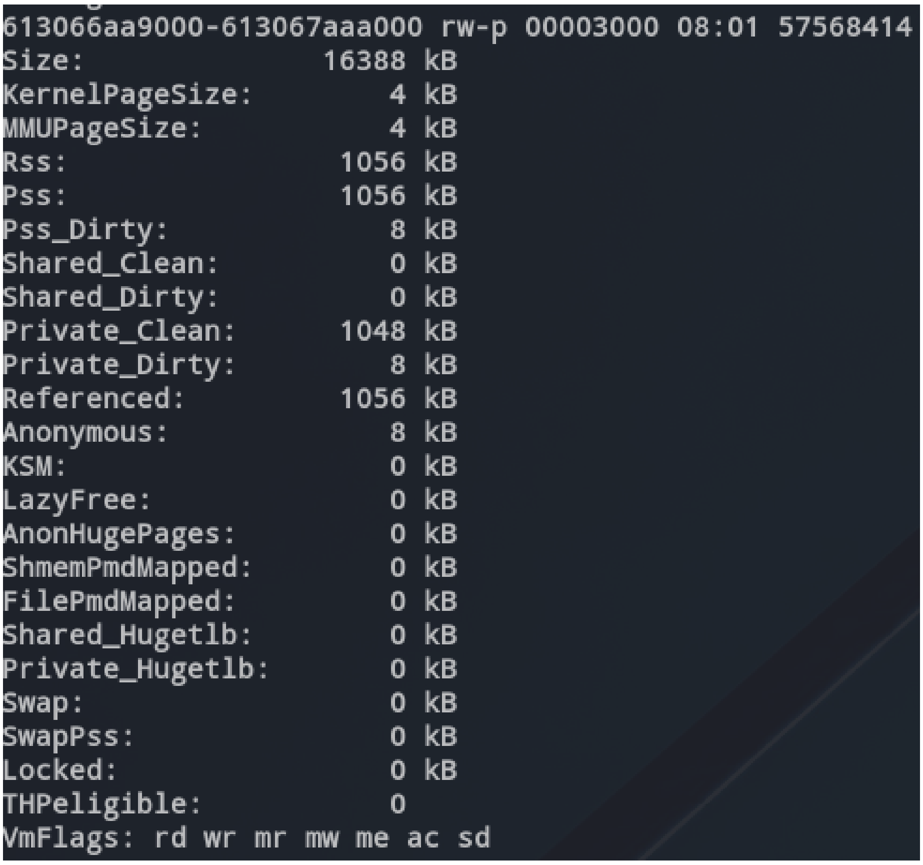
```
int acumulator = 0;
for (int i = 0; i <= 256 * 4096; i += 4096) {
    acumulator += data[i];
};
```

Isto faz aceder a 256 bytes da *array* `data` de 4KB em 4KB, com impacto nas páginas *Private_Clean*.



