Testing Ptask Library

Luca Donzella and Michele Scaletti

We have created 4 types of test to use all the features offered by the library Ptask.

Test Type:

1. **System Test**

   It allows to test the system configuration such as:

   - scheduler algorithm
   - protocol of access to resources
   - the ability to run a task on one or different core each time the task becomes active ;

2. **Activation Task Test**

   It provide the opportunity to try the three different modes of creation-activation:

   - creation with activation
   - creation with deferred activation
   - creation with deferred activation with time offset

3. **Advanced Creation Test**

   It allows you to create a task with all parameters. It also provides the ability to change some parameters during the execution of the task.

4. **Handling Mode Test**

   It allow you to test Handling Mode Changes.

In the following pages show the screenshot of output of the various tests.

System Test

To compile the program test, enter into directory SystemTest/ and type: make.

Before running the tests, remember to become super-user, otherwise Linux will not allow you to create real-time tasks!

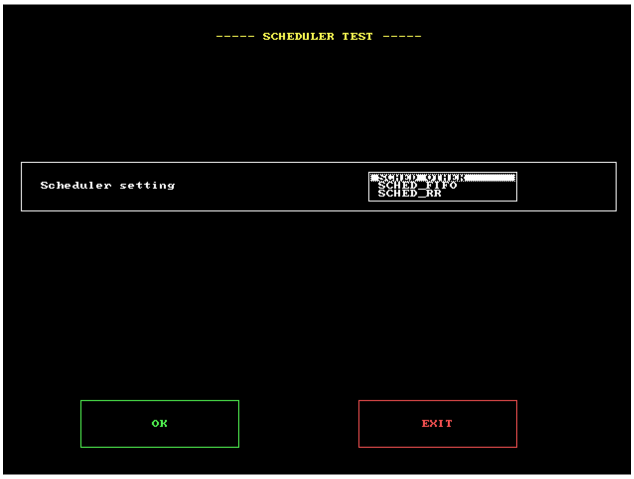
Scheduler Test

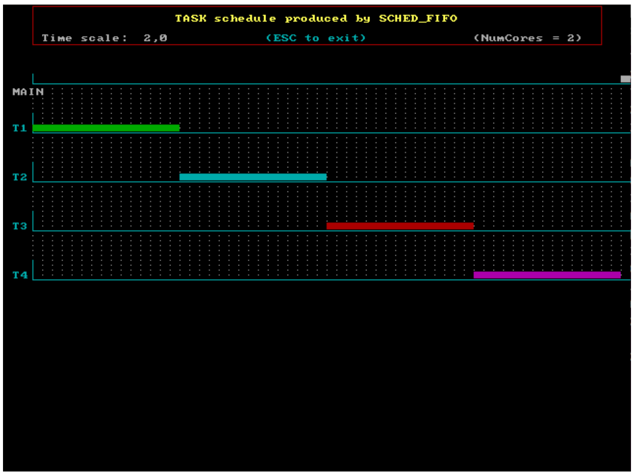
To run Scheduler Test use `taskset` `0x00000001` (mask of processor #0).

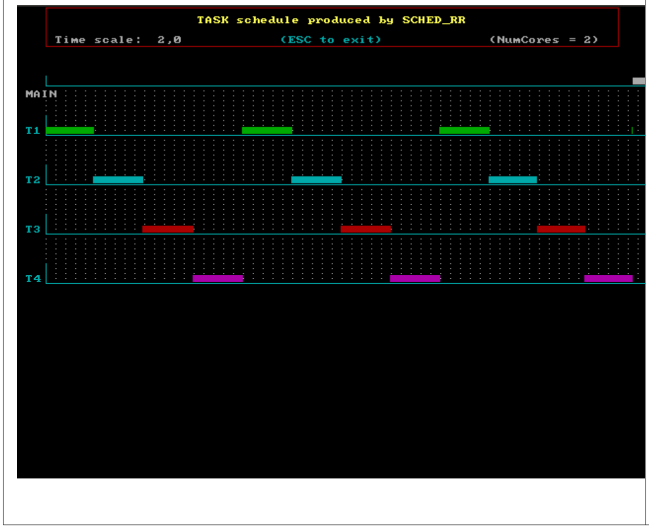
Command example : **`sudo taskset  0x00000001  ./schedulerTest`**

