

Tutorial: Instalando RTAI

Renato Besen

11 de setembro de 2007

Sumário

1	Introdução	1
2	Requisitos	3
3	Versões do RTAI	4
4	Baixando o RTAI	5
4.1	Tarball	5
4.2	CVS	5
5	Baixando os sources do Linux	6
6	Aplicando o patch do RTAI	7
7	Configurando o Kernel	8
8	Compilando o Kernel	10
8.1	Ubuntu (Debian like)	10
8.2	Instruções Genéricas	10
9	Configurando o RTAI	12
10	Instalando o RTAI	13
11	Testando a Instalação	14

1 Introdução

Este tutorial é para pessoas que querem instalar o RTAI Linux, porém não tem um grande conhecimento no processo de compilação de um kernel.

A origem deste documento é um projeto de pesquisa do PET Computação - UFSC sobre desenvolvimento de real-time software, com intenção de demonstrar os passos para se ter um sistema capaz de rodar software de tempo real, usando para isso o RTAI (Real-Time Application Interface). O RTAI consiste de um patch no kernel do Linux e alguns módulos, e com isso o escalonamento de processos deixa de ser responsabilidade do kernel e passa a ser do RTAI.

Há pouca (ou nenhuma) documentação em português, e a maioria dos documentos em inglês pressupõe que se tenha um conhecimento prévio no processo de compilação do kernel. Este documento se propõe a cobrir de maneira clara o processo de configuração, compilação e instalação do kernel e do RTAI, mesmo para quem nunca compilou um kernel.

2 Requisitos

Antes de prosseguir com o tutorial, temos que verificar alguns requisitos:

Um computador - :-)

Linux - Para confecção deste tutorial, foi usada a distribuição Xubuntu 7.04, uma distro derivada do Ubuntu, que usa XFCE como interface gráfica. Esta distro prove facilidades (apt-get) para instalar pacotes externos, então caso você esteja usando outra distribuição, consulte os meio adequados para instalar pacotes nela. Logicamente, você precisará de permissões de super-usuário para poder executar os passos deste tutorial.

gcc e g++ 3.x - O desenvolvimento do RTAI é feito com a versão 3 dos compiladores da GNU, portanto não há garantias de que tudo vá funcionar com as versões mais recentes (≥ 4.0). Você pode conferir a versão do gcc e g++ no seu computador executando:

```
$ gcc --version
$ g++ --version
```

Para instalar os compiladores no Xubuntu, execute:

```
$ sudo apt-get install gcc-3.4 g++-3.4
```

Para consultar informações independentes de distribuição para a instalação, entre em <http://gcc.gnu.org/install/>.

make - Preferencialmente uma versão acima da 3.80. Make é a ferramenta responsável por executar os scripts de compilação ("Makefile"). Normalmente, esta ferramenta vem instalada por padrão em em todas as distribuições. Em todo caso, para instalar, execute:

```
$ sudo apt-get install make
```

module-init-tools - Versão ≥ 3.0 . Ferramentas necessárias para carregar módulos, entre outras utilidades. Para instalar, execute:

```
$ sudo apt-get install module-init-tools
```

Após instalar todos os pacotes necessários, podemos agora selecionar as versões do kernel e RTAI, para prosseguirmos a instalação.

3 Versões do RTAI

No momento em que este tutorial está sendo escrito, a última versão estável do RTAI é a 3.5. Ela pode ser encontrada na página do RTAI, mais precisamente em <https://www.rtai.org/RTAI/>.

Caso você queira usar uma versão mais recente, para testar novas funcionalidades ou deseja alguma correção de um bug que não tenha sido portada para a versão estável, você pode baixar a versão Magma do CVS, que é o branch de desenvolvimento do projeto.

Aqui está uma listagem dos branches disponíveis:

Kilauea : branch fechado (antigo)

Magma : Versão de desenvolvimento

rtai : Vazio

Showroom : Contém vários exemplos de aplicações

Stromboli : branch fechado (antigo)

Vesuvio : branch fechado (antigo)

Vulcano : Versão atual (estável)

Você pode encontrar mais informações sobre o repositório em <https://gna.org/cvs/?group=rtai>.

4 Baixando o RTAI

O diretório padrão para deixar o código-fonte do RTAI e do kernel é o `/usr/src`, portanto entre na pasta:

```
$ cd /usr/src
```

Para mexer nesta pasta, será necessário a permissão de super-usuário. No Ubuntu, você pode fazer isso com o seguinte comando:

```
$ sudo -s
```

Em outras distribuições, use o comando "su", e digite a senha de root.

