Лабораторная работа 1.4
Xenomai Real Time Linux — Межзадачное взаимодействие.

Цель работы: Знакомство с Xenomai real time Linux, запуск простых задач и исследование механизмов межзадачного взаимодействия.

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 lab14.c содержащий код программы запуска двух задач и обмена сообщениями между данными задачами. 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 <native/mutex.h>
#include <native/queue.h>

#include  <rtdk.h>

#define NTASKS 2
#define QUEUE_SIZE 255
#define MAX_MESSAGE_LENGTH 40

RT_TASK task_struct[NTASKS];

#define QUEUE_SIZE 255

RT_QUEUE myqueue;

void taskOne(void *arg)
{
   int retval;
   char message[] = "Message from taskOne";

   /* send message */
   retval = rt_queue_write(&myqueue,message,sizeof(message),Q_NORMAL);

   if (retval < 0 ) {
     rt_printf("Sending error\n");
```

```
    } else {
      rt_printf("taskOne sent message to mailbox\n");
    }
}

void taskTwo(void *arg)
{
    int retval;
    char msgBuf[MAX_MESSAGE_LENGTH];

    /* receive message */
    retval = rt_queue_read(&myqueue,msgBuf,sizeof(msgBuf),TM_INFINITE);

    if (retval < 0 ) {
         rt_printf("Receiving error\n");
    } else {
        rt_printf("taskTwo received message: %s\n",msgBuf);

        rt_printf("with length %d\n",retval);
    }
}

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

  void (*task_func[NTASKS]) (void *arg);
  task_func[0]=taskOne;
  task_func[1]=taskTwo;

  rt_queue_create(&myqueue,"myqueue",QUEUE_SIZE,10,Q_FIFO);

  rt_timer_set_mode(0);// set timer to tick in nanoseconds and not in jiffies

  for(i=0; i < NTASKS; i++) {
    rt_printf("start task  : %d\n",i);
    sprintf(str,"task%d",i);
    rt_task_create(&task_struct[i], str, 0, 50, 0);
    rt_task_start(&task_struct[i], task_func[i], &i);
  }
}

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();

  pause();
}
```

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

4. Добавьте в программу две дополнительные задачи — TaskSF и TaskRF. Задача TaskSF должна передавать в качестве сообщения фамилии выполняющих работу. Задача TaskRF должна принимать и выводить в консоль принятое сообщение. Выполните компилирование и запустите программу.

Источники информации:

www.xenomai.org
www.google.ru