Real Time Application Interface

RTAI-3.x

Walter Fetter Lages

w.fetter@ieee.org

Universidade Federal do Rio Grande do Sul

Escola de Engenharia

Departamento de Engenharia Elétrica

ENG04008 Sistemas de Tempo Real

UFRGS
UNIVERSIDADE FEDERAL
DO RIO GRANDE DO SUL

Introdução

- Desenvolvido por Paolo Mantegazza no Instituto Politécnico de Milão a partir do RT-Linux
- Utiliza o conceito de HAL e *pipeline* de interrupções
- Mais ênfase a aplicabilidade do sistema do que à índices de desempenhos teóricos
- Suporta programação em tempo real a nível de usuário (LXRT) e ponto flutuante desde as primeiras versões

UFRGS
UNIVERSIDADE FEDERAL
DO RIO GRANDE DO SUL

RTAI

- Chaveamento entre RTAI e kernel padrão
  - Se o RTAI não está *montado* o kernel normal do Linux é utilizado
- Suporta SMP
- Interface estilo POSIX threads

UFRGS
UNIVERSIDADE FEDERAL
DO RIO GRANDE DO SUL

Módulos Básicos

- `rtai_hal`
  - Serviços básicos, despacho de interrupções e timer.
- `rtai_ksched`
  - Escalonamento de tarefas no kernel
  - link para
- `rtai_lxrt`
  - Escalonamento nos espaços do kernel e do usuário

UFRGS
UNIVERSIDADE FEDERAL
DO RIO GRANDE DO SUL

IPC

- `rtai_bits` - sincronização composta
- `rtai_fifos` - fifos
- `rtai_mbx` - mailboxes
- `rtai_mq` - fila de mensagens POSIX
- `rtai_msg` - mensagens
- `rtai_netrpc` - RPC
- `rtai_sem` - semáforos
- `rtai_shm` - memória compartilhada
- `rtai_tbx` - typed mailboxes

UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Utilitários

- `rtai_leds` - bits de I/O
- `rtai_math` - funções matemáticas
- `rtai_serial` - comunicação serial
- `rtai_tasklets` - tarefas temporizadas
- `rtai_usi` - interrupções no espaço do usuário
- `rtai_wd` - watchdog
- `rtai_rtdm` - driver de tempo real
- `rtai_16550A` - driver para 16550A
- `rtai_calibrate` - calibração do *timer*
- `rtai_signal` - tratamento de sinais
- `rtai_smi` - SMI workaround

UFRGS
UNIVERSIDADE FEDERAL
DO RIO GRANDE DO SUL

Escalonadores

- UP scheduler
  - Máquinas monoprocessador
  - Baseado no 8254
- SMP scheduler
  - Máquinas SMP
  - Baseado no 8254 ou APIC timer
- MUP scheduler
  - Máquinas SMP com apenas 1 processador
  - Baseado no 8254 ou APIC timer
- LXRT scheduler
  - Escalonamento uniforme nos espaços do kernel e do usuário

UFRGS
UNIVERSIDADE FEDERAL
DO RIO GRANDE DO SUL

Modos do Timer

- One-shot
  - Tarefas temporizadas arbitrariamente
  - Timer programado para gerar interrupção no próximo instante
  - Apropriado para processadores com TSC
- Periódico
  - Tarefas temporizadas em múltiplos do período do timer
  - Período definido na inicialização do timer
  - Apropriado para processadores em TSC

UFRGS
UNIVERSIDADE FEDERAL
DO RIO GRANDE DO SUL

APIs

- API nativa
  - API definida pelo RTAI
- POSIX Threads
  - API pthreads
- API comum
  - API de compatibilidade entre o RTLinux e o RTAI
- LXRT
  - Permite a implementação de tarefas de tempo real no espaço de usuário

Exemplo API Nativa

- Piscar um led no bit 0 da porta 0x378 a 0.5 Hz

```
#include <linux/module.h>
#include <asm/io.h>
#include <asm/segment.h>
#include <rtai_sched.h>

#define DESIRED_TICK 1000000  /* 1ms *
static int lpt=0x378;
module_param(lpt,int,0);
MODULE_PARM_DESC(lpt,"Port for I/O");

static RT_TASK blink_task;
```

UFRGS
UNIVERSIDADE FEDERAL
DO RIO GRANDE DO SUL

Inicialização

```
int init_module(void)
{
    RTIME tick;
    RTIME now;

    rt_task_init(&blink_task,
        blink_thread,
        lpt, /* data */
        2048, /* stack size */
        1, /* priority */
        0, /* FPU flag */
        NULL  /*sgnl hdlr */ );
```

UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Inicialização

```
    rt_set_oneshot_mode();

    tick=start_rt_timer(
        nano2count(DESIRED_TICK));
    now=rt_get_time();
    rt_task_make_periodic(
        &blink_task,
        now+tick,
        1000*tick);
    return 0;
}
```

UFRGS
UNIVERSIDADE FEDERAL
DO RIO GRANDE DO SUL

Finalização

```
void cleanup_module(void)
{
    stop_rt_timer();

    rt_busy_sleep(10000000);

    rt_task_delete(&blink_task);

    outb(0,lpt);
}
```

