

Programação de Sistemas de Tempo Real

Laboratório 5 - Instalação do RTOS para execução e simulação em um ambiente de tempo real

*Edson Nascimento Silva - 21350694, *Evaldo Patrik Cardoso - 21453640

I. OBJETIVOS

Este relatório tem como objetivo a instalação de um RTOS para realizarmos simulações de sistemas em um ambiente de tempo real. O qual será utilizado para executarmos as simulações do sistema de controle de um robô móvel no próximo laboratório *Lab 6*. Cujas variáveis internas acabam influenciando na execução do sistema.

Neste novo experimento o robô móvel foi instalados os pacotes para compilação e desenvolvimento do kernel conforme os pré-requisitos do RTOS, foi aplicado no código fonte do Linux o patch do RTOS configurar para compilação de acordo com o hardware.

Além do mais, foi realizado a compilação do kernel, assim com adicionado ao grub de inicialização.

Nos próximos tópicos será aprofundado o problema e sua resolução.

II. PROBLEMA PROPOSTO

De acordo com o enunciado proposto pelo professor era solicitado a instalação de um RTOS.

Contudo a instalação de um Sistema Operacional de Tempo Real (RTOS - Real Time Operating System) não é uma tarefa das mais fáceis, pois aparecem muitas complicações pelo caminho, por exemplo:

- Incompatibilidade de hardware.
- falta de documentações.
- as restrições de RTOSs para sistemas específicos. exemplo: parrot os.
- dependências muito específicas.

Logo para nos livrarmos de todas estas complicações utilizaremos a distribuição linux Ubuntu em nossos experimentos e simulações.

*Engenharia da Computação, Departamento de Eletrônica e Computação, Faculdade de Tecnologia, Universidade Federal do Amazonas, Manaus, AM. E-mail: netosilvan78@gmail.com, evaldo.cardoso@icomp.ufam.edu.br

III. INTRODUÇÃO TEÓRICA

A. RTOS

Sistema Operacional de Tempo Real (RTOS - Real Time Operating System) é um sistema operacional destinado à execução de múltiplas tarefas onde o tempo de resposta a um evento (externo ou interno) é pré-definido.

Eles são voltados para aplicações onde é essencial a confiabilidade e a execução de tarefas em prazos compatíveis com a ocorrência de eventos.

Structure of RTOS

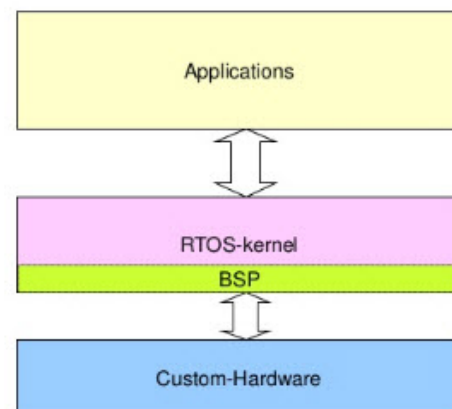


Figura 1. <https://www.elprocus.com/real-time-operating-system-rtos-and-how-it-works/>

Há muitos detalhes em um Sistema Operacional de Tempo Real, por exemplo. O principal onde as rotinas de processamento em geral são bastante especializadas e curtas, pois devem executar a sua tarefa no menor tempo possível. Há um forte paralelismo na execução das atividades e um estabelecimento de prioridades (thread, semáforos), onde as atividades mais prioritárias interrompem as menos prioritárias e assumem o controle do processador as quais são gerenciadas pelo escalonador de processos do sistema.

No caso a divisão de tarefas em rotinas especializadas requer que haja alguma forma de sincronismo ou troca de informação entre elas, por exemplo, para que seja informado que tarefa já foi concluída ou não por outra rotina (flags, semáforos).

Para esclarecermos melhor os conceitos a retina e postura de processos na unidade central de processamento é chamada de troca de contexto.

IV. AMBIENTE

Em nosso ambiente utilizamos os seguintes componentes:

- Distribuição Ubuntu 18.04.3 LTS[1].
- Notebook Acer Aspire E5-574, Intel® Core™ i3-6100U CPU @ 2.30GHz × 4, Intel® HD Graphics 520 (Skylake GT2), 12Gb RAM.
- kernel version linux-4.9.115.tar.gz[2].
- patch-4.9.115-rt93.patch.gz[3].

Tendo em mãos nos componentes podemos iniciar o processo de instalação.

