

# OS Assignment 4 report

name: 刘乐奇

SID: 12011327

Ubuntu用户名: lynchrocket

## 1. CPU Scheduling

Time	HRRN	FIFO/FCFS	RR	SJF	Priority
1	A	A	A	A	A
2	A	A	A	A	B
3	A	A	B	A	A
4	A	A	A	A	D
5	B	B	D	B	D
6	D	D	A	D	C
7	D	D	C	D	C
8	C	C	D	C	C
9	C	C	C	C	A
10	C	C	C	C	A
Avg. Turn-around Time	4.5	4.5	4.75	4.5	4.25

## 2. Preemptive process scheduling

Add comment here to unable the clock interrupt. (/kern/init/init.c line 37)

C clock.h

C clock.c 1

C init.c

X

```
kern > init > C init.c > kern_init(void)
23     const char *message = "OS is loading ...";
24     cprintf("%s\n\n", message);
25
26     pmm_init();           // init physical memory management
27
28     idt_init();           // init interrupt descriptor table
29
30     vmm_init();           // init virtual memory management
31     sched_init();
32     proc_init();          // init process table
33
34     ide_init();           // init ide devices
35     swap_init();          // init swap
36
37     // |clock_init();        // init clock interrupt
38     intr_enable();         // enable irq interrupt
39
40     cpu_idle();            // run idle process
41 }
```



In /user/ex3.c line 41 we can find a function named `set_good()`. This is the function we need to write.

Activities Visual Studio Code 4月 10 23:08

File Edit Selection View Go Run Terminal Help

EXPLORER EX3

user > C ex3.c main(void)

```
30     memset(pids, 0, sizeof(pids));
31     int goods[TOTAL] = {3,1,4,5,2};
32     for (i = 0; i < TOTAL; i++) {
33         acc[i] = 0;
34         if ((pids[i] = fork()) == 0) {
35             acc[i] = 0;
36             while (1) {
37                 spin_delay();
38                 ++ acc[i];
39
40                 if (acc[i] == 2000000)
41                     set_good(goods[i]);
42
43                 if (acc[i] > 4000000) {
44                     cprintf("child pid %d, acc %d\n", getpid(), acc[i]);
45                     exit(acc[i]);
46
47                 }
48             }
49             if (pids[i] < 0) {
50                 goto failed;
51             }
52         }
53     }
54     cprintf("main: fork ok, now need to wait pids.\n");
55
56     for (i = 0; i < TOTAL; i++) {
57         status[i] = 0;
58         waitpid(pids[i], &status[i]);

```

Ln 41, Col 45 Spaces: 5 UTF-8 LF C Linux

## In user mode

Declare `set_good()` in `/user/libs/ulib.h` and define in `/user/libs/ulib.c`

Activities Visual Studio Code 4月 11 00:17

File Edit Selection View Go Run Terminal Help

EXPLORER user > libs > C ulib.h > set\_good(int)

C ulib.h  
C sbi.h  
C skew\_heap.h  
C stdarg.h  
C stdio.h  
C stdlib.h  
C string.c  
C string.h  
C unistd.h  
> obj  
> tools  
> user  
> libs  
ASM initcode.S  
C panic.c  
C stdio.c  
C syscall.c  
C syscall.h  
C ulib.c  
C ulib.h  
C umain.c  
C ex1.c  
C ex2.c  
C ex3.c

LN 35, Col 7 Spaces: 4 UTF-8 LF C Linux

```
32 int getpid(void);
33 unsigned int gettime_msec(void);
34 ?
35 int set_good(int); -----^
36
37 #endif /* !__USER_LIBS_ULIB_H__ */
38
39
```