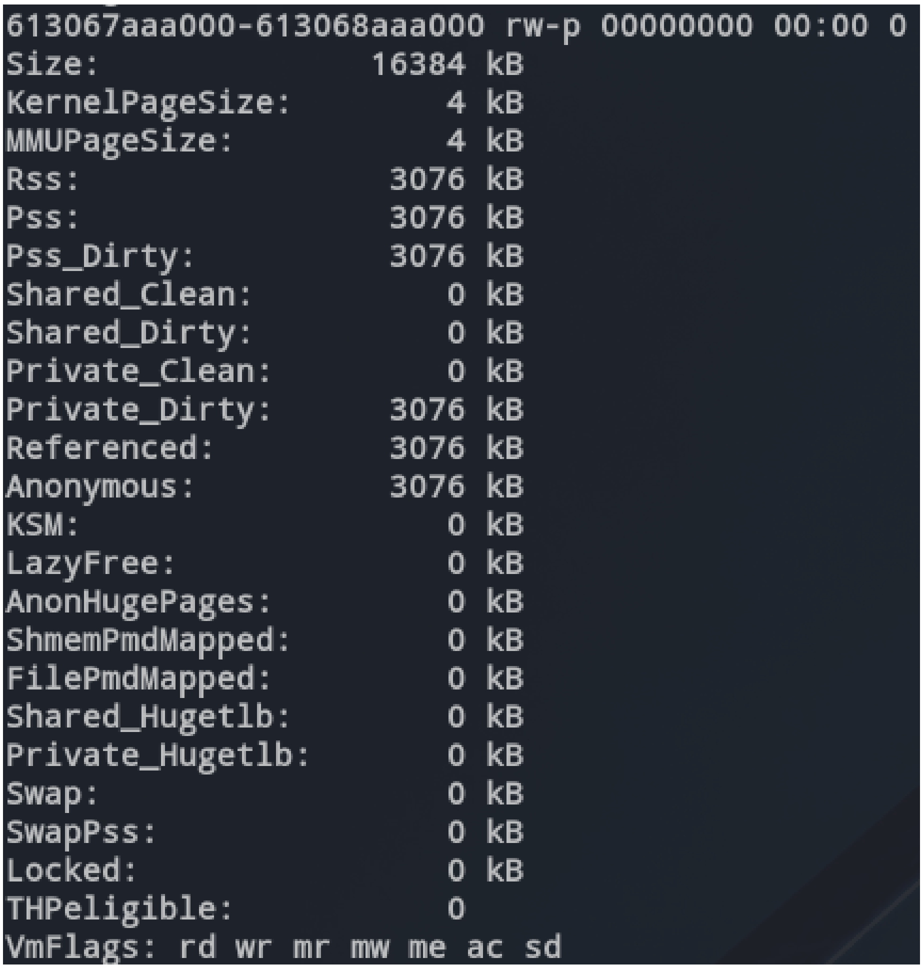
```
613066aa9000-613067aaa000 rw-p 00003000 08:01 57568414
Size:                 16388 kB
KernelPageSize:           4 kB
MMUPageSize:              4 kB
Rss:                     64 kB
Pss:                     64 kB
Pss_Dirty:                8 kB
Shared_Clean:             0 kB
Shared_Dirty:             0 kB
Private_Clean:           56 kB
Private_Dirty:            8 kB
Referenced:              64 kB
Anonymous:                8 kB
KSM:                      0 kB
LazyFree:                 0 kB
AnonHugePages:            0 kB
ShmemPmdMapped:           0 kB
FilePmdMapped:            0 kB
Shared_Hugetlb:           0 kB
Private_Hugetlb:          0 kB
Swap:                     0 kB
SwapPss:                  0 kB
Locked:                   0 kB
THPeligible:              0
VmFlags: rd wr mr mw me ac sd
```

Figure 4: Antes



```
613066aa9000-613067aaa000 rw-p 00003000 08:01 57568414
Size:               16388 kB
KernelPageSize:         4 kB
MMUPageSize:            4 kB
Rss:                 1056 kB
Pss:                 1056 kB
Pss_Dirty:              8 kB
Shared_Clean:           0 kB
Shared_Dirty:           0 kB
Private_Clean:       1048 kB
Private_Dirty:          8 kB
Referenced:          1056 kB
Anonymous:              8 kB
KSM:                    0 kB
LazyFree:               0 kB
AnonHugePages:          0 kB
ShmemPmdMapped:         0 kB
FilePmdMapped:          0 kB
Shared_Hugetlb:         0 kB
Private_Hugetlb:        0 kB
Swap:                   0 kB
SwapPss:                0 kB
Locked:                 0 kB
THPeligible:            0
VmFlags: rd wr mr mw me ac sd
```