Você pode baixar o RTAI de duas formas: baixar um tarball, ou baixar do CVS.

4.1 Tarball

Para baixar um tarball do RTAI, execute:

```
# wget --no-check-certificate https://www.rtai.org/RTAI/rtai-3.5.tar.bz2
```

E descompacte:

```
# tar xvf rtai-3.5.tar.bz2
```

Após isso, a pasta `rtai-3.5` é criada com os sources do RTAI.

4.2 CVS

Como já foi explicado na seção anterior, você pode preferir baixar a versão de desenvolvimento. Para obter um dos branches do cvs, execute:

```
# cvs -d:pserver:anonymous@cvs.gna.org:/cvs/rtai co <branch>
```

onde `<branch>` deve ser um dos branches citados na seção anterior. Caso este comando peça uma password, simplesmente aperte enter. Após isso, será criada uma pasta `<branch>` com os sources do RTAI.

5 Baixando os sources do Linux

Normalmente o kernel específico de uma distribuição é modificado para melhorar alguns aspectos específicos, e muitas vezes essas mudanças são incompatíveis com o RTAI. Por isso, é necessário baixar um kernel sem patches, que também é conhecido como "kernel vanilla".

Normalmente, o source do linux também é baixado no diretório `/usr/src`.

```
$ cd /usr/src
```

O primeiro passo aqui é decidir qual versão do kernel baixar. Cada versão do RTAI tem suporte para algumas versões do kernel. Para verificar quais são as versões suportadas, veja quais patches estão disponíveis na pasta `<caminho_para_rtai>/base/arch/i386/patches/`. Supondo que você esteja na pasta `/usr/src/` e o RTAI esteja em `/usr/src/rtai-3.5`, você pode executar o seguinte comando:

```
$ ls -1 rtai-3.5/base/arch/i386/patches/
```

Isso resultará numa saída parecida com essa:

```
hal-linux-2.4.32-i386-1.2-07.patch
hal-linux-2.4.33-i386-1.3-03.patch
hal-linux-2.4.34-i386-1.3-03.patch
hal-linux-2.6.15-i386-1.3-07.patch
hal-linux-2.6.16-i386-1.3-08.patch
hal-linux-2.6.17-i386-1.5-02.patch
hal-linux-2.6.19-i386-1.7-01.patch
README
```

O kernel mais novo suportado é o 2.6.19, e é este que será usado como exemplo. Na página <http://kernel.org/> você encontrará os sources de várias versões do kernel. Encontre o link do kernel que você deseja, e execute o seguinte comando:

```
# wget http://www.kernel.org/pub/linux/kernel/v2.6/linux-2.6.19.tar.gz
```

Após baixar o source do kernel, descompacte o mesmo da seguinte forma:

```
# tar xvf linux-2.6.19.tar.gz
```

Agora, você deve ter uma pasta correspondente à versão do kernel que você baixou.

6 Aplicando o patch do RTAI

Aplicar o patch no kernel é muito simples. Considerando que a estrutura de diretórios seja a seguinte:

```
/usr
|-/linux-2.6.19
|-/rtai-3.5
```

Entre na pasta do kernel foi descompactado:

```
# cd linux-2.6.19
```

Agora aplique o patch da seguinte forma:

```
# patch -p1 -b < ../rtai-3.5/base/arch/i386/patches/hal-linux-2.6.19-i386-1.7-01.patch
```

Pronto, o patch do RTAI foi aplicado ao kernel.

A numeração do kernel pode ter um parâmetro adicional, totalizando assim 4 números (e.g. 2.6.19.1), porém os patches do RTAI se destinam apenas às versões com 3 números. Provavelmente o patch funcionará com os kernels com a numeração extra, porém estes kernels não são completamente testados e suportados.

7 Configurando o Kernel

Não entra no escopo deste tutorial verificar todas as opções de configuração do kernel, portanto este tópico será apenas superficial.

A configuração do kernel é a parte mais difícil da instalação. Uma boa idéia é copiar o arquivo de configuração do kernel que está rodando, afinal de contas ele está funcionando.

Existem múltiplas maneiras de como fazer isso. A maioria das distribuições faz uma cópia do arquivo de configurações no diretório /boot. Há também uma opção no kernel que se estiver habilitada criará uma cópia do arquivo em /proc/config.gz, mas a maioria das distribuições não vem com esta opção habilitada.