Primeiramente criaremos um diretório de trabalho. Abrimos o terminal linux e digitamos os seguintes comandos:

```
$mkdir ~/kernel && cd ~/kernel
```

depois fazemos a extração linux-4.9.115.tar.gz dentro diretório /kernel:

```
$tar -xzf linux-4.9.115.tar.gz
```

logo em seguida caminhamos para o diretório extraído e adicionamos o patch: \$cd linux-4.9.115

```
$gzip -cd ../patch-4.9.115-rt93.patch.gz | patch -p1 --verbose
```

Pronto com isso já temos o patch do RT, agora vamos instalar algumas dependências necessária para seu bom funcionamento. Obs: senão tiver instalado o gcc e g++ recomendo instalar antes do próximo comando.

continuando vamos instalar o pacote libncurses-dev libssl-dev:

```
$sudo apt-get install libncurses-dev libssl-dev
```

Em seguida usaremos o comando make para iniciar a configuração do RT.

```
$make menuconfig
```

Depois de compilar aparecerá o menu de configuração.

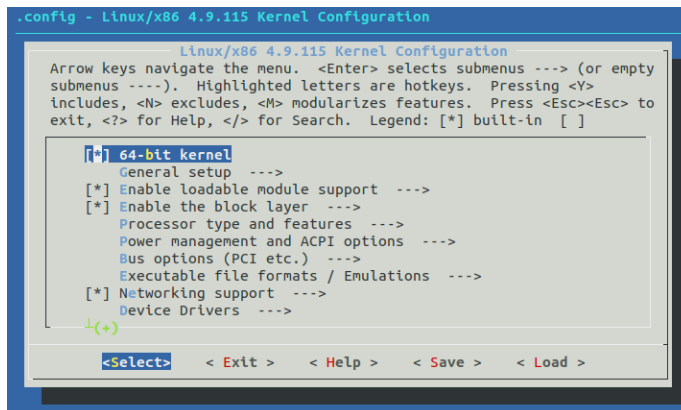


Figura 2. MenuConfig

Ao qual seguiremos os seguintes passos:

```
##Graphical Menu##

Processor type and features --> [Enter]
Preemption Model (Voluntary Kernel Preemption (Desktop)) [Enter]
Fully Preemptible Kernel (RT) [Enter] #Select

[Esc][Esc]

Kernel hacking --> [Enter]
Memory Debugging [Enter]
Check for stack overflows #Already deselected - do not select

[Esc][Esc]

[Right Arrow][Right Arrow]

<Save> [Enter]

.config

<Okay> [Enter]

<Exit> [Enter]

[Esc][Esc]

[Right Arrow]
<Exit> [Enter]
```

Figura 3. passos

Terminado a configuração iniciaremos a compilação do kernel com os seguintes comandos:

```
$make -j20
```

```
$sudo make modules_install -j20
```

```
$sudo make install -j20
```

E para finalizarmos a instalação Verifique se initrd.img-4.9.115-rt93, vmlinuz-4.9.115-rt93 e config-4.9.115-rt93 foram gerados no diretório /boot.

```
$cd /boot
```

caso sim atualize o grub.

```
$sudo update-grub
```

Pronto. finalizada instalação reinicie a máquina e seja feliz.

```
$sudo reboot
```

V. CONCLUSÃO

O objetivo do trabalho foi, entre outras coisas, criar um ambiente RTOS(Sistema operacional de tempo real) para podermos expandir o trabalho realizado no laboratório anterior Lab 4 para utilizar múltiplas threads e aplicar os conceitos de Produtor-Consumidor e ferramentas para a resolução de tais problemas utilizando agora o RT para melhorarmos e validarmos o desempenho de nosso sistema, o qual faz a simulação de um robô que foi particionado em duas tarefas(threads).

VI. REFERÊNCIAS

- [1] Download Ubuntu Desktop.
; <https://ubuntu.com/download/desktop>; Acesso em
20/10/2019 - 15:23P.M
- [2] BLOG DE CES33 - SISTE-
MAS OPERACIONAIS - ITA T.10
; <https://mirrors.edge.kernel.org/pub/linux/kernel/v4.x/linux-4.9.115.tar.gz>; Acesso em 21/11/2019 - 09:23A.M
- [3] Index of /pub/linux/kernel/projects/rt/
; <https://mirrors.edge.kernel.org/pub/linux/kernel/projects/rt/4.9/older/patch-4.9.115-rt93.patch.gz>; Acesso em 21/11/2019 - 10:00A.M