Figure 5: Depois



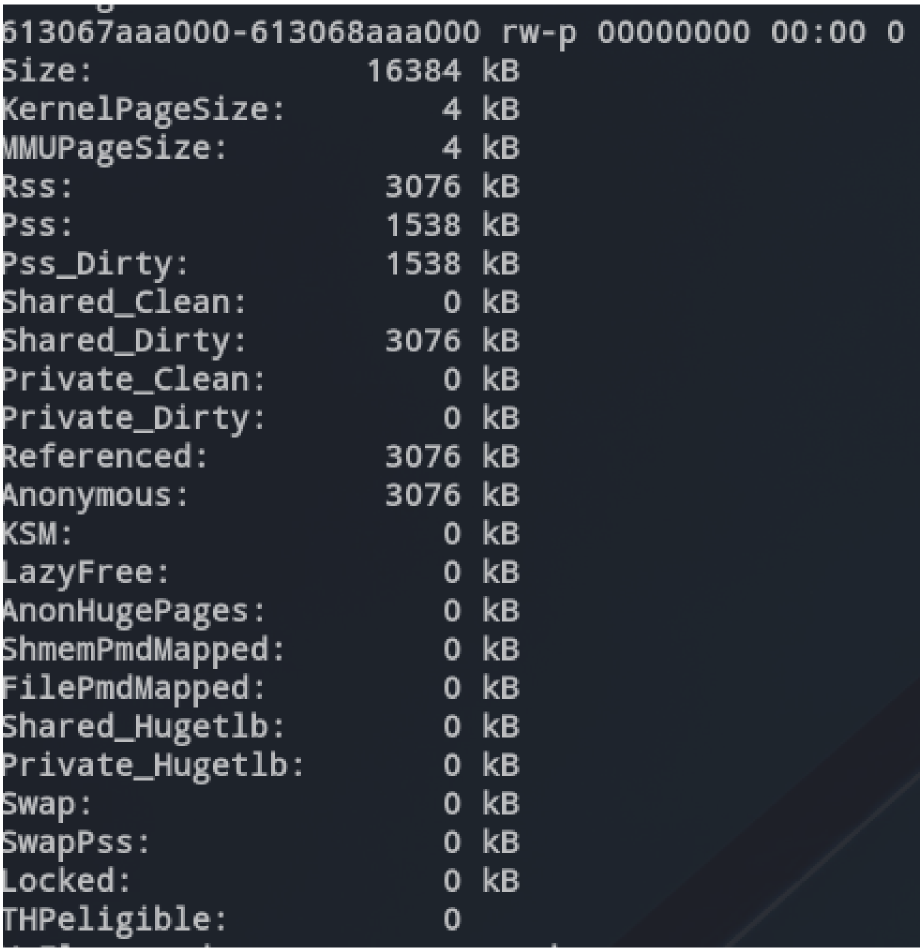
```
613067aaa000-613068aaa000 rw-p 00000000 00:00 0
Size:              16384 kB
KernelPageSize:        4 kB
MMUPageSize:           4 kB
Rss:                3076 kB
Pss:                3076 kB
Pss_Dirty:          3076 kB
Shared_Clean:          0 kB
Shared_Dirty:          0 kB
Private_Clean:         0 kB
Private_Dirty:      3076 kB
Referenced:         3076 kB
Anonymous:          3076 kB
KSM:                   0 kB
LazyFree:              0 kB
AnonHugePages:         0 kB
ShmemPmdMapped:        0 kB
FilePmdMapped:         0 kB
Shared_Hugetlb:        0 kB
Private_Hugetlb:       0 kB
Swap:                  0 kB
SwapPss:               0 kB
Locked:                0 kB
THPeligible:           0
VmFlags: rd wr mr mw me ac sd
```

Figure 6: Antes

3. c. Reduce the Pss of non-initialized data (.bss) to around 1.5MB for 30 seconds, while keeping Rss:

```
pid_t pid = fork();
if (pid == 0) {
    sleep(30);
    return 0;
} else {
    int status;
    waitpid(pid, &status, 0);
}
```

Ao criar um processo *child* dividimos o .bss por 2 processos, reduzindo assim o PSS ao mesmo tempo que se mantém o RSS.



```
613067aaa000-613068aaa000 rw-p 00000000 00:00 0
Size:              16384 kB
KernelPageSize:        4 kB
MMUPageSize:           4 kB
Rss:                3076 kB
Pss:                1538 kB
Pss_Dirty:          1538 kB
Shared_Clean:          0 kB
Shared_Dirty:       3076 kB
Private_Clean:         0 kB
Private_Dirty:         0 kB
Referenced:         3076 kB
Anonymous:          3076 kB
KSM:                   0 kB
LazyFree:              0 kB
AnonHugePages:         0 kB
ShmemPmdMapped:        0 kB
FilePmdMapped:         0 kB
Shared_Hugetlb:        0 kB
Private_Hugetlb:       0 kB
Swap:                  0 kB
SwapPss:               0 kB
Locked:                0 kB
THPeligible:           0
```

Figure 7: Depois

4. d. Execute a single operating system function that results in two new regions being added to the existing address space:

Existem 4 formas de adicionar espaço à memória mapeada, nomeadamente, alocação (quando existem tantos `malloc` que é necessária mais memória), criação de `threads` (cria mais `stacks`), carregamento de bibliotecas dinâmicas a meio da execução através do `dlopen` e mapeamento de ficheiros através do `mmap`.