Activities Visual Studio Code 4月 11 00:18

File Edit Selection View Go Run Terminal Help

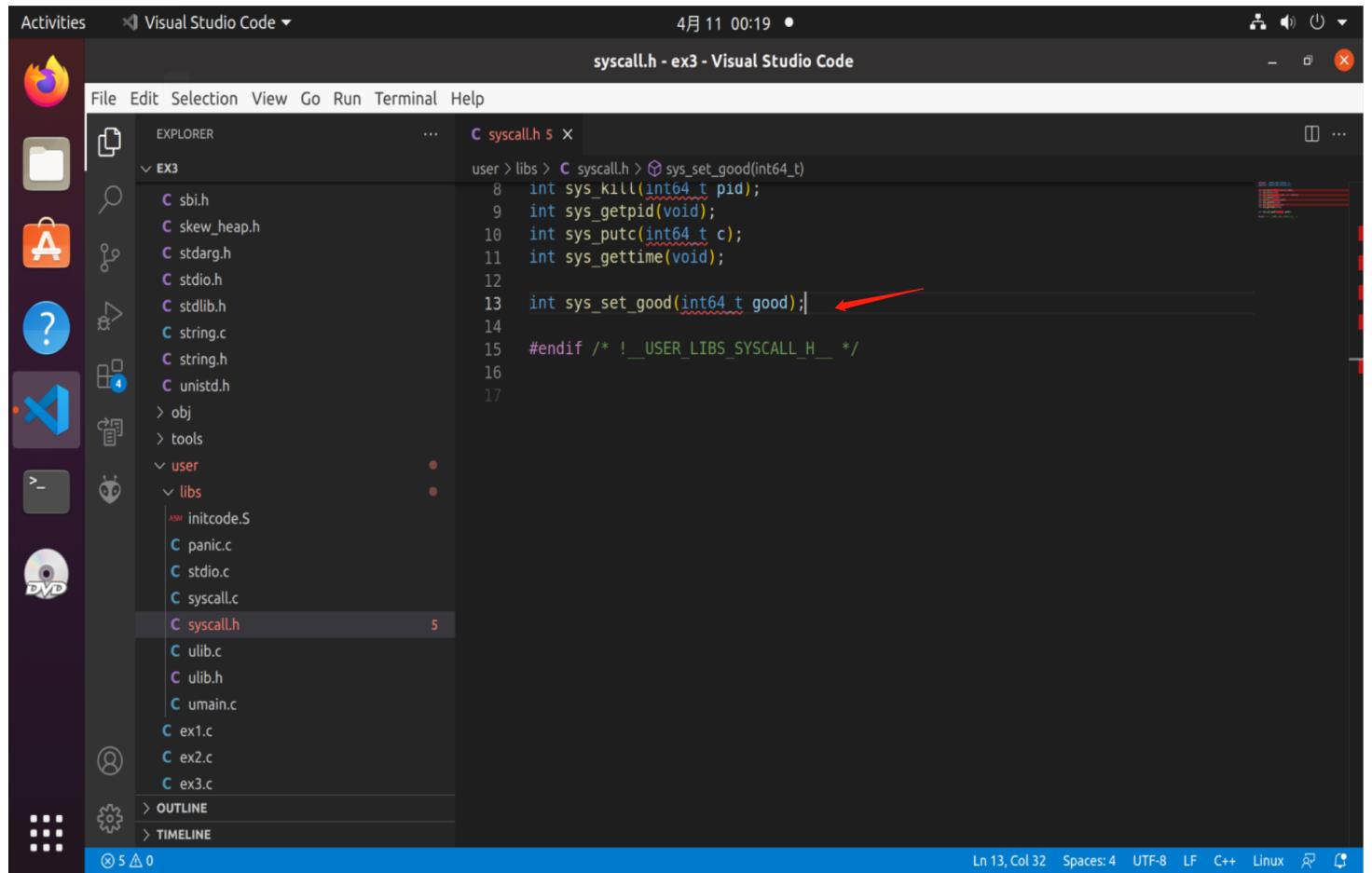
EXPLORER user > libs > C ulib.c > set\_good(int)

C ulib.c  
C sbi.h  
C skew\_heap.h  
C stdarg.h  
C stdio.h  
C stdlib.h  
C string.c  
C string.h  
C unistd.h  
> obj  
> tools  
> user  
> libs  
ASM initcode.S  
C panic.c  
C stdio.c  
C syscall.c  
C syscall.h  
C ulib.c  
C ulib.h  
C umain.c  
C ex1.c  
C ex2.c  
C ex3.c

