МГТУ им. Баумана

Лабораторная работа №2

По курсу: "Операционные системы"

# Процессы. Защищенный режим.

Работу выполнил: Мокеев Даниил, ИУ7-56

Преподаватель: Рязанова Н.Ю.

*Москва, 2020*

# 1 Листинг кода алгоритмов

В данном разделе будут приведены листинги кода реализованных программ.

Листинг 1.1: Процессы-сироты

```
;
.386p

descr struc
limit  dw 0
base_l dw 0
base_m db 0
attr_1 db 0
arrt_2 db 0
base_h db 0
descr ends


intr struc
offs_l dw 0
sel    dw 0
rsrv   db 0
attr   db 0
offs_h dw 0
intr ends


pm_seg segment para public 'code' use32
assume cs:pm_seg

gdt             label byte
gdt_null        descr <>
gdt_data        descr <0FFFFh,0,0,92h,0CFh,0>
gdt_code16      descr <rm_seg_size-1,0,0,98h,0,0>
gdt_code32      descr <pm_seg_size-1,0,0,98h,0CFh,0>
gdt_data32      descr <pm_seg_size-1,0,0,92h,0CFh,0>
gdt_stack32     descr <stack_size-1,0,0,92h,0CFh,0>
gdt_size=$-gdt

gdtr            dw gdt_size-1
dd ?