Como devemos usar apenas uma operação e criar duas novas regiões, criamos uma biblioteca que tivesse `.text` e `.data`

```
#define DATA_SIZE 512 * 1024 // 512KB for .data section

char data[DATA_SIZE] = {1};

void increase_128KB() {
  for (int i = 0; i < 128 * 1024; i++) {
    data[i] += 1;
  }
}
```

Para agilizar o processo de transformação em `.so`, linkagem e compilação do programa, criamos um *script bash*, que depois de `chmod +x compile.sh` podemos correr com `$ ./compile.sh`

```
#!/bin/bash

# Compile the shared library
gcc -shared -o mylib.so mylib.c -fPIC

# Compile the main program
gcc -o prog prog.c -ldl
```

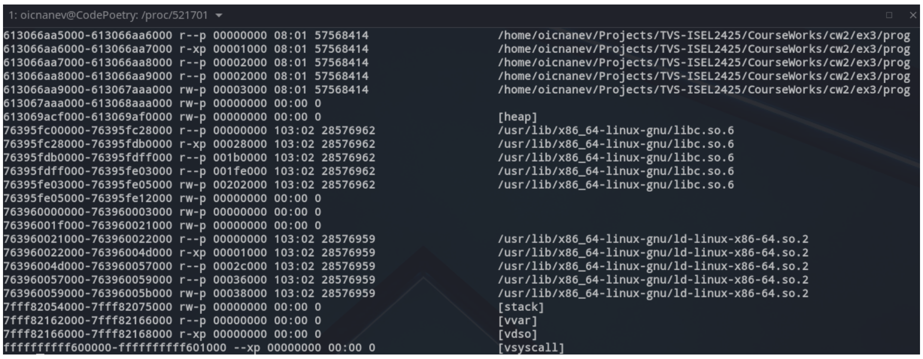
Por fim no `prog.c` importamos dinâmicamente esta biblioteca com `dlopen`

```
void *handle = dlopen("./mylib.so", RTLD_LAZY);
if (!handle) {
    printf("Erro ao carregar biblioteca: %s\n", dlerror());
    return 1;
}
```

São assim mapeadas regiões novas (mais de 2):

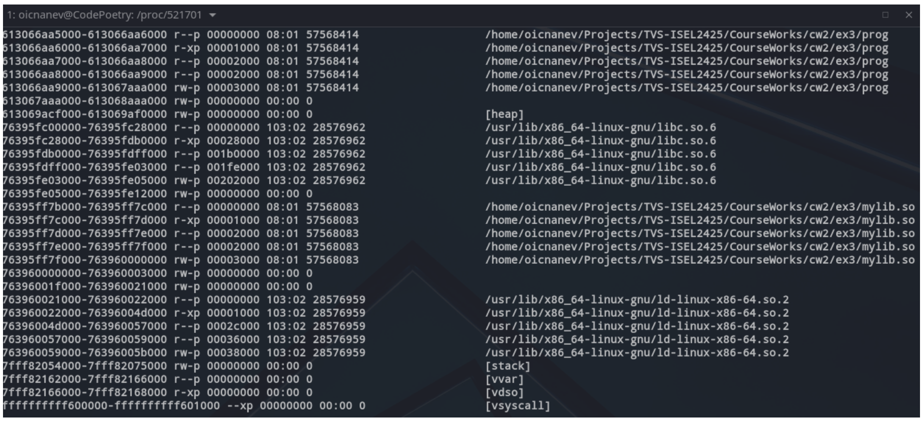
5. e. Increase Private dirty pages by about 128KB in the region for the data section created in d):

```
void (*increase_128KB)(void) = dlsym(handle, "increase_128KB");
const char *dlsym_error = dlerror();
if (dlsym_error) {
    printf("Error finding the function: %s\n", dlsym_error);
    dlclose(handle);
    return 1;
```



```
613066aa5000-613066aa6000 r--p 00000000 08:01 57568414                   /home/oicnanev/Projects/TVS-ISEL2425/CourseWorks/cw2/ex3/prog
613066aa6000-613066aa7000 r-xp 00001000 08:01 57568414                   /home/oicnanev/Projects/TVS-ISEL2425/CourseWorks/cw2/ex3/prog
613066aa7000-613066aa8000 r--p 00002000 08:01 57568414                   /home/oicnanev/Projects/TVS-ISEL2425/CourseWorks/cw2/ex3/prog
613066aa8000-613066aa9000 r--p 00002000 08:01 57568414                   /home/oicnanev/Projects/TVS-ISEL2425/CourseWorks/cw2/ex3/prog