LN 51, Col 18 Spaces: 4 UTF-8 LF C Linux

```
48
49 int
50 set_good(int good) {
51     return sys_set_good(good);
52 }
```

Write some code in /user/libs/syscall.h and /user/libs/syscall.c



The screenshot shows the Visual Studio Code interface with the following details:

- Title Bar:** Activities > Visual Studio Code > syscall.h - ex3 - Visual Studio Code
- Header Bar:** File Edit Selection View Go Run Terminal Help
- Left Sidebar (Explorer):** Shows the project structure under EX3:
  - user > libs > syscall.h (selected)
  - sbi.h
  - skew\_heap.h
  - stdarg.h
  - stdio.h
  - stdlib.h
  - string.c
  - string.h
  - unistd.h
  - > obj
  - > tools
  - > user
  - > libs
    - ASM initcode.S
    - panic.c
    - stdio.c
    - syscall.c
    - syscall.h (selected)
    - ulib.c
    - ulib.h
    - umain.c
    - ex1.c
    - ex2.c
    - ex3.c
- Right Side Panel:** Shows a preview of the assembly code for the selected file.
- Bottom Status Bar:** Ln 13, Col 32 Spaces: 4 UTF-8 LF C++ Linux

A red arrow points to the line of code in the editor:

```
user > libs > C syscall.h > sys_set_good(int64_t)
8 int sys_kill(int64_t pid);
9 int sys_getpid(void);
10 int sys_putc(int64_t c);
11 int sys_gettime(void);
12
13 int sys_set_good(int64_t good);|
```

Activities Visual Studio Code 4月 11 00:20

syscall.c - ex3 - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER

C syscall.h 5 C syscall.c 5

user > libs > C syscall.c > sys\_set\_good(int64\_t)

```
67 }
68
69 int
70 sys_putc(int64_t c) {
71     return syscall(SYS_putc, c);
72 }
73
74 int
75 sys_gettime(void) {
76     return syscall(SYS_gettime);
77 }
78
79 int
80 sys_set_good(int64_t good) {
81     return syscall(SYS_SET_GOOD, good);
82 }
```

Ln 82, Col 2 Spaces: 4 UTF-8 LF C Linux

Notice that the constant `SYS_SET_GOOD` seems never being defined. So in `/libs/unistd.h`, add a line of code. Set 6 as requested.

```
unistd.h - ex3 - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
EX3
> init
> libs
> mm
> process
> schedule
  default_sched.c
  default_sched.h
  sched.c
  sched.h
> sync
  syscall
    syscall.c
    syscall.h
  trap
  libs
    atomic.h
    defs.h
    elf.h
    error.h
    hash.c
    list.h
    printfmt.c
    rand.c
> OUTLINE
> TIMELINE
C unistd.h X
libs > C unistd.h > SYS_SET_GOOD
21 #define SYS_putc      30
22 #define SYS_pgdir     31
23 /*only for labschedule*/
24 #define SYS_labschedule_set_priority 255
25
26 #define SYS_SET_GOOD    6 ←
27
28 /* SYS_fork flags */
29 #define CLONE_VM        0x00000100 // set if VM shared between processes
30 #define CLONE_THREAD     0x00000200 // thread group
31
32 #endif /* !__LIBS_UNISTD_H__ */
33
34
```

# In kernel mode

We need to add some code in /kern/syscall/syscall.c

Activities Visual Studio Code 4月 11 00:51

sySCALL.c - ex3 - Visual Studio Code

```
kern > syscall > C syscall.c [e] syscalls
64 }
65
66 static int sys_set_good(uint64_t arg[]){
67     do_set_good((int)arg[0]);
68     return 0;
69 }
70
71 static int (*syscalls[])(uint64_t arg[]) = {
72     [SYS_exit]           sys_exit,
73     [SYS_fork]            sys_fork,
74     [SYS_wait]            sys_wait,
75     [SYS_exec]            sys_exec,
76     [SYS_yield]           sys_yield,
77     [SYS_kill]             sys_kill,
78     [SYS_getpid]          sys_getpid,
79     [SYS_putc]             sys_putc,
80     [SYS_gettime]          sys_gettime,
81     [SYS_SET_GOOD]         sys_set_good,
82 };
83 }
84
85 #define NUM_SYSCALLS      ((sizeof(syscalls)) / (sizeof(syscalls[0])))
86
87 void
88 syscall(void) {
89     struct trapframe *tf = current->tf;
90     uint64_t arg[5];
91     int num = tf->gpr.a0;
92     if (num >= 0 && num < NUM_SYSCALLS) {
```