UFRGS
UNIVERSIDADE FEDERAL
DO RIO GRANDE DO SUL

Tarefa

```
void blink_thread(int port)
{
     char data=0;
     for(;;)
     {
          outb(data,port);
          data=~data;
          rt_task_wait_period();
     }
}
```

UFRGS
UNIVERSIDADE FEDERAL
DO RIO GRANDE DO SUL

Exemplo POSIX

```
#include <linux/module.h>
#include <asm/io.h>
#include <asm/segment.h>
#include <rtai_sched.h>
#include <rtai_posix.h>

#define DESIRED_TICK 1000000  /* 1ms *
static int lpt=0x378;
module_param(lpt,int,0);
MODULE_PARM_DESC(lpt,"Port for I/O");
static RTIME rttick;
volatile int end=0;
```

Inicialização POSIX

```
int init_module(void)
{
    pthread_t blink_id;
    rt_set_oneshot_mode();
    rttick=start_rt_timer(
        nano2count(DESIRED_TICK));
    pthread_create(&blink_id,
        NULL,
        blink_thread,
        (void *)&lpt);
    return 0;
}
```

Finalização POSIX

```
void cleanup_module(void)
{
     end=1;
     stop_rt_timer();

     rt_busy_sleep(10000000);

     outb(0,lpt);
}
```

UFRGS
UNIVERSIDADE FEDERAL
DO RIO GRANDE DO SUL

Tarefa POSIX

```
void *blink_thread(void *port)
{
     RTIME now;
     char data=0;
     now=rt_get_time();
     rt_task_make_periodic(rt_whoami(),
          now+rttick,1000*rttick);
     while(!end) {
          outb(data,*((int *)port));
          data=~data;
          rt_task_wait_period();
     }
     pthread_exit(0);
     return NULL;
}
```

UFRGS
UNIVERSIDADE FEDERAL
DO RIO GRANDE DO SUL

Exemplo LXRT

```
#include <sys/io.h>
#include <sys/mman.h>

#include <rtai_lxrt.h>

#define DESIRED_TICK 1000000
#define LPT 0x378
```

UFRGS
UNIVERSIDADE FEDERAL
DO RIO GRANDE DO SUL

Exemplo LXRT

```
int main(void)
{
    unsigned long maintsk_name =
                    nam2num("MAIN");
    struct sched_param mainsched;
    RT_TASK *maintsk;
    int period;
    int i;
    char data=0;
```

Exemplo LXRT

```
rt_allow_nonroot_hrt();

mainsched.sched_priority=
    sched_get_priority_max(
                SCHED_FIFO)-1;

sched_setscheduler(0,SCHED_FIFO,
    &mainsched);

mlockall(MCL_CURRENT | MCL_FUTURE)
```

UFRGS
UNIVERSIDADE FEDERAL
DO RIO GRANDE DO SUL

Exemplo LXRT

```
maintsk=rt_task_init(maintsk_name,
                                1,0,0);

rt_set_oneshot_mode();
period=(int)nano2count(
                (RTIME)DESIRED_TICK);
start_rt_timer(period);

rt_make_hard_real_time();

rt_task_make_periodic(maintsk,
      rt_get_time()+period,
      period);
```

Exemplo LXRT

```
        ioperm(LPT,1,1);
        for(i=0; i < 1000; i++)
        {
            outb(data,LPT);
            data=~data;
            rt_task_wait_period();
        }
        ioperm(LPT,1,0);

        rt_make_soft_real_time();
        stop_rt_timer();
        rt_task_delete(maintsk);
        return 0;
}
```

UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Runinfo

- Para executar aplicações de tempo real pode-se utilizar o script `/usr/realtime/bin/rtai-load`
- Informação de como executar a aplicação é obtida do arquivo `.runinfo`
  - Arquivo oculto no diretório corrente
  - Descreve os módulos que devem ser carregados e os programas no espaço do usuário que devem ser executados

UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Formato do `.runinfo`

- Cada linha do `.runinfo` tem a forma

  `target_name:module_dependencies:run_actions:init_comment`

- `target_name` é um nome para o alvo
  - O default é o primeiro alvo
- `module_dependencies` é a lista de módulos a ser carregada
  - Os módulos são separados por + e sem o prefixo `rtai_`
  - O módulo `rtai_hal` é carregado sempre e não precisa ser listado
- `run_actions` é a lista dos comandos do shell a serem executados, separados por ponto-e-vírgula

UFRGS
UNIVERSIDADE FEDERAL
DO RIO GRANDE DO SUL

Ações

- `push <module>`
- `pop [<module-list>]`
- `popall`
- `flush`
- `klog`
- `exec`
- Comandos precedidos por `!` são executados por `sudo`

UFRGS
UNIVERSIDADE FEDERAL
DO RIO GRANDE DO SUL

init_comment

- Mensagem a ser exibida antes da execução das ações
- control_c
  - Mensagem para pressionar ^C

UFRGS
UNIVERSIDADE FEDERAL
DO RIO GRANDE DO SUL

Exemplo

```
lxrtled:lxrt:./lxrtled;popall:
```