613066aa9000-613067aaa000 rw-p 00003000 08:01 57568414                   /home/oicnanev/Projects/TVS-ISEL2425/CourseWorks/cw2/ex3/prog
613067aaa000-613068aaa000 rw-p 00000000 00:00 0
613069acf000-613069af0000 rw-p 00000000 00:00 0                          [heap]
76395fc00000-76395fc28000 r--p 00000000 103:02 28576962                  /usr/lib/x86_64-linux-gnu/libc.so.6
76395fc28000-76395fdb0000 r-xp 00028000 103:02 28576962                  /usr/lib/x86_64-linux-gnu/libc.so.6
76395fdb0000-76395fdff000 r--p 001b0000 103:02 28576962                  /usr/lib/x86_64-linux-gnu/libc.so.6
76395fdff000-76395fe03000 r--p 001fe000 103:02 28576962                  /usr/lib/x86_64-linux-gnu/libc.so.6
76395fe03000-76395fe05000 rw-p 00202000 103:02 28576962                  /usr/lib/x86_64-linux-gnu/libc.so.6
76395fe05000-76395fe12000 rw-p 00000000 00:00 0
763960000000-763960003000 rw-p 00000000 00:00 0
76396001f000-763960021000 rw-p 00000000 00:00 0
763960021000-763960022000 r--p 00000000 103:02 28576959                  /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
763960022000-76396004d000 r-xp 00001000 103:02 28576959                  /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
76396004d000-763960057000 r--p 0002c000 103:02 28576959                  /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
763960057000-763960059000 r--p 00036000 103:02 28576959                  /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
763960059000-76396005b000 rw-p 00038000 103:02 28576959                  /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
7fff82054000-7fff82075000 rw-p 00000000 00:00 0                          [stack]
7fff82162000-7fff82166000 r--p 00000000 00:00 0                          [vvar]
7fff82166000-7fff82168000 r-xp 00000000 00:00 0                          [vdso]
ffffffffff600000-ffffffffff601000 --xp 00000000 00:00 0                  [vsyscall]
```

