

1º Trabalho Prático PIn 2005/06

Driver elementar para RTC

1 Descrição

Desenvolver um driver de carácter para o relógio de tempo real (RTC), que permita consultar e alterar a hora e data de acordo com as especificações seguintes:

1.1

O driver deverá chamar-se **rtci**, de modo que deverá ser identificado em **/proc/devices** por:

```
<Major>    rtci
```

1.2

Terá um parâmetro de linha de comando que permita que o utilizador indique o número Major do dispositivo:

```
insmod rtci.ko [Major = ??]
```

No caso do parâmetro ser omitido o Major deverá ser automaticamente atribuído. No caso do Major especificado pelo utilizador não estar livre o driver não deverá ser inserido, sendo enviada uma mensagem para o registo do sistema com **printk()**.

1.3

Quando o driver for inserido ou removido deverá também ser registada a operação com **printk()**.

1.4

Deverá ser escrito um script, chamado **rtci**, que receba como primeiro parâmetro **start** ou **stop**, e que respectivamente insira ou remova o driver no kernel. Como segundo parâmetro (opcional), este script poderá receber o Major do driver especificado pelo utilizador:

```
rtci <start | stop> [Major = ??]
```

Que deverá ser passado para o driver no momento da inserção.

1.5

Os serviços disponibilizados pelo driver deverão ser:

a) **open** e **release**, de modo que garantam que o driver não poderá ser acedido por mais que um utilizador ao mesmo tempo.

b) **read** que indica a data e hora actual do sistema, no formato:

```
cat /dev/rtci  
  
<dia da semana> DD/MM/AAAA-HH:MM:SS <AM | PM>
```

Por exemplo, para o modo horário de 24 horas:

```
Terça-feira 18/04/2006-20:02:12
```

e para o modo horário de 12 horas:

```
Terça-feira 18/04/2006-08:02:12 PM
```

c) **write** que permite alterar a data e hora do sistema, no formato:

```
cat > /dev/rtdc  
  
DD/MM/AAAA-HH:MM:SS  
^D
```

A string acima assume sempre o modo horário de 24 horas. Se o modo corrente do RTC for 12h deverá ser tomado isso em consideração de modo que a hora seja correctamente ajustada.

Uma opção será passar para o modo 24 horas, ajustar a hora e voltar ao modo inicial. Outra poderá ser subtrair 12 à hora, se esta for maior que 12 horas, e seleccionar o bit mais alto do byte da hora do RTC.

A opção utilizada, podendo ser outra além das duas acima indicadas desde que funcional, deverá ser justificada no relatório.

d) **ioctl** que permita alterar o modo horário de 24 horas para 12 horas e que permita também invocar as operações de escrita e leitura. O ficheiro header a utilizar, com nome “**rtciioctl.h**” deverá ser:

```
/* IOCTLS para rtdc  
*/  
  
#ifndef RTCIOCTL  
#define RTCIOCTL  
  
#include <linux/ioctl.h>  
  
#define IOCID 0x16  
  
// argumento=1 - set 24h, 0 - set 12h  
#define RTC_SET_24H _IOW (IOCID, 1, int)  
  
// return 1 se modo 24h ou 0 se 12h  
#define RTC_GET_24H _IO (IOCID, 2)  
  
// Permite ajustar a hora, com a string passada como argumento  
// Retorna o nº de bytes transferidos ou <0 em caso de erro  
#define RTC_SET _IOR (IOCID, 3, char *)  
  
// Retorna na string argumento a data e hora  
// Retorna o nº de bytes transferidos ou <0 em caso de erro  
#define RTC_GET _IOW (IOCID, 4, char *)  
  
#endif
```

1.6

Deverá ser desenvolvido também um programa em C, que possa ser executado por qualquer utilizador, e que aceda ao RTC através do driver. O programa deverá chamar-se **rtcm.c** e suportar os seguintes parâmetros na linha de comandos:

```
rtcm 12h           Selecciona modo de 12 horas  
rtcm 24h           Selecciona modo de 24 horas
```

```
rtcm read                               Imprime data e hora no formato
                                         indicado em 1.5 b)

rtcm write DD:MM:AA-HH:MM:SS           Ajusta data e hora no formato
                                         indicado 1.5 c)
```

As operações de alteração do modo horário e de leitura e escrita, deverão ser efectuadas usando IOCTL, de acordo com os comandos especificados em 1.5 d).

Se o número de parâmetros não for o correcto, ou não for conhecido deverá ser impressa a mensagem:

```
Usar com:
rtcm [read | 24h | 12h ]
rtcm write DD/MM/AAAA-HH:MM:SS
```

2 Entrega do trabalho

2.1 Data

A data limite de entrega é terça-feira 16 de Maio de 2006. A entrega para além do prazo sofre uma penalização de 0.5 valores por dia.

2.2 Código fonte

O ficheiro com o código fonte do driver deverá ter o nome **rtci.c** e deve ser acompanhado de um *makefile* para gerar o módulo com o nome **rtci.ko**. O módulo deverá ser compilado de modo que com o comando **modinfo** obtenha-se informação pelo menos sobre a licença, a descrição dos parâmetros (indicando o valor por defeito), uma descrição breve do driver, a versão, e os Nomes, N^{os}. e curso dos autores:

```
modinfo rtci.ko

filename:      rtci.ko
license:      GPL
author:       Helder Daniel
version:      1.0
description:   Simple RTC driver
vermagic:     2.6.14-ipipe preempt 386 gcc-3.3
depends:
srcversion:   B805AF6B7B4FFDBB3C1CE8E
```

De qualquer forma, todos os ficheiros devem ter como comentário os Nomes, N^{os}. e curso dos elementos do grupo.

Deve ser entregue também o script **rtci.i** que insere e remove o módulo do kernel, referido em 1.4.

Pode ser entregue por e-mail ou em disquete ao docente do turno prático em que está inscrito o grupo.

2.3 Relatório

Com o código fonte terá de ser entregue um relatório, que deverá incluir um pequeno manual do utilizador e um manual de implementação. Um *template* para este relatório está disponível na página da disciplina:

http://w3.ualg.pt/~hdaniel/pin/avaliacao/relat_template.pdf

<http://w3.ualg.pt/~hdaniel/pin/avaliacao/tp1.pdf>

No manual do utilizador deverá ser descrito tanto o driver **rtci.ko** como a aplicação **rtcm**, referida no ponto 1.6, incluindo os parâmetros da linha de comandos de ambos.

O relatório deverá ser entregue em papel, ao docente do turno prático em que está inscrito o grupo.

3 Avaliação

De acordo com o indicado na aula teórica 1. Será tomada especial atenção à correcção do programa, à eficiência da implementação e à estrutura e clareza do código.

Nota: Funcionalidades não pedidas nos enunciados não serão avaliadas.

4 Bibliografia recomendada

Corbet, Jonathan, Alessandro Rubini e Kroah-Hartman, Greg (2005). “*Linux Device Drivers 3rd edition*”, O’Reilly, <http://lwn.net/Kernel/LDD3/>

Salzman, Peter Jay, Michael Burian and Ori Pomerantz (2005). “*The Linux Kernel Module Programming Guide*”, <http://www.tldp.org/LDP/lkmpg/2.6/lkmpg.pdf>

Aulas teóricas e práticas disponíveis em <http://w3.ualg.pt/~hdaniel/Ed/>, especialmente:

Capítulo 5 e 6 das aulas teóricas.

Ficha prática 4.2 e 5.

Nota:

podem ser utilizados ou adaptados funções e scripts desenvolvidas nas aulas teóricas, práticas ou dadas como código auxiliar para as fichas práticas.