THE DEVELOPER'S CONFERENCE

Trilha Embedded – Escalonador Earliest Deadline First

Rafael de Moura Moreira

Universidade Federal de Itajubá

Globalcode

Objetivos

THE DEVELOPER'S CONFERENCE

- Introduzir o algoritmo Earliest Deadline First, suas vantagens e suas desvantagens
- Introduzir a implementação do EDF no Linux e apontar as diferenças em relação a seus outros escalonadores e outras implementações
- Demonstrar o projeto de um sistema de tempo real baseado no EDF, os seus testes e apresentar os resultados obtidos

Globalcode

Agenda

THE DEVELOPER'S CONFERENCE

- Conceitos básicos
- Earliest Deadline First
- Escalonamento de tarefas no Linux
- Implementação de EDF para sistemas de baixo custo

Globalcode

Agenda

THE DEVELOPER'S CONFERENCE

- Conceitos básicos
  - Kernel
  - Arquitetura de kernel
    - Monolítico
    - Microkernel
  - Sistemas de tempo real
  - Escalonamento de processos
    - Processos
    - Cooperativo x Preemptivo
    - Prioridade estática x dinâmica
    - Time slice

Globalcode

Kernel

THE DEVELOPER'S CONFERENCE

- Gerenciamento de recursos computacionais
  - Processador: escalonamento de processos
  - Memória: particionamento, endereçamento virtual, paginação etc
  - Dispositivos I/O: device drivers

Globalcode

Kernel

THE DEVELOPER'S CONFERENCE

- Camada de abstração para programadores
  - System calls e interrupções
  - Comunicação e sincronização entre processos
  - Device drivers

Globalcode

Arquitetura de kernel

THE DEVELOPER'S CONFERENCE

- Kernel monolítico
  - Serviços do sistema rodam no próprio kernel
  - Vantagens: maior velocidade e eficiência
  - Desvantagens: difícil manutenção, bugs em setores isolados comprometem o sistema todo

Globalcode

Arquitetura de kernel

THE DEVELOPER'S CONFERENCE

- Microkernel
  - Kernel possui apenas o essencial
  - Serviços e drivers rodam no espaço de usuário
  - Vantagens: manutenção mais fácil, bugs não comprometem o sistema como um todo
  - Desvantagens: perda de velocidade e eficiência

Globalcode

Arquitetura de kernel

THE DEVELOPER'S CONFERENCE

Sistemas operacionais embarcados GERALMENTE usam microkernel

Globalcode

Sistemas de tempo real

THE DEVELOPER'S CONFERENCE

- Sistemas onde tempo é uma restrição
- O mais importante não é velocidade, mas determinismo
  - O sistema deve garantir que as tarefas são executadas precisamente quando elas devem
- Alguns sistemas possuem restrições mais rigorosas do que outros

Globalcode

Sistemas de tempo real

THE DEVELOPER'S CONFERENCE

- *Soft real time*
  - Permite pequenos atrasos e falhas
  - Exs: videoconferência, sistemas multimídia etc
- *Hard real time*
  - Atrasos e falhas podem ser críticos
  - Exs: controle industrial, sistemas de emergência, monitoramento de vôos etc

Globalcode

Escalonamento de processos

THE DEVELOPER'S CONFERENCE

Voltando ao kernel...

Globalcode

Escalonamento de processos

THE DEVELOPER'S CONFERENCE

- As tarefas geralmente são chamadas de processos
- Processo é um programa em execução
- Processo inclui informações (program counter, stack pointer, variáveis etc).

Globalcode

Escalonamento de processos

THE DEVELOPER'S CONFERENCE

- A maioria dos computadores executa 1 instrução por vez
- Em vários sistemas é necessário executar diversas tarefas
- Kernel deve gerenciar acesso dos processos à CPU segundo algum critério

Globalcode

Escalonamento de processos

THE DEVELOPER'S CONFERENCE

- Ao encerrar (ou interromper) um processo para executar outro, realiza-se a troca de contexto

- As informações do processo atuais são salvas e as do novo processo são carregadas

Globalcode

Escalonamento de processos

THE DEVELOPER'S CONFERENCE

- Escalonadores podem ser cooperativos ou preemptivos
- Escalonamento cooperativo
  - Processo executa até pedir para parar ou entrar em espera
- Escalonamento preemptivo
  - O kernel pode interromper um processo para executar outro

Globalcode

Escalonamento de processos

THE DEVELOPER'S CONFERENCE

- Os critérios para a execução podem ser diversos: primeiro a chegar, tarefa mais curta etc.
- Alguns escalonadores implementam um sistema de prioridades
- As prioridades podem ser fixas ou dinâmicas

Globalcode

Escalonamento de processos

THE DEVELOPER'S CONFERENCE

- Escalonamento com prioridades fixas
  - As prioridades de cada processo são definidas inicialmente e mantidas
  - Comportamento mais previsível
- Escalonamento por prioridades dinâmicas
  - As prioridades são calculadas pelo escalonador e podem ser recalculadas e mudadas
  - Comum em sistemas que trabalham com tempo
  - Sob certas condições, imprevisível

Globalcode

Escalonamento de processos

THE DEVELOPER'S CONFERENCE

- Sistemas com prioridade são vulneráveis a *resource starvation*
- Processos de alta prioridade podem se repetir indefinidamente, impedindo a execução dos outros processos
- Possível solução: *time slice*

Globalcode

Escalonamento de processos

THE DEVELOPER'S CONFERENCE

- Alguns escalonadores limitam o tempo que um processo pode ser executado
- Atingido o tempo, ele é interrompido para que outro processo seja executado
- Exemplo clássico: Round Robin

Globalcode

Agenda

THE DEVELOPER'S CONFERENCE

- *Earliest Deadline First*
  - Introdução
  - Funcionamento
  - Performance

Globalcode

Earliest Deadline First

THE DEVELOPER'S CONFERENCE

- Escalonador com prioridades dinâmicas
- Geralmente utilizado em sistemas de tempo real
- É baseado em *deadlines* (prazos)

Globalcode

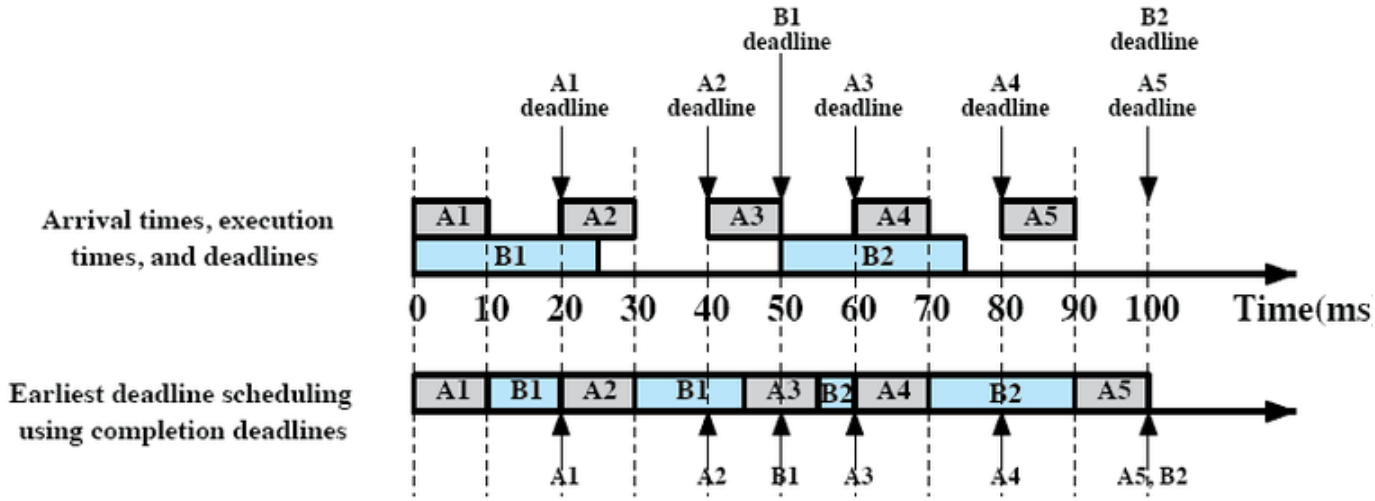
Earliest Deadline First

THE DEVELOPER'S CONFERENCE

- Cada processo recebe uma *deadline* para ser cumprida
- O algoritmo sempre busca o prazo cuja *deadline* está mais próxima
- Interrompe processos em execução caso processos com *deadlines* menores entrem no pool

Globalcode

EDF - Funcionamento



(*fonte: http://stackoverflow.com/questions/7619080/earliest-deadline-scheduling*)

EDF - Performance

THE DEVELOPER'S CONFERENCE

- Desempenho ótimo em sistemas preemptivos com 1 processador

- Se uma conjunto de processos pode ser escalonado garantindo o cumprimento das *deadlines*, o EDF consegue escaloná-lo

Globalcode

EDF - Performance

THE DEVELOPER'S CONFERENCE

- Limitação: o uso de CPU não deve passar de 100%

- Carga de CPU acima de 100% torna o algoritmo imprevisível e prazos serão perdidos

Globalcode

Agenda

THE DEVELOPER'S CONFERENCE

- Escalonamento no Linux
  - Classes de escalonadores
    - SCHED_FIFO
    - SCHED_RR
  - SCHED_DEADLINE
    - História
    - O algoritmo
    - Constant Bandwidth Server (CBS)
    - Vantagens

Globalcode

Escalonamento no Linux

THE DEVELOPER'S CONFERENCE

- Escalonadores no Linux são chamados de classes de escalonamento
- Na versão 3.14, existem 4 classes:
  - SCHED_OTHER: Completely Fair Scheduler
  - SCHED_FIFO: First-In-First-Out (tempo real)
  - SCHED_RR: Round-Robin (tempo real)
  - SCHED_DEADLINE: Earliest Deadline First (tempo real)

Escalonamento no Linux

THE DEVELOPER'S CONFERENCE

- SCHED_FIFO
  - Filas (FIFO) com prioridades
  - Processos executam até encerrarem ou entrarem em espera
  - Processos de prioridade alta podem interromper processos de prioridade baixa

Globalcode

Escalonamento no Linux

THE DEVELOPER'S CONFERENCE

- SCHED_RR
  - Semelhante ao SCHED_FIFO, com filas de prioridades
  - Utiliza o conceito de time slice do Round Robin clássico
  - Processos que estourem o tempo retornam ao fim da fila para sua prioridade
  - O time slice de um processo é “pausado” quando ele é interrompido por um processo com maior prioridade

Globalcode

SCHED_DEADLINE

THE DEVELOPER'S CONFERENCE

- História
  - Proposto em 2009 no 11th Real-Time Linux Workshop
  - Patch distribuído pela Linux Kernel Mailing List
  - Distribuído oficialmente no kernel a partir da versão 3.14 (março de 2014)

Globalcode

SCHED_DEADLINE

THE DEVELOPER'S CONFERENCE

- O algoritmo
  - Combinado com CBS (Constant Bandwidth Server)
  - Processos possuem 3 parâmetros: *runtime*, *period* e *deadline*
  - *Runtime* determina a duração máxima do processo
  - *Period* determina o intervalo para o processo se repetir, caso seja periódico
  - *Deadline* é o prazo máximo para o processo ser executado

Globalcode

SCHED_DEADLINE

THE DEVELOPER'S CONFERENCE

- Constant Bandwidth Server
  - Evita que processos usem mais tempo de CPU do que o previsto
  - *Runtime* da tarefa deve ser maior que o pior caso possível
  - Caso uma tarefa estoure o *Runtime* especificado, ela será interrompida e retorna quando atingir sua *deadline*

Globalcode

SCHED_DEADLINE

THE DEVELOPER'S CONFERENCE

- Vantagens:
  - Garantias de tempo: a tarefa receberá tempo de CPU independentemente do comportamento de outras tarefas
  - Limitação de tempo: Tarefas prioritárias não monopolizarão a CPU, tampouco bugs em uma tarefa prejudicarão outras

Globalcode

Agenda

THE DEVELOPER'S CONFERENCE

- Escalonador RT de baixo custo
  - Motivação
  - Objetivos
  - Hardware utilizado
  - Aplicação
  - O escalonador
  - Testes
  - Resultados
  - Considerações finais

Globalcode

Escalonador RT de baixo custo

THE DEVELOPER'S CONFERENCE

- Motivação:
  - Muitos sistemas de tempo real possuem recursos computacionais limitados
  - Sistemas como Linux são inviáveis