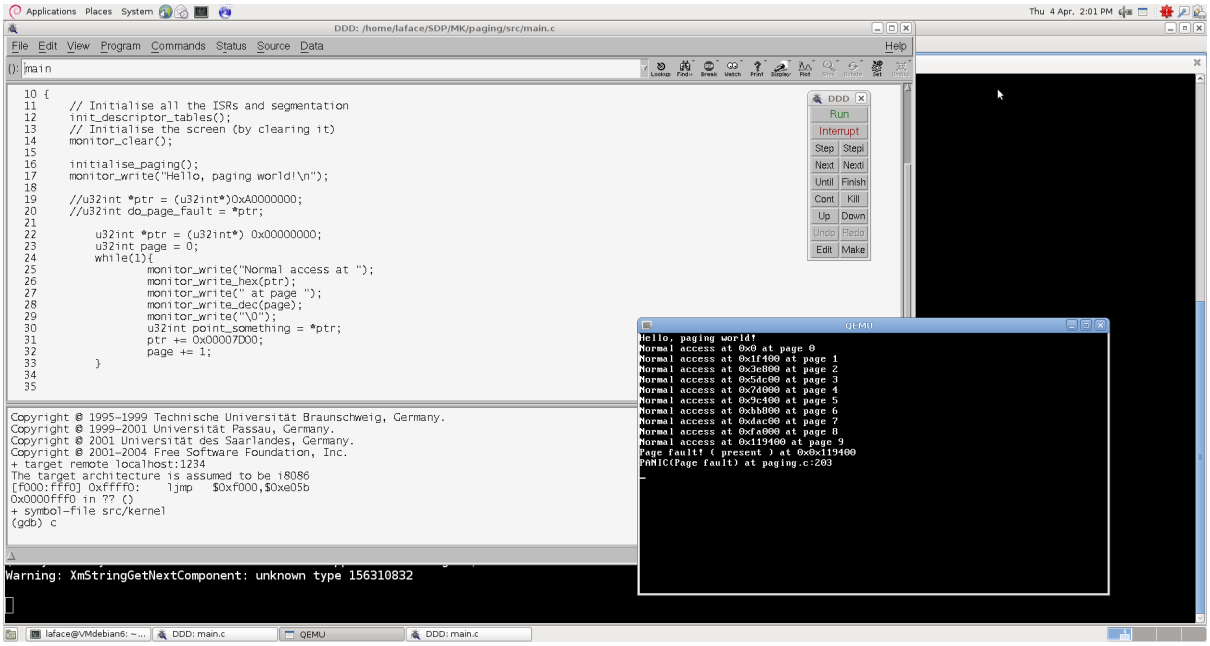
Report 4
System and device programming
Prof. Pietro Laface

Nicola Sabino
mat:253839
Date: 04/04/2019

Laboratory 4.2

Change the file main.c to verify which is the first address that produces a page fault.
Hint: Increment ptr, starting from 0, by the dimension of a page (take care of the pointer arithmetic).

```
#include "monitor.h"
#include "descriptor_tables.h"
#include "timer.h"
#include "paging.h"

int main(struct multiboot *mboot_ptr)
{
    // Initialise all the ISRs and segmentation
    init_descriptor_tables();
    // Initialise the screen (by clearing it)
    monitor_clear();

    initialise_paging();
    monitor_write("Hello, paging world!\n");

    //u32int *ptr = (u32int*)0xA0000000;
    //u32int do_page_fault = *ptr;

        u32int *ptr = (u32int*) 0x00000000;
        u32int page = 0;
        while(1){
                monitor_write("Normal access at ");
                monitor_write_hex(ptr);
                monitor_write(" at page ");
                monitor_write_dec(page);
                monitor_write("\0");
                u32int point_something = *ptr;
                ptr += 0x00007D00;
                page += 1;
        }


    return 0;
}
```


Applications Places System
Thu 4 Apr, 2:01 PM
DDD: /home/laface/SDP/MK/paging/src/main.c
File Edit View Program Commands Status Source Data Help
(): main
10 {
11 // Initialise all the ISRs and segmentation
12 init_descriptor_tables();
13 // Initialise the screen (by clearing it)
14 monitor_clear();
15
16 initialise_paging();
17 monitor_write("Hello, paging world!\n");
18
19 //u32int *ptr = (u32int*)0xA0000000;
20 //u32int do_page_fault = *ptr;
21
22 u32int *ptr = (u32int*) 0x00000000;
23 u32int page = 0;
24 while(1){
25 monitor_write("Normal access at ");
26 monitor_write_hex(ptr);
27 monitor_write(" at page ");
28 monitor_write_dec(page);
29 monitor_write("\0");
30 u32int point_something = *ptr;
31 ptr += 0x00007D00;
32 page += 1;
33 }
34
35
DDD
Run
Interrupt
Step Stepi
Next Nexti
Until Finish
Cont Kill
Up Down
Undo Redo
Edit Make
Copyright © 1995-1999 Technische Universität Braunschweig, Germany.
Copyright © 1999-2001 Universität Passau, Germany.
Copyright © 2001 Universität des Saarlandes, Germany.
Copyright © 2001-2004 Free Software Foundation, Inc.
+ target remote localhost:1234
The target architecture is assumed to be i8086
[f000:fff0] 0xffff0: ljmp $0xf000,$0xe05b
0x0000fff0 in ?? ()
+ symbol-file src/kernel
(gdb) c
Warning: XmStringGetNextComponent: unknown type 156310832
QEMU
Hello, paging world!
Normal access at 0x0 at page 0
Normal access at 0x1f400 at page 1
Normal access at 0x3e800 at page 2
Normal access at 0x5dc00 at page 3
Normal access at 0x7d000 at page 4
Normal access at 0x9c400 at page 5
Normal access at 0xbb800 at page 6
Normal access at 0xdac00 at page 7
Normal access at 0xfa000 at page 8
Normal access at 0x119400 at page 9
Page fault! (present) at 0x0x119400
PANIC(Page fault) at paging.c:203
laface@VMdebian6: ~...
DDD: main.c
QEMU
DDD: main.c