Entre na pasta onde o kernel foi descompactado:

```
# cd /usr/src/linux-2.6.19
```

O nome do arquivo de configuração pode variar, então adapte ao que existir no seu computador. Execute:

```
# cp /boot/config-2.6.19-generic .config
```

Agora que você possui um arquivo de configuração consistente, pode alterar o que achar necessário no kernel. Se você possuir mais de um compilador na sua distribuição, você deve especificar qual deve ser usado. Para isso, basta adicionar alguns parâmetros quando for configurar o kernel:

```
# make menuconfig CC=/usr/bin/gcc-3.4 CXX=/usr/bin/g++-3.4
```

Lembre-se de adaptar os caminhos ao seu caso.

Agora você deve estar dentro do menu de configuração do kernel. Para carregar o arquivo de configuração que você copiou previamente, entre em "Load Alternate Configuration File", digite ".config" e pressione Enter.

Algumas opções do kernel devem ser verificadas antes de partir para a compilação.

- Processor type and features -> Processor family
Escolha o a família em que o seu processador se encontra. Se o seu processador é da família AMD K7, escolha o K6, por causa de bugs na FPU.
- Processor type and features -> Use register arguments
Esta opção deve ser desabilitada.
- Loadable module support -> Enable loadable module support
Esta opção deve ser habilitada (normalmente está por padrão). Com ela, poderemos carregar os módulos do RTAI mais tarde.

- Loadable module support -> Module versioning support
Esta opção NÃO deve ser habilitada, pois segundo o FAQ de instalação do RTAI, ela causa alguns problemas para o mesmo.
- Kernel hacking -> Compile the kernel with frame pointers
Esta opção deve ser desabilitada.

O processo de compilação do kernel é algo que exige tentativas e erros para chegar a um bom resultado, e quanto mais você conhecer seu hardware, mais você poderá otimizar. Por exemplo, se você sabe qual é a sua placa de rede, escolha a opção correspondente no kernel para ela (de preferência estática, e não como módulo), e desabilite o suporte a todas as outras placas. Desta forma, você diminuirá o tempo de compilação (menos módulos serão compilados), e deixará o kernel mais rápido, pois o suporte à rede é interno, não precisando carregar módulos e fazer chamadas para o mesmo.

8 Compilando o Kernel

8.1 Ubuntu (Debian like)

Caso você não esteja usando Ubuntu, pode conferir os procedimentos genéricos na próxima seção.

No Ubuntu, o melhor modo de instalar um kernel customizado é criar um arquivo ".deb" de instalação do kernel. Para isso, você necessitará de alguns pacotes instalados:

```
$ apt-get install kernel-package fakeroot
```

Tendo estes pacotes instalados, execute os seguintes comandos na pasta onde está o kernel para gerar os pacotes de instalação:

```
# make-kpkg clean
# fakeroot make-kpkg --initrd --append-to-version=-rtai \
kernel_image kernel_headers
```

Após o parâmetro --append-to-version="você pode colocar qualquer nome que achar conveniente.

Dependendo da configuração do kernel, e das configurações do seu computador, a compilação pode demorar algumas horas, portanto seja paciente.

Assim que a compilação terminar, você encontrará dois arquivos ".deb" na pasta /usr/src. Agora você pode instalar o kernel com o seguinte comando:

```
# dpkg -i *.deb
```

Pronto. Se tudo deu certo, quando você rebotar o computador, deverá ver uma nova opção no grub. É só escolher ela, e torcer pra funcionar.

8.2 Instruções Genéricas

O processo de compilação do kernel 2.6 se dá basicamente em 3 comandos para o make:

- clean, para garantir que não existam resíduos de configurações/compilações anteriores;
- bzImage, que gera a imagem do kernel que ficará na pasta /boot;
- modules, que compila os módulos do kernel.

O comando modules_install é usado para instalar os módulos na pasta adequada. Para executar todos os comandos de uma vez, digite:

```
# make clean bzImage modules modules_install
```

Lembrando que os comando devem estar nesta ordem. Como foi falado anteriormente, dependendo das configurações do kernel e da máquina este processo pode demorar vários minutos.

Após este processo, você deverá copiar os arquivos gerados para os lugares apropriados, e configurar o gerenciador de boot, para que a nova opção seja exibida.

```
# cp arch/i386/boot/bzImage /boot/vmlinuz-2.6.19-rtai
# cp arch/i386/Kconfig /boot/Kconfig-2.6.19-rtai
# cp System.map /boot/System.map-2.6.19-rtai
```

Depois de copiar os arquivos, vamos editar o gerenciador de boot, neste caso o GRUB. Existe uma ampla documentação e inúmeros foruns de usuários onde você pode encontrar ajuda, caso use outros gerenciadores (LILO).