Ln 83, Col 3 Spaces: 4 UTF-8 LF C Linux

In /kern/process/proc.h and /kern/process/proc.c

Activities Visual Studio Code 4月 10 23:36 • proc.h - ex3 - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER

EX3

- kern
  - debug
  - driver
  - fs
  - init
  - libs
  - mm
- process
  - entry.S
  - proc.c
  - proc.h
  - switch.S
  - schedule
  - sync
  - syscall
    - syscall.c
    - syscall.h
  - trap
  - libs
  - tools
- user
  - libs
    - initcode.S

C/C++ Configuration

```
1 kern > process > C proc.h > do_set_good(int)
2     #define le2proc(le, member) \
3         to_struct((le), struct proc_struct, member)
4
5     extern struct proc_struct *idleproc, *initproc, *current;
6
7     void proc_init(void);
8     void proc_run(struct proc_struct *proc);
9     int kernel_thread(int (*fn)(void *), void *arg, uint32_t clone_flags);
10
11    char *set_proc_name(struct proc_struct *proc, const char *name);
12    char *get_proc_name(struct proc_struct *proc);
13    void cpu_idle(void) __attribute__((noreturn));
14
15    //FOR labschedule, set the process's priority (bigger value will get more CPU time)
16    void labschedule_set_priority(uint32_t priority);
17
18
19
20    struct proc_struct *find_proc(int pid);
21    int do_fork(uint32_t clone_flags, uintptr_t stack, struct trapframe *tf);
22    int do_exit(int error_code);
23    int do_yield(void);
24    int do_execve(const char *name, size_t len, unsigned char *binary, size_t size);
25    int do_wait(int pid, int *code_store);
26    int do_kill(int pid);
27
28    void do_set_good(int good); ←
29
30    #endif /* !__KERN_PROCESS_PROC_H__ */
31
32
33
34
35
36
37
38
39
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
59 Ln 99, Col 15 Spaces: 4 UTF-8 LF C Linux
```

Activities Visual Studio Code 4月 10 23:35 • proc.c - ex3 - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER

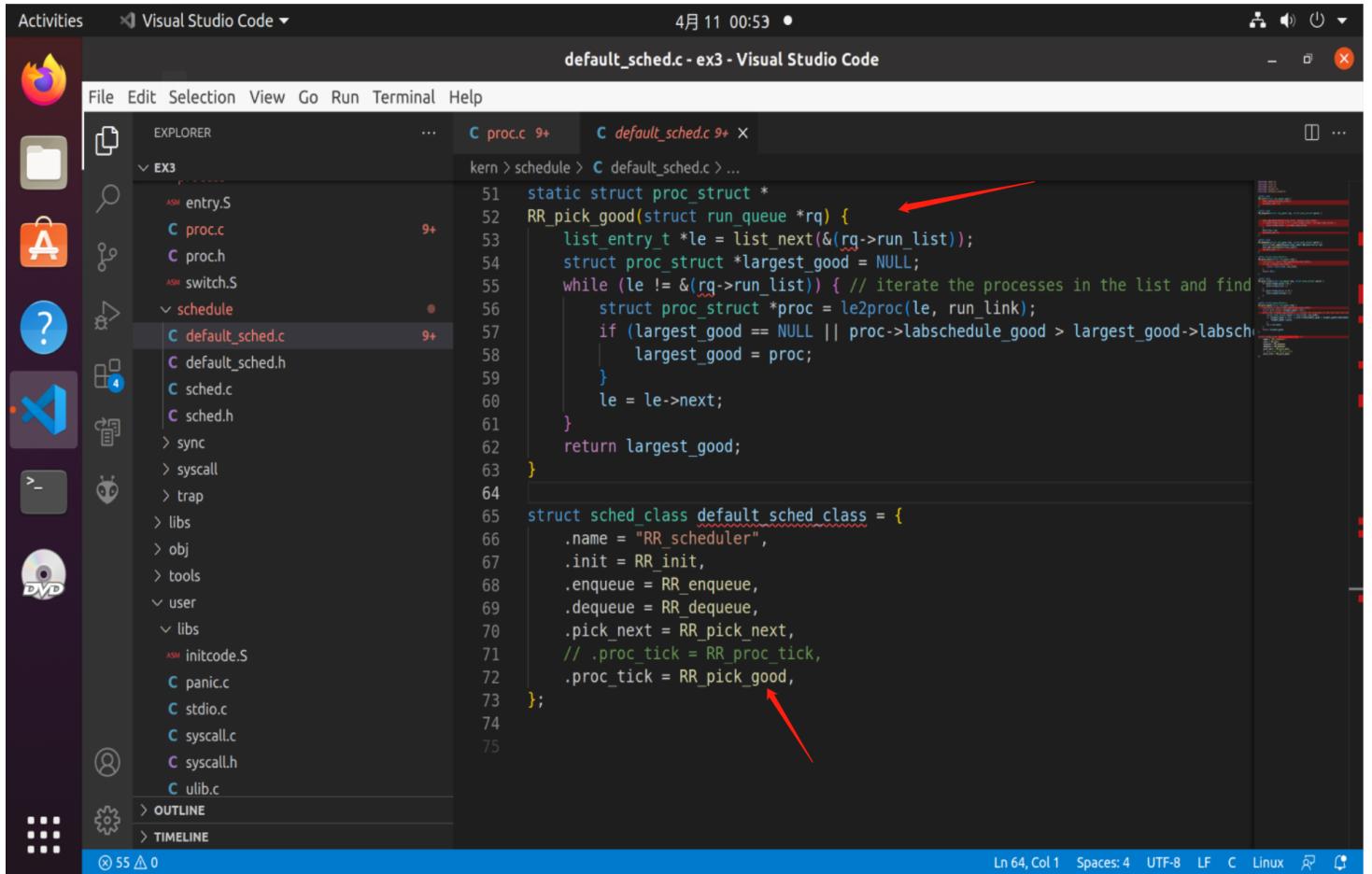
EX3

- kern
  - debug
  - driver
  - fs
  - init
  - libs
  - mm
- process
  - entry.S
  - proc.c
  - proc.h
  - switch.S
  - schedule
  - sync
  - syscall
    - syscall.c
    - syscall.h
  - trap
  - libs
  - tools
- user
  - libs
    - initcode.S

C/C++ Configuration

```
1 kern > process > C proc.c > kernel_execve(const char *, unsigned char *, size_t)
2     return -E_KILLED;
3
4     return -E_INVALID;
5
6
7     void
8     do_set_good(int good) { ←
9         current->labschedule_good = good;
10        current->need_resched = 1;
11        cprintf("set good to %d\n", good);
12    }
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
39
40
41
42
43
44
45
46
47
48
49
49
50
51
52
53
54
55
56
57
58
59
59 Ln 746, Col 20 Spaces: 4 UTF-8 LF C Linux
```

At this time, a process can syscall to modify `labschedule_good`. The next task we need to do is to finish process scheduling. (in /kern/schedule/default\_sched.c)



Activities Visual Studio Code • 4月 11 00:53 • default\_sched.c - ex3 - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER C proc.c 9+ C default\_sched.c 9+ X

kern > schedule > C default\_sched.c > ...

```
51 static struct proc_struct *  
52 RR_pick_good(struct run_queue *rq) {  
53     list_entry_t *le = list_next(&(rq->run_list));  
54     struct proc_struct *largest_good = NULL;  
55     while (le != &(rq->run_list)) { // iterate the processes in the list and find  
56         struct proc_struct *proc = le2proc(le, run_link);  
57         if (largest_good == NULL || proc->labschedule_good > largest_good->labsche  
58             | largest_good = proc;  
59         }  
60         le = le->next;  
61     }  
62     return largest_good;  
63 }  
64  
65 struct sched_class default_sched_class = {  
66     .name = "RR_scheduler",  
67     .init = RR_init,  
68     .enqueue = RR_enqueue,  
69     .dequeue = RR_dequeue,  
70     .pick_next = RR_pick_next,  
71     // .proc_tick = RR_proc_tick,  
72     .proc_tick = RR_pick_good,  
73 };  
74  
75
```

Ln 64, Col 1 Spaces: 4 UTF-8 LF C Linux

Do not forget here!!