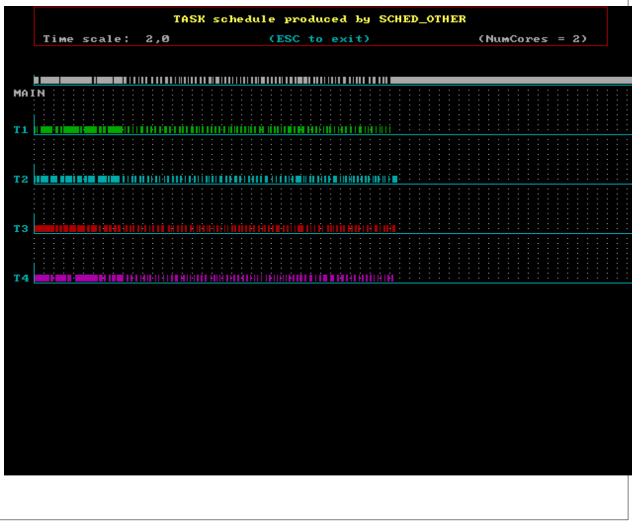
`Taskset` is used to set or retrieve the CPU affinity of a running process given its PID or to launch a new COMMAND with a given CPU affinity.

The parameters of the task that you can insert or modify as priority , period, deadline and computation time are in the file *scheduler.dat* into directory SystemTest/ .





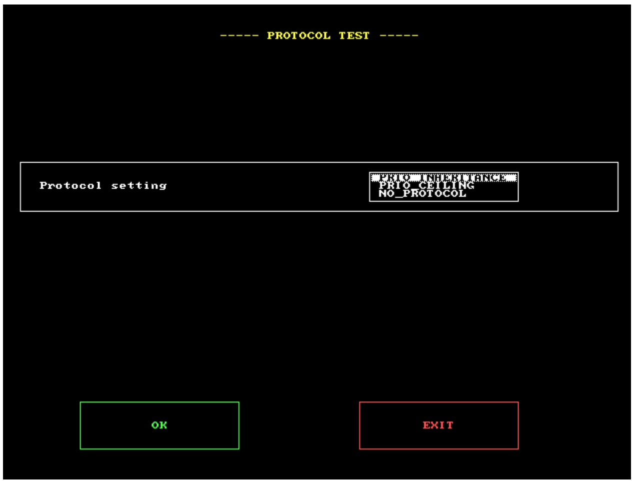




Protocol Test

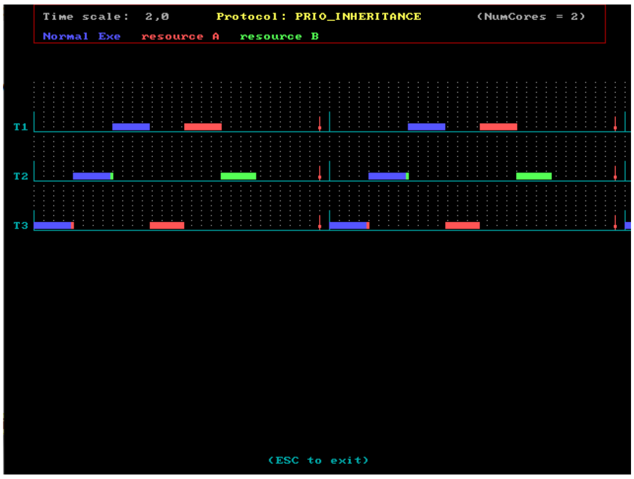
To run Protocol Test use ./protocolTest.

The parameters of the task that you can insert or modify are in the file protocol.dat into directory SystemTest/ .