```

```
38 ;
39 sel_data          equ 8
40 sel_code16       equ 16
41 sel_code32       equ 24
42 sel_data32       equ 32
43 sel_stack32      equ 40
44
45 idt              label byte
46 trap1            intr 13 dup (<0,sel_code32,0,8Fh,0>)
47 trap13           intr <0,sel_code32,0,8Fh,0>
48 trap2            intr 18 dup (<0,sel_code32,0,8Fh,0>)
49 int_time         intr <0,sel_code32,0,8Eh,0>
50 int_keyboard     intr <0,sel_code32,0,8Eh,0>
51 idt_size=$-idt
52
53 idtr             dw idt_size-1
54 dd ?
55
56 rm_idtr          dw 3FFh,0,0
57
58 hex              db 'h'
59 hex_len=$-hex
60 mb               db 'MB'
61 mb_len=$-mb
62
63 hello_msg        db 'DOS is in real mode now.$'
64 pm_msg           db 'DOS switched to protected mode.'
65 pm_msg_len=$-pm_msg
66 tt_msg           db 'Timer ticks:      '
67 tt_msg_len=$-tt_msg
68 am_msg           db 'Available memory: '
69 am_msg_len=$-am_msg
70 esc_from_pr      db 'Press ESC to switch to real mode...'
71 esc_from_pr_len=$-esc_from_pr
72 ret_to_rm_msg   db 'DOS switched to real mode.$'
73
74 scan2ascii       db 0,1Bh,'1','2','3','4','5','6','7','8','9','0','-','=
       ',8
75 db ' ','q','w','e','r','t','y','u','i','o','p','[',']','$'
76 db ' ','a','s','d','f','g','h','j','k','l',';','"',0
77 db '\','z','x','c','v','b','n','m',',','.','/',0,0,0,' ',0,0
78 db 0,0,0,0,0,0,0,0,0,0,0,0
79
```

```
80  attr1            db 3Fh
81  attr2            db 4Fh
82  screen_addr      dd 640
83  timer            dd 0
84
85  master           db 0
86  slave            db 0
87
88  ;
89  print_str macro msg,len,offset
90  local print
91  push   ebp
92  mov    ecx,len
93  mov    ebp,0B8000h
94  add    ebp,offset
95  xor    esi,esi
96  mov    ah,attr2
97  print:
98  mov    al,byte ptr msg[esi]
99  mov    es:[ebp],ax
100 add    ebp,2
101 inc    esi
102 loop   print
103 pop    ebp
104 endm
105
106 ;                                                    EOI

107 send_eoi macro
108 mov    al,20h
109 out    20h,al
110 endm
111
112 pm_start:
113 ;

114 mov    ax,sel_data
115 mov    ds,ax
116 mov    es,ax
117 mov    ax,sel_stack32
118 mov    ebx,stack_size
119 mov    ss,ax
120 mov    esp,ebx
```

```
121
122 ;
123 sti
124
125 ;

126 print_str pm_msg,pm_msg_len,360
127 print_str tt_msg,tt_msg_len,520
128 print_str am_msg,am_msg_len,5*160+40
129 print_str esc_from_pr,esc_from_pr_len,6*160+40
130
131 call available_memory
132 jmp     $
133
134 ;
135 exc13 proc
136 pop     eax
137 iret
138 exc13 endp
139
140 ;
141 dummy_exc proc
142 iret
143 dummy_exc endp
144
145 ;

146 int_time_handler:
147 push    eax
148 push    ebp
149 push    ecx
150 push    dx
151
152 ;
153 mov     eax,timer
154
155 ;
156 mov     ebp,0B8000h
157 mov     ecx,8
158 add     ebp,530+2*(tt_msg_len)
159 mov     dh,attr2
160 main_loop:
161 mov     dl,al
```

```
162 and     dl,0Fh
163 cmp     dl,10
164 jl      less_than_10
165 sub     dl,10
166 add     dl,'A'
167 jmp     print
168 less_than_10:
169 add     dl,'0'
170 print:
171 mov     es:[ebp],dx
172 ror     eax,4
173 sub     ebp,2
174 loop    main_loop
175
176 ;
177 inc     eax
178 mov     timer,eax
179
180 send_eoi
181 pop     dx
182 pop     ecx
183 pop     ebp
184 pop     eax
185
186 iretd
187
188 ;
189 int_keyboard_handler:
190 push    eax
191 push    ebx
192 push    es
193 push    ds
194
195 ;                                                              –

196 in      al,60h
197
198 ;                                   ESC
199 cmp     al,01h
200 je      esc_pressed
201
202 ;
203 cmp     al,39h
```

```
204 ja      skip_translate
205
206 ;                                       -           ASCII
207 mov     bx,sel_data32
208 mov     ds,bx
209 mov     ebx,offset scan2ascii
210 xlatb
211 mov     bx,sel_data
212 mov     es,bx
213 mov     ebx,screen_addr
214
215 ;                                   Backspace
216 cmp     al,8
217 je      bs_pressed
218
219 ;
220 mov     es:[ebx+0B8000h],al
221 add     dword ptr screen_addr,2
222 jmp     skip_translate
223
224 bs_pressed:
225 ;
226 mov     al,' '
227 sub     ebx,2
228 mov     es:[ebx+0B8000h],al
229 mov     screen_addr,ebx
230
231 skip_translate:
232 ;
233 in      al,61h
234 or      al,80h
235 out     61h,al
236
237 send_eoi
238 pop     ds
239 pop     es
240 pop     ebx
241 pop     eax
242
243 iretd
244
245 esc_pressed:
246 ;
```

```
247 in      al,61h
248 or      al,80h
249 out     61h,al
250
251 send_eoi
252 pop     ds
253 pop     es
254 pop     ebx
255 pop     eax
256
257 ;
258 cli
259
260 ;
261 db     0EAh
262 dd     offset rm_return
263 dw     sel_code16
264
265 ;

266 available_memory proc
267 push    ds
268
269 mov     ax,sel_data
270 mov     ds,ax
271
272 ;
273 mov     ebx,100001h
274 ;
275 mov     dl,0FFh
276 ;

277 mov     ecx,0FFEFFFFFh
278
279 check:
280 ;
281 mov     dh,ds:[ebx]
282 mov     ds:[ebx],dl
283 cmp     ds:[ebx],dl
284 jnz     end_of_memory
285 mov     ds:[ebx],dh
286 inc     ebx
287 loop    check
```

```
288
289 end_of_memory:
290 pop     ds
291 xor     edx,edx
292 mov     eax,ebx
293
294 ;
295 mov     ebx,100000h
296 div     ebx
297
298 push    ecx
299 push    dx
300 push    ebp
301
302 ;