Activities Visual Studio Code • 4月 11 00:54

proc.c - ex3 - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER

proc.c 9+ x default\_sched.c 9+

kern > process > proc.c > user\_main(void \*)

```
766 #define KERNEL_EXECVE2(x, xstart, xsiz...  
767 | | | extern unsigned char xstart[], xsiz...  
768 | | | KERNEL_EXECVE(#x, xstart, (size_t)xsiz...  
769 | | }  
770  
771 #define KERNEL_EXECVE2(x, xstart, xsiz...  
772 | | | _KERNEL_EXECVE2(x, xstart, xsiz...  
773 // user_main - kernel thread used to exec a user program  
774 static int  
775 user_main(void *arg) {  
776 #ifdef TEST  
777 | | KERNEL_EXECVE2(TEST, TESTSTART, TESTSIZE);  
778 #else  
779 | | // KERNEL_EXECVE(rr);  
780 | | KERNEL_EXECVE(ex3);  
781 #endif  
782 | | panic("user_main execve failed.\n");  
783 }  
784  
785 // init_main - the second kernel thread used to create user_main kernel threads  
786 static int  
787 init_main(void *arg) {  
788 | | size_t nr_free_pages_store = nr_free_pages();  
789 | | size_t kernel_allocated_store = kallocated();  
790  
791 | | int pid = kernel_thread(user_main, NULL, 0);  
792 | | if (pid <= 0) {  
793 | | | panic("create user_main failed.\n");  
794 }
```

Go to Line/Column

Ln 780, Col 22 (3 selected) Spaces: 4 UTF-8 LF C Linux

The result is

Activities Visual Studio Code • 4月 11 00:57

ex3 - Visual Studio Code

File Edit Selection View Go Run Terminal Help

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
MIDLEG : 0x0000000000000022  
MIDLEG : 0x0000000000000109  
MPW : 0x0000000000000000-0x000000000000ffff (A)  
OS : 0x000000000000-0xffffffffffff (A,R,W,X)  
OS is loading ...  
  
memory management: default_pmm_manager  
physical memory map:  
memory: 0x00000000, [0x00000000, 0x885fffff].  
scheduler: manager = fifo swap manager  
The next proc is pid1:  
The next proc is pid2:  
kernel execute: pid = 2, name = "ex3".  
Breakpoint  
main: wait pids ok, now need to wait pids.  
The next proc is pid3:  
set good to 3  
The next proc is pid4:  
set good to 4  
The next proc is pid5:  
set good to 5  
The next proc is pid6:  
set good to 6  
The next proc is pid7:  
set good to 7  
The next proc is pid8:  
child id 3, acc 4000001  
The next proc is pid9:  
set good to 9  
The next proc is pid10:  
set good to 10  
The next proc is pid11:  
set good to 11  
The next proc is pid12:  
The next proc is pid13:  
child id 4, acc 4000001  
The next proc is pid14:  
set good to 14  
The next proc is pid15:  
set good to 15  
The next proc is pid16:  
set good to 16  
The next proc is pid17:  
set good to 17  
The next proc is pid18:  
child id 5, acc 4000001  
The next proc is pid19:  
set good to 19  
The next proc is pid20:  
child id 6, acc 4000001  
The next proc is pid21:  
set good to 21  
The next proc is pid22:  
The next proc is pid23:  
child id 7, acc 4000001  
The next proc is pid24:  
main: wait pids over!  
The next proc is pid1:  
All kernel threads have quit.  
The end of init main.  
kernel panic at kern/process/proc.c:413:  
initproc exit.
```

lynchrocket@lynchrocket-virtual-machine:~/Desktop/ex3\$