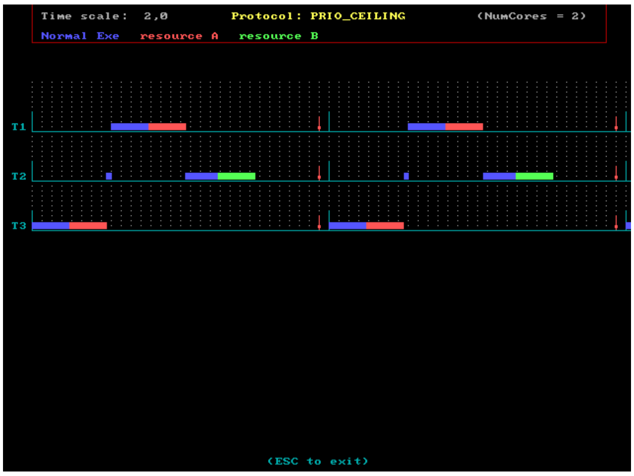
----- PROTOCOL TEST -----

Protocol setting: PRIO_INHERITANCE / PRIO_CEILING / NO_PROTOCOL



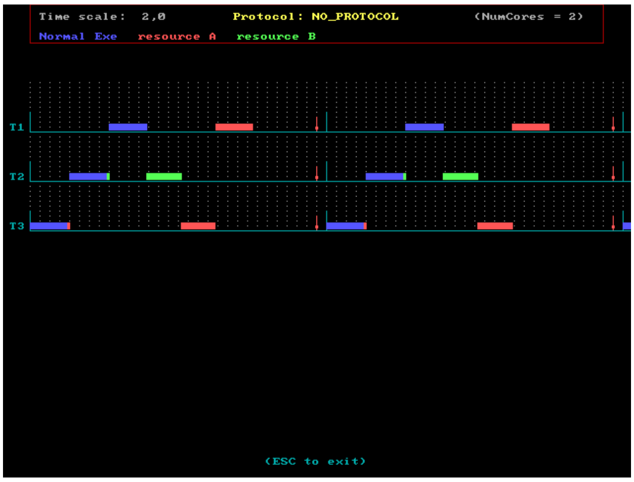
Time scale: 2,0 Protocol: PRIO_INHERITANCE (NumCores = 2)

Normal Exe resource A resource B



Time scale: 2,0 Protocol: PRIO_CEILING (NumCores = 2)

Normal Exe resource A resource B

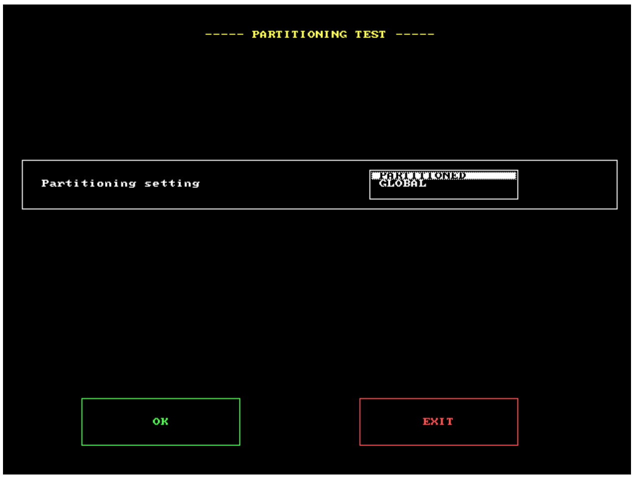


Time scale: 2,0 Protocol: NO_PROTOCOL (NumCores = 2)