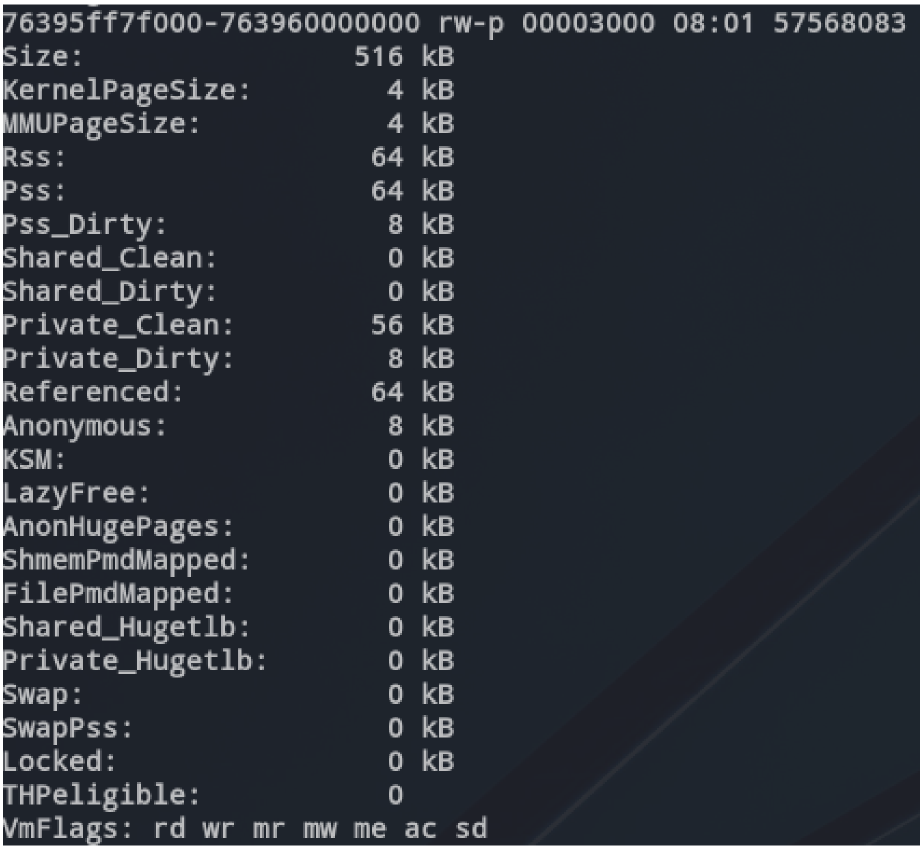
Figure 8: maps antes do dlopen



```
613066aa5000-613066aa6000 r--p 00000000 08:01 57568414                   /home/oicnanev/Projects/TVS-ISEL2425/CourseWorks/cw2/ex3/prog
613066aa6000-613066aa7000 r-xp 00001000 08:01 57568414                   /home/oicnanev/Projects/TVS-ISEL2425/CourseWorks/cw2/ex3/prog
613066aa7000-613066aa8000 r--p 00002000 08:01 57568414                   /home/oicnanev/Projects/TVS-ISEL2425/CourseWorks/cw2/ex3/prog
613066aa8000-613066aa9000 r--p 00002000 08:01 57568414                   /home/oicnanev/Projects/TVS-ISEL2425/CourseWorks/cw2/ex3/prog
613066aa9000-613067aaa000 rw-p 00003000 08:01 57568414                   /home/oicnanev/Projects/TVS-ISEL2425/CourseWorks/cw2/ex3/prog
613067aaa000-613068aaa000 rw-p 00000000 00:00 0
613069acf000-613069af0000 rw-p 00000000 00:00 0                          [heap]
76395fc00000-76395fc28000 r--p 00000000 103:02 28576962                  /usr/lib/x86_64-linux-gnu/libc.so.6
76395fc28000-76395fdb0000 r-xp 00028000 103:02 28576962                  /usr/lib/x86_64-linux-gnu/libc.so.6
76395fdb0000-76395fdff000 r--p 001b0000 103:02 28576962                  /usr/lib/x86_64-linux-gnu/libc.so.6
76395fdff000-76395fe03000 r--p 001fe000 103:02 28576962                  /usr/lib/x86_64-linux-gnu/libc.so.6
76395fe03000-76395fe05000 rw-p 00202000 103:02 28576962                  /usr/lib/x86_64-linux-gnu/libc.so.6
76395fe05000-76395fe12000 rw-p 00000000 00:00 0
76395ff7b000-76395ff7c000 r--p 00000000 08:01 57568083                   /home/oicnanev/Projects/TVS-ISEL2425/CourseWorks/cw2/ex3/mylib.so
76395ff7c000-76395ff7d000 r-xp 00001000 08:01 57568083                   /home/oicnanev/Projects/TVS-ISEL2425/CourseWorks/cw2/ex3/mylib.so
76395ff7d000-76395ff7e000 r--p 00002000 08:01 57568083                   /home/oicnanev/Projects/TVS-ISEL2425/CourseWorks/cw2/ex3/mylib.so
76395ff7e000-76395ff7f000 r--p 00002000 08:01 57568083                   /home/oicnanev/Projects/TVS-ISEL2425/CourseWorks/cw2/ex3/mylib.so
76395ff7f000-763960000000 rw-p 00003000 08:01 57568083                   /home/oicnanev/Projects/TVS-ISEL2425/CourseWorks/cw2/ex3/mylib.so
763960000000-763960003000 rw-p 00000000 00:00 0
76396001f000-763960021000 rw-p 00000000 00:00 0
763960021000-763960022000 r--p 00000000 103:02 28576959                  /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
763960022000-76396004d000 r-xp 00001000 103:02 28576959                  /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
76396004d000-763960057000 r--p 0002c000 103:02 28576959                  /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
763960057000-763960059000 r--p 00036000 103:02 28576959                  /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
763960059000-76396005b000 rw-p 00038000 103:02 28576959                  /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
7fff82054000-7fff82075000 rw-p 00000000 00:00 0                          [stack]
7fff82162000-7fff82166000 r--p 00000000 00:00 0                          [vvar]
7fff82166000-7fff82168000 r-xp 00000000 00:00 0                          [vdso]
ffffffffff600000-ffffffffff601000 --xp 00000000 00:00 0                  [vsyscall]
```