303 mov     ebp,0B8000h
304 mov     ecx,8
305 add     ebp,5*160+2*(am_msg_len+7)+40
306 mov     dh,attr2
307 cycle:
308 mov     dl,al
309 and     dl,0Fh
310 cmp     dl,10
311 jl      number
312 sub     dl,10
313 add     dl,'A'
314 jmp     print_m
315 number:
316 add     dl,'0'
317 print_m:
318 mov     es:[ebp],dx
319 ror     eax,4
320
321 sub     ebp,2
322 loop    cycle
323 sub     ebp,0B8000h
324
325 pop     ebp
326 pop     dx
327 pop     ecx
328 ret
329 available_memory  endp
330
```

```
331 pm_seg_size=$-gdt
332 pm_seg ends
333
334
335 rm_seg segment para public 'code' use16
336 assume cs:rm_seg,ds:pm_seg,ss:s_seg
337
338 ;
339 cls macro
340 mov     ax,3
341 int     10h
342 endm
343
344 ;
345 print_str macro msg
346 mov     ah,9
347 mov     edx,offset msg
348 int     21h
349 endm
350
351 rm_start:
352 mov     ax,pm_seg
353 mov     ds,ax
354
355 cls
356
357 mov     AX, 0B800h
358 mov     ES, AX
359 mov     DI, 200
360 mov     cx, 24
361 mov     ebx, offset hello_msg
362 mov     ah, attr1
363 mov     al, byte ptr [ebx]
364 screen0:
365 stosw
366 inc     bx
367 mov     al, byte ptr [ebx]
368 loop    screen0
369
370
371 ;

372 xor     eax,eax
```

```
373 mov     ax,rm_seg
374 shl     eax,4
375 mov     word ptr gdt_code16+2,ax
376 shr     eax,16
377 mov     byte ptr gdt_code16+4,al
378 mov     ax,pm_seg
379 shl     eax,4
380 push    eax
381 push    eax
382 mov     word ptr gdt_code32+2,ax
383 mov     word ptr gdt_stack32+2,ax
384 mov     word ptr gdt_data32+2,ax
385 shr     eax,16
386 mov     byte ptr gdt_code32+4,al
387 mov     byte ptr gdt_stack32+4,al
388 mov     byte ptr gdt_data32+4,al
389
390 ;                                                   GDT
391 pop     eax
392 add     eax,offset GDT
393 mov     dword ptr gdtr+2,eax
394 mov     word ptr gdtr,gdt_size-1
395
396 ;                                          GDTR
397 lgdt    fword ptr gdtr
398
399 ;                                                   IDT
400 pop     eax
401 add     eax,offset idt
402 mov     dword ptr idtr+2,eax
403 mov     word ptr idtr,idt_size-1
404
405 ;

406 mov     eax,offset dummy_exc
407 mov     trap1.offs_l,ax
408 shr     eax,16
409 mov     trap1.offs_h,ax
410 mov     eax,offset exc13
411 mov     trap13.offs_l,ax
412 shr     eax,16
413 mov     trap13.offs_h,ax
414 mov     eax,offset dummy_exc
```

```
415 mov     trap2.offs_l,ax
416 shr     eax,16
417 mov     trap2.offs_h,ax
418 mov     eax,offset int_time_handler
419 mov     int_time.offs_l,ax
420 shr     eax,16
421 mov     int_time.offs_h,ax
422 mov     eax,offset int_keyboard_handler
423 mov     int_keyboard.offs_l,ax
424 shr     eax,16
425 mov     int_keyboard.offs_h,ax
426
427 ;  C

428 in      al,21h
429 mov     master,al
430 in      al,0A1h
431 mov     slave,al
432
433 ;

434 mov     dx,20h
435 mov     al,11h
436 out     dx,al
437 inc     dx
438 mov     al,20h
439 out     dx,al
440 mov     al,4
441 out     dx,al
442 mov     al,1
443 out     dx,  al
444
445 ;
                                ,              IRQ0     IRQ1
446 mov     al,11111100b
447 out     dx,al
448
449 ;

450 mov     dx,0A1h
451 mov     al,0FFh
452 out     dx,al
453
```

```
454 ;                                                    IDTR
455 lidt    fword ptr idtr
456
457 ;                                          20
458 mov     al,0D1h
459 out     64h,al
460 mov     al,0DFh
461 out     60h,al
462
463 ;

464 cli
465 in      al,70h
466 or      al,80h
467 out     70h,al
468
469 ;

                                CR0
470 mov     eax,cr0
471 or      al,1
472 mov     cr0,eax
473
474 ;

475 db      66h
476 db      0EAh
477 dd      offset pm_start
478 dw      sel_code32
479
480 rm_return:
481 ;                                                                CR0
482 mov     eax,cr0
483 and     al,0FEh
484 mov     cr0,eax
485
486 ;                                                                CS
487 db      0EAh
488 dw      $+4
489 dw      rm_seg
490
```

```
491 ;

492 mov     ax,pm_seg
493 mov     ds,ax
494 mov     es,ax
495 mov     ax,s_seg
496 mov     ss,ax
497 mov     ax,stack_size
498 mov     sp,ax
499
500 ;

501 mov     al,11h
502 out     20h,al
503 mov     al,8
504 out     21h,al
505 mov     al,4
506 out     21h,al
507 mov     al,1
508 out     21h,al
509
510 ;

511 mov     al,master
512 out     21h,al
513 mov     al,slave
514 out     0A1h,al
515
516 ;

517 lidt    fword ptr rm_idtr
518
519 ;                                   20
520 mov     al,0D1h
521 out     64h,al
522 mov     al,0DDh
523 out     60h,al
524
525 ;

526 in      al,70h
527 and     al,07FH
528 out     70h,al
```

```
529 sti
530
531 ;cls
532 mov     AX, 0B800h
533 mov     ES, AX
534 mov     DI, 7*160+40
535 mov     cx, 26
536 mov     ebx, offset ret_to_rm_msg
537 mov     ah, attr1
538 mov     al, byte ptr [ebx]
539 screen01:
540 stosw
541 inc     bx
542 mov     al, byte ptr [ebx]
543 loop    screen01
544
545 mov     ah,4Ch
546 int     21h
547 rm_seg_size = $-rm_start
548 rm_seg ends
549
550 s_seg segment para stack 'stack'
551 stack_start db 100h dup(?)
552 stack_size=$-stack_start
553 s_seg ends
554 end     rm_start
```