Normal Exe resource A resource B

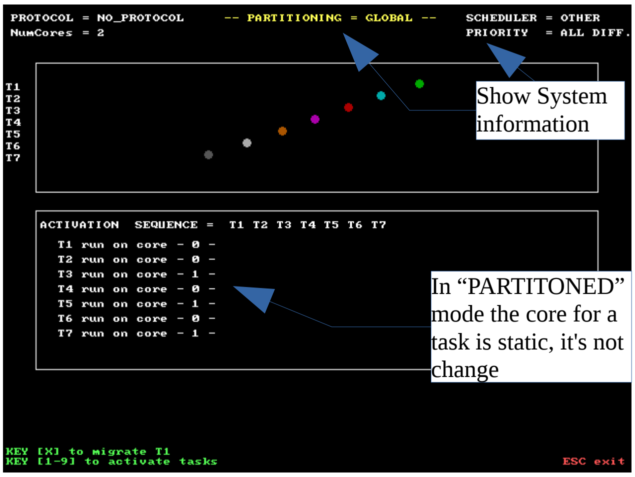
Partitioning Test

To run Partitioning Test use ./partitioningTest.



----- PARTITIONING TEST -----

Partitioning setting: PARTITIONED / GLOBAL



```
PROTOCOL = NO_PROTOCOL      -- PARTITIONING = GLOBAL --     SCHEDULER = OTHER
NumCores = 2                                                PRIORITY  = ALL DIFF.
```

```
ACTIVATION  SEQUENCE =  T1 T2 T3 T4 T5 T6 T7
  T1 run on core - 0 -
  T2 run on core - 0 -
  T3 run on core - 1 -
  T4 run on core - 0 -
  T5 run on core - 1 -
  T6 run on core - 0 -
  T7 run on core - 1 -

KEY [X] to migrate T1
KEY [1-9] to activate tasks                                    ESC exit
```

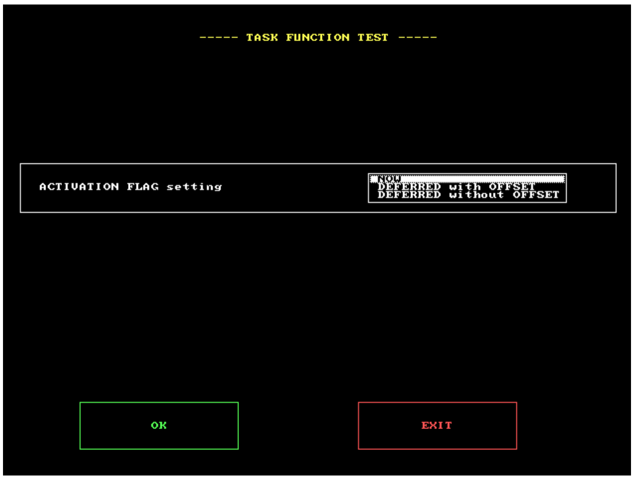
Show System information

In “PARTITONED” mode the core for a task is static, it's not change

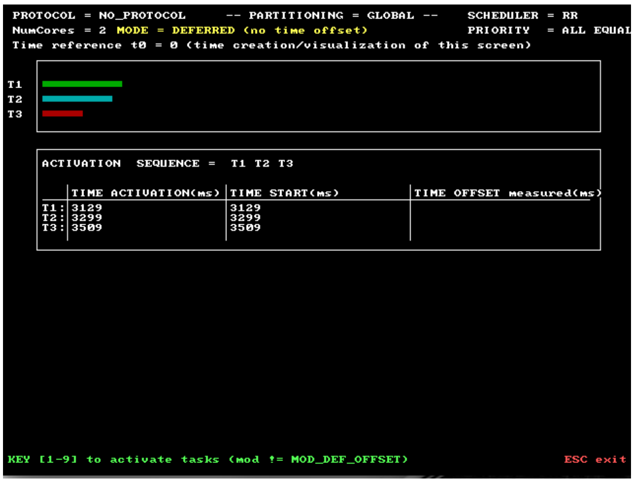
Activation Test

To compile this program test, enter into directory ActivationTaskTest/ and type: make.