Figure 9: maps depois do dlopen

```
}
increase_128KB();
```



```
76395ff7f000-763960000000 rw-p 00003000 08:01 57568083
Size:                  516 kB
KernelPageSize:          4 kB
MMUPageSize:             4 kB
Rss:                    64 kB
Pss:                    64 kB
Pss_Dirty:               8 kB
Shared_Clean:            0 kB
Shared_Dirty:            0 kB
Private_Clean:          56 kB
Private_Dirty:           8 kB
Referenced:             64 kB
Anonymous:               8 kB
KSM:                     0 kB
LazyFree:                0 kB
AnonHugePages:           0 kB
ShmemPmdMapped:          0 kB
FilePmdMapped:           0 kB
Shared_Hugetlb:          0 kB
Private_Hugetlb:         0 kB
Swap:                    0 kB
SwapPss:                 0 kB
Locked:                  0 kB
THPeligible:             0
VmFlags: rd wr mr mw me ac sd
```

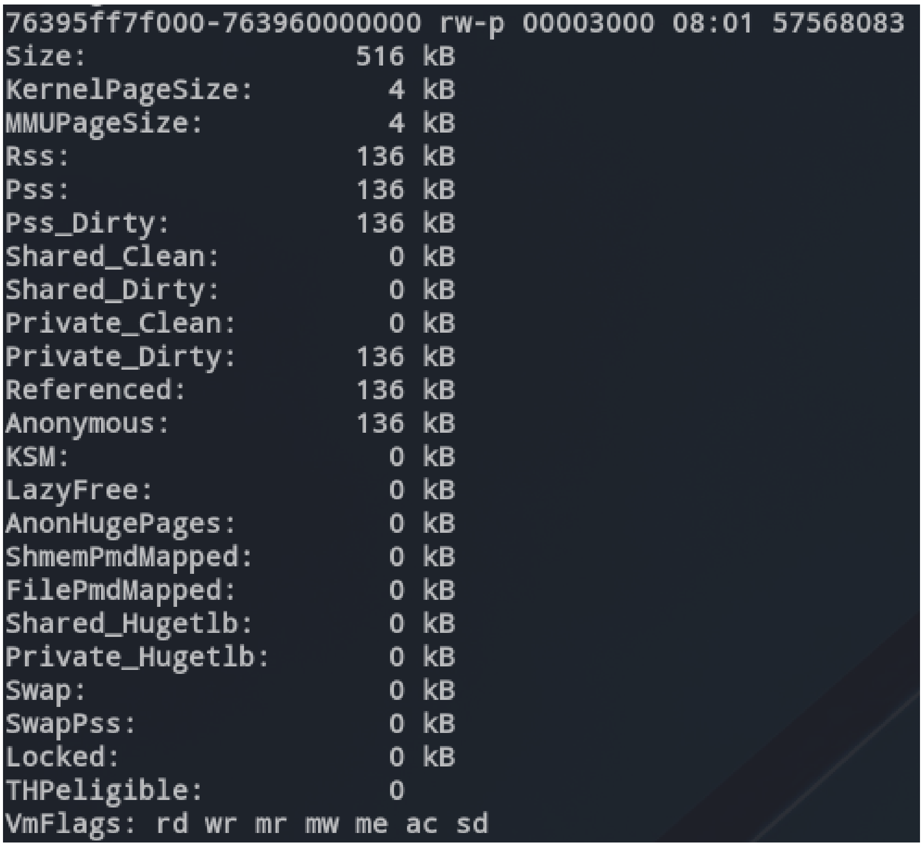
Figure 10: Antes

Para inicializar 128KB da nova região de dados mapeada, aumentando as páginas *Private_dirty*, basta chamar a função `increase_128KB` da biblioteca importada dinamicamente.

Fim

Por fim fechamos a biblioteca carregada com:

```
dlclose(handle);
```



```
76395ff7f000-763960000000 rw-p 00003000 08:01 57568083
Size:                516 kB
KernelPageSize:        4 kB
MMUPageSize:           4 kB
Rss:                 136 kB
Pss:                 136 kB
Pss_Dirty:           136 kB
Shared_Clean:          0 kB
Shared_Dirty:          0 kB
Private_Clean:         0 kB
Private_Dirty:       136 kB
Referenced:          136 kB
Anonymous:           136 kB
KSM:                   0 kB
LazyFree:              0 kB
AnonHugePages:         0 kB
ShmemPmdMapped:        0 kB
FilePmdMapped:         0 kB
Shared_Hugetlb:        0 kB
Private_Hugetlb:       0 kB
Swap:                  0 kB
SwapPss:               0 kB
Locked:                0 kB
THPeligible:           0
VmFlags: rd wr mr mw me ac sd
```

Figure 11: Depois