

## Лабораторная работа 1.5

### Xenomai Real Time Linux — Алгоритмы диспетчеризации.

**Цель работы:** Знакомство с Xenomai real time Linux, запуск простых задач и исследование различных алгоритмов диспетчеризации.

**Аппаратное и программное обеспечение:** PC, ОС Linux, Virtual Box, Xenomai virtual box image.

#### Порядок выполнения работы:

1. Работа выполняется на локальной или удаленной виртуальной машине с установленным Xenomai. В случае выполнения работы на локальной виртуальной машине запустите VirtualBox и активируйте виртуальную машину с установленным Xenomai. Для входа в систему используйте имя пользователя **root** и пустой пароль. В случае использования удаленной виртуальной машины запустите терминал и введите команду:

**ssh -l root ip\_адрес\_удаленной\_машины**

2. Если необходимо, создайте рабочий каталог, в качестве имени каталога необходимо указать фамилии выполняющих работу, например:

**mkdir Ivanov\_Petrova**

3. В рабочем каталоге необходимо создать файл lab15.c содержащий код программы запуска трех задач в режиме диспетчеризации FIFO. Пример кода:

```
#include <stdio.h>
#include <signal.h>
#include <unistd.h>
#include <sys/mman.h>

#include <native/task.h>
#include <native/timer.h>
#include <native/sem.h>

#include <rtdk.h>
#include <sys/io.h>

#define NTASKS 3

RT_TASK demo_task[NTASKS];
RT_SEM mysync;

// set new baseperiod for the clock :
// - clock only increments counter when a base period is passed
// - the unit of the clock counter is called a jiffie, which in fact
//   represents that a baseperiod is passed
//   e.g. If 10 baseperiods are passed, the clock gives an increase
//   in 10 jiffies

#define BASEPERIOD 1e6 // baseperiod in ns.

// when base period is set, all times in the api are expressed in jiffies
#define EXEETIME 200 // execution time in jiffies
#define SPINTIME 10 // spin time in jiffies

void demo(void *arg)
{
    RTIME starttime, runtime;
```

```

int num=*(int *)arg;

RT_TASK *curtask;
RT_TASK_INFO curtaskinfo;

rt_printf("Task : %d\n",num);
rt_sem_p(&mysync,TM_INFINITE);

// let the task run RUNTIME(=200) jiffies in steps of SPINTIME(=20) jiffies
runtime = 0;

while(runtime < EXECTIME) {
    rt_timer_spin(SPINTIME*BASEPERIOD); // spin cpu doing nothing

    // note: rt_timer_spin function does not accept jiffies only nanoseconds
    //       deviates from timing conventions throughout the Xenomai API

    runtime = runtime + SPINTIME;

    rt_printf("Running Task : %d at time : %d\n",num,runtime);
}
rt_printf("End Task : %d\n",num);
}

//startup code
void startup()
{
    int i;
    char str[10];

    // semaphore to sync task startup on
    rt_sem_create(&mysync,"MySemaphore",0,S_FIFO);

    // change to period mode because round robin does not work
    // in one shot mode
    rt_timer_set_mode(BASEPERIOD);// set tick period

    for(i=0; i < NTASKS; i++) {
        rt_printf("start task : %d\n",i);
        sprintf(str,"task%d",i);

        rt_task_create(&demo_task[i], str, 0, 50, 0);
        rt_task_start(&demo_task[i], &demo, &i);
    }

    rt_printf("wake up all tasks\n");
    rt_sem_broadcast(&mysync);

}

void init_xenomai() {
    /* Avoids memory swapping for this program */
    mlockall(MCL_CURRENT|MCL_FUTURE);

    /* Perform auto-init of rt_print buffers if the task doesn't do so */
    rt_print_auto_init(1);
}

int main(int argc, char* argv[])
{
    printf("\nType CTRL-C to end this program\n\n");

    // code to set things to run xenomai
    init_xenomai();

    //startup code
    startup();

    // wait for CTRL-c is typed to end the program
    pause();
}

```

Выполните компилирование программы. Запустите программу (подробные инструкции по компилированию и запуску программ см в методических указаниях к лабораторной работе 1.1).

4. Измените режим диспетчеризации задач на Round-robin. Изменение режима диспетчеризации для задачи осуществляется с помощью вызова функции

```
rt_task_set_mode(0,T_RRB ,NULL);
```

также необходимо установить период выполнения задачи (timeslice) с помощью функции

```
rt_task_slice(task_descr, quantum);
```

где task\_descr — идентификатор задачи, quantum — период выполнения. Измените код таким образом, чтобы задачи выполнялись «по кругу» (0 1 2 0 1 2 и т. д.). Выполните компилирование и запустите программу.

#### **Источники информации:**

[www.xenomai.org](http://www.xenomai.org)

[www.google.ru](http://www.google.ru)