To run Activation Test use ./taskFunTest.



```
----- TASK FUNCTION TEST -----

ACTIVATION FLAG setting                    NOW
                                           DEFERRED with OFFSET
                                           DEFERRED without OFFSET
```



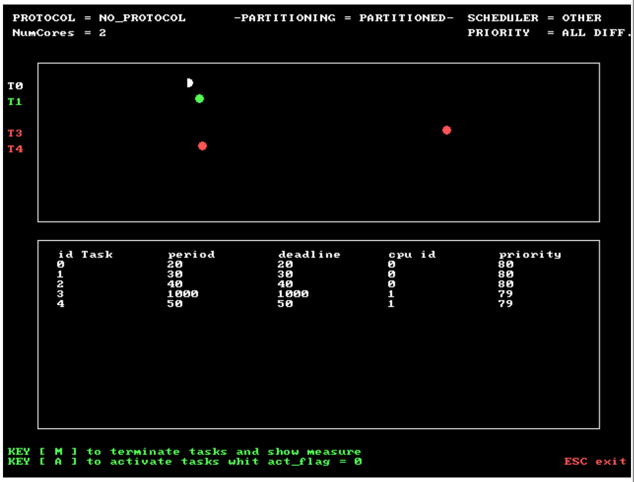
```
PROTOCOL = NO_PROTOCOL      -- PARTITIONING = GLOBAL --     SCHEDULER = RR
NumCores = 2  MODE = DEFERRED (no time offset)              PRIORITY  = ALL EQUAL
Time reference t0 = 0 (time creation/visualization of this screen)

T1
T2
T3

ACTIVATION  SEQUENCE =  T1 T2 T3

   |TIME ACTIVATION(ms)|TIME START(ms)         |TIME OFFSET measured(ms)
T1:|3129               |3129                  |
T2:|3299               |3299                  |
T3:|3509               |3509                  |

KEY [1-9] to activate tasks (mod != MOD_DEF_OFFSET)                  ESC exit
```

Advanced Creation Test

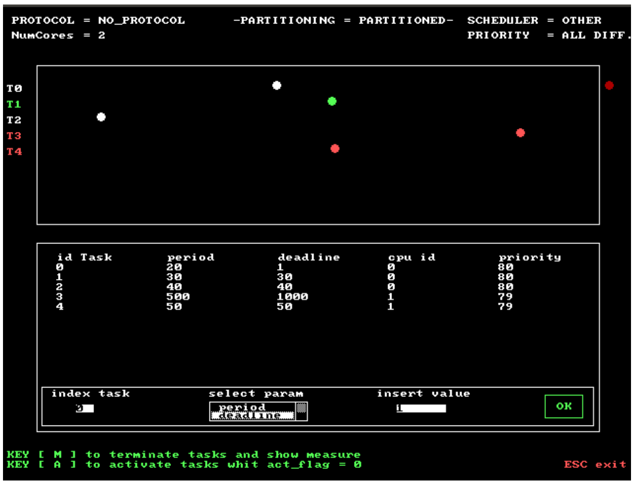
To compile this program test, enter into directory AdvancedTest/ and type: make.

To run Activation Test use ./taskFunTest.

The parameters of the task as : priority , period, deadline and cpu id are in the file param.*dat* into directory AdvancedTest/ .



```
PROTOCOL = NO_PROTOCOL      -PARTITIONING = PARTITIONED-  SCHEDULER = OTHER
NumCores = 2                                              PRIORITY  = ALL DIFF.

T0
T1

T3
T4

   id Task        period        deadline      cpu id        priority
   0              20            20            0             80
   1              30            30            0             80
   2              40            40            0             80
   3              1000          1000          1             79
   4              50            50            1             79

KEY [ M ] to terminate tasks and show measure
KEY [ A ] to activate tasks whit act_flag = 0                       ESC exit
```