Normalmente, o arquivo de configuração do GRUB é /boot/grub/menu.lst, porém em algumas distribuições o arquivo é /boot/grub/grub.conf. Com o bom e velho Ctrl-C Ctrl-V você deve conseguir adicionar uma nova opção ao menu, que deve ser algo parecido com o trecho a seguir, variando de acordo com configurações da máquina (e.g. partições):

```
title Linux RTAI
root (hd0,0)
kernel /boot/vmlinuz-2.6.19-rtai root=/dev/hda1 ro single
```

Lembre, isto é só um exemplo, e pode ter variações dependendo da distribuição. Procure analisar as opções já existentes no arquivo e se basear nelas para criar a nova.

Feito isso, reinicie o computador, escolha a nova opção de boot, e cruze os dedos.

9 Configurando o RTAI

Se tudo deu certo, você deve estar com o seu novo kernel funcionando. Partimos agora para a configuração do RTAI.

Depois de entrar na pasta onde o RTAI foi descompactado, o primeiro passo é criar uma nova pasta onde ele será compilado.

```
# cd /usr/src/rtai-3.5
# mkdir build
# cd build
```

Agora, podemos configurar o RTAI propriamente:

```
# make -f ../makefile CC=/usr/bin/gcc-3.4 CXX=/usr/bin/g++-3.4
```

As seguintes opções devem ser verificadas:

- General -> Instalation Directory
Onde o RTAI será instalado. O padrão é /usr/realtime
- General -> Linux Build Tree
Se você não estiver rodando no kernel recém-compilado do RTAI, mude esta opção para algo como /lib/module/<versao-do-kernel-rtai>/build.

Agora você pode salvar e sair da configuração.

10 Instalando o RTAI

A instalação do RTAI é extremamente simples. Primeiro compilamos o RTAI:

```
# make CC=/usr/bin/gcc-3.4 CXX=/usr/bin/g++-3.4
```

Após o processo de compilação, instalamos com o comando:

```
# make install
```

Pronto. Se tudo isso ocorreu sem problemas, reinicie o computador, escolha o kernel do RTAI, e agora temos um sistema RTAI funcional rodando.

11 Testando a Instalação

Por padrão, as aplicações de teste ficam em `/usr/realtime/testsuite`. Vamos executar o teste de latência em modo usuário e em modo kernel, para verificarmos se o sistema funciona como deveria.

Para testar em modo usuário, execute:

```
# cd /usr/realtime/testsuite/user/latency
# ./run
```

Com isso, você verá algo assim na sua tela:

```
RTAI Testsuite - USER latency (all data in nanoseconds)
2007/08/15 13:30:48
RTH|   lat min|   ovl min|   lat avg|   lat max|   ovl max|   overruns
RTD|  -1229|  -1229|   3436|   8852|   8852|           0
RTD|   -869|  -1229|   3443|   6812|   8852|           0
RTD|  -1220|  -1229|   3393|   6708|   8852|           0
RTD|  -1092|  -1229|   3436|  10219|  10219|           0
RTD|  -1073|  -1229|   3445|   8541|  10219|           0
```

Agora, testaremos em modo kernel:

```
# cd /usr/realtime/testsuite/kern/latency
# ./run
```

Temos uma saída parecida no modo kernel:

```
RTAI Testsuite - KERNEL latency (all data in nanoseconds)
RTH|   lat min|   ovl min|   lat avg|   lat max|   ovl max|   overruns
RTD|  -2084|  -2084|   2418|   4891|   4891|           0
RTD|  -2130|  -2130|   2424|   5779|   5779|           0
RTD|  -2097|  -2130|   2436|   6094|   6094|           0
RTD|  -2090|  -2130|   2434|   8015|   8015|           0
RTD|  -2088|  -2130|   2455|   9054|   9054|           0
```

Em alguns casos, quando você for testar o modo kernel, aparecerá uma mensagem de erro, informando que um dispositivo não foi encontrado:

```
Error opening /dev/rtf3
```

Para resolver este problema, copie e cole o código abaixo em um arquivo, de permissões de execução, e rode o script:

```
#!/bin/bash
mknod -m 666 /dev/rtai_shm c 10 254
for n in `seq 0 31`
do
f=/dev/rtf$n
mknod -m 666 $f c 150 $n
done
```