



Operating System

USER PROGRAM'S GUIDE

May 28, 2010

Adrien Forest
fjxokt@gmail.com

870604-P152

Matthieu Maury
mayeu.tik@gmail.com

860928-P210

Christophe Carasco
carasco.christophe@gmail.com

870606-P812

Contents

1	Shell (named <i>coquille</i>)	3
2	Increment	3
3	Fibonacci	3
4	Text scroller	4
5	Ring	4
6	Dining philosophers	5
7	Change priority	5
8	List the processes	5
9	Kill a process	6
10	Print to the malta display	6
11	Get the process information	6
12	List all the available programs	6

1 Shell (named *coquille*)

This program is the only way for the user to launch programs. It is started at the launch of the kernel.

From the shell, the user can do the following (by creating new processes):

- Change priorities of processes (`chg_prio`).
- Obtain information about present processes (`ps`).
- Output to the Malta LCD display (`malta`).
- Terminate processes (`kill`).
- launch all the user program (`coquille`, `increment`, `fibonacci`, `scroller`, `ring`, `dining philosopher` and `supervision`).

Then, the program should look like this

coquille>

The user can launch a new shell with this command

coquille> coquille

or launch other commands as described in the next sections.

2 Increment

The program prints out an increasing number sequence $1, 2, 3, \dots, N$ to the console (each number being printed on a separate line).

The number, N , to reach at the end of the sequence is set by the user as an argument to the program.

Then, the program should be used like this

coquille> increment N

which will print (with $N = 5$)

```
coquille> increment 5
1
2
3
4
5
```

3 Fibonacci

The program prints out the Fibonacci number serie: $1, 1, 2, 3, 5, 8, 13, \dots$ (each number being printed on a separate line).

The number, N , of numbers of the Fibonacci serie to print out is set by the user as an argument to the program.

Then, the program should be used like this

coquille> fibonacci N

which will prints

```
coquille> fibonacci 5
1
1
2
3
5
```

4 Text scroller

The programs scrolls a predefined text *alpha* on the Malta board LCD display. It displays the first 8 caracters (the Malta screen being able to print 8 caracters). Then it scrolls *alpha* until the last caracters of *alpha* are printed. Finally the caracters disappear one by one until the screen is empty. At this point, the program restart displaying *alpha*.

5 Ring

The program starts a set of n processes, P_1, P_2, \dots, P_n . The processes should be set up in a communications ring, where P_1 sends messages to P_2 , P_2 sends messages to P_3 , and so on until P_n sends messages to P_1 which lead to a ring and the processes can redo the same thing.

Each process displays the message it receives and transmits to the next one.

Then, the program should be used like this

```
coquille> ring n
```

which will prints (with $n = 5$)

```
coquille> ring 5 2
Process 0 sent 'Hello world' to Process 1
Process 1 received 'Hello world' from Process 0
Process 1 sent 'Hello world' to Process 2
Process 2 received 'Hello world' from Process 1
Process 2 sent 'Hello world' to Process 3
Process 3 received 'Hello world' from Process 2
Process 3 sent 'Hello world' to Process 4
Process 4 received 'Hello world' from Process 3
Process 4 sent 'Hello world' to Process 0
Process 0 received 'Hello world' from Process 4
Process 0 sent 'Hello world' to Process 1
Process 1 received 'Hello world' from Process 0
Process 1 sent 'Hello world' to Process 2
Process 2 received 'Hello world' from Process 1
Process 2 sent 'Hello world' to Process 3
Process 3 received 'Hello world' from Process 2
Process 3 sent 'Hello world' to Process 4
Process 4 received 'Hello world' from Process 3
Process 4 sent 'Hello world' to Process 0
Process 0 received 'Hello world' from Process 4
```

6 Dining philosophers

The program solves the well-known problem with resource sharing as follows. A group of philosophers are sitting around a dining table (formed as a circle). Between each pair of philosophers is one fork. Occasionally, one philosopher wants to eat and to do that he needs two forks. This means that two philosophers sitting next to each other can not eat at the same time.

The program allows the philosophers to eat or think at some times without any deadlock situation occurs.

The user specifies the number, p , of philosophers and the number c of cycles (think - eat) each philosopher performs.

Then, the program should be used like this

```
coquille> philosopher  $p$   $c$ 
```

which will prints (with $p = 4$ and $c = 2$)

```
coquille> philosopher 4 2
Process 2 thinks
Process 1 thinks
Process 2 eats
Process 4 thinks
Process 3 thinks
Process 2 thinks
Process 3 eats
Process 1 eats
Process 1 thinks
Process 3 thinks
Process 4 eats
Process 4 thinks
Process 3 eats
Process 1 eats
Process 3 thinks
Process 3 thinks
Process 2 eats
Process 4 eats
Process 2 thinks
Process 4 thinks
```

7 Change priority

User program that change the priority of a process identified by the given pid.

Then, the program should be used like this

```
coquille> chg_prio pid new_prio
```

8 List the processes

User program that displays the currently-running processes.

Then, the program should be used like this

```
coquille> ps
```

which will prints

```
coquille> ps
PID   NAME     PRIORITY  STATUS
0      init      42        WAITFOR
1      scroll     42        WAIT
2      coquille  21        WAITFOR
3      ps        21        RUNNING
```

9 Kill a process

User program that kill a process from its identifier.

Then, the program should be used like this

```
coquille> kill 2
```

which will remove the process from the system and free all its memory allocated.

10 Print to the malta display

User program that request the user for typing some text and print it to the malta display instead of the previous text.

Then, the program should be used like this

```
coquille> malta
```

11 Get the process information

User program that prints some information about a process identified by the given pid.

Then, the program should be used like this

```
coquille> proc_info 2
```

which will prints

```
coquille> proc_info 2
pid      2
name     coquille
priority 21
supervised —
supervisor 0
```

12 List all the available programs

User program that list all the available programs. The list specifies the arguments needed to each program and describes what the process performs.

Then, the program should be used like this

```
coquille> help
```

which will prints

```
coquille> help
List of available user programs
-----
coquille          Spawn a new shell.
increment n       Print a sequence from from 1 to n.
fibonacci n       Print the fibonacci sequence up to n
                  numbers.
scroller          Create a process that scroll a
                  predefined string of the LCD.
ring nb_proc nb_loop Create a ring of nb_proc communicating
                  processes.
philosopher nb_philo nb_loop nb_philo philosophers try to eat/think
                  nb_loop times.
supervision       demonstration of process supervision.
ps               print all the running processes.
chg_prio p pri    Change the priority of the process of
                  pid p with the new priority pri.
tuer p           Kill the process of pid p.
malta str         Print the string str on the malta LCD.
```