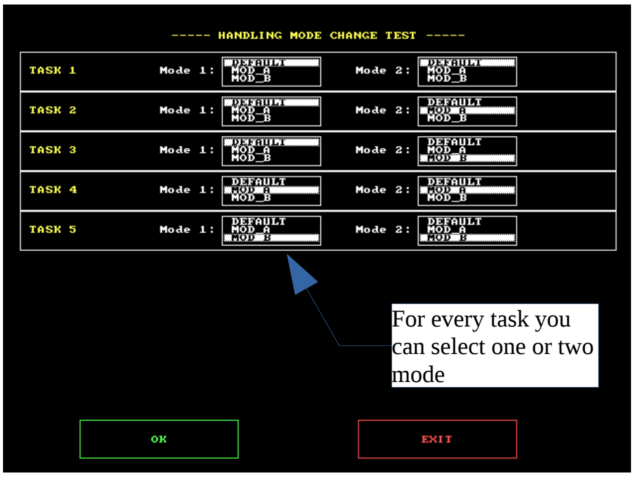


```
PROTOCOL = NO_PROTOCOL      -PARTITIONING = PARTITIONED-  SCHEDULER = OTHER
NumCores = 2                                              PRIORITY  = ALL DIFF.

T0
T1
T2
T3
T4

   id Task        period        deadline      cpu id        priority
   0              20            1             0             80
   1              30            30            0             80
   2              40            40            0             80
   3              500           1000          1             79
   4              50            50            1             79

  index task           select param          insert value
     3                  period                1
                        deadline

KEY [ M ] to terminate tasks and show measure
KEY [ A ] to activate tasks whit act_flag = 0                       ESC exit
```

Handling Mode Test

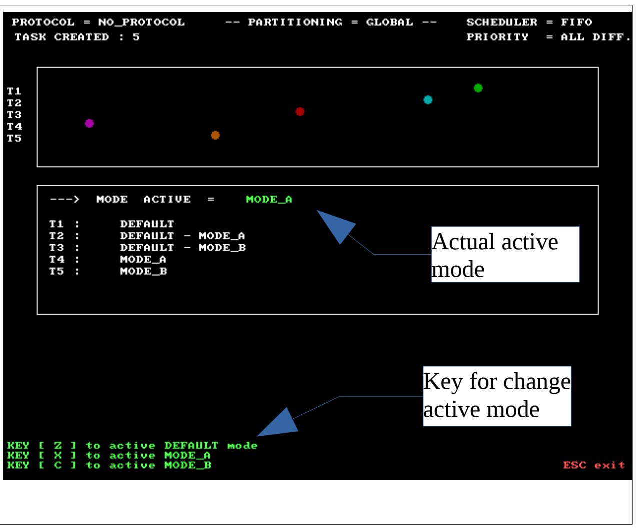
To compile this program test, enter into directory HandlingModeTest/ and type: make.

To run Handling Mode Change Test use ./modeTest.



```
----- HANDLING MODE CHANGE TEST -----

TASK 1        Mode 1: DEFAULT    Mode 2: DEFAULT
                      MOD_A              MOD_A
                      MOD_B              MOD_B
TASK 2        Mode 1: DEFAULT    Mode 2: DEFAULT
                      MOD_A              MOD_A
                      MOD_B              MOD_B
TASK 3        Mode 1: DEFAULT    Mode 2: DEFAULT
                      MOD_A              MOD_A
                      MOD_B              MOD_B
TASK 4        Mode 1: DEFAULT    Mode 2: DEFAULT
                      MOD_A              MOD_A
                      MOD_B              MOD_B
TASK 5        Mode 1: DEFAULT    Mode 2: DEFAULT
                      MOD_A              MOD_A
                      MOD_B              MOD_B
```

For every task you can select one or two mode



```
PROTOCOL = NO_PROTOCOL     -- PARTITIONING = GLOBAL --    SCHEDULER = FIFO
 TASK CREATED : 5                                         PRIORITY  = ALL DIFF.

T1
T2
T3
T4
T5

 --->  MODE  ACTIVE  =    MODE_A

 T1 :     DEFAULT
 T2 :     DEFAULT - MODE_A
 T3 :     DEFAULT - MODE_B
 T4 :     MODE_A
 T5 :     MODE_B

KEY [ Z ] to active DEFAULT mode
KEY [ X ] to active MODE_A
KEY [ C ] to active MODE_B                                              ESC exit
```

Actual active mode

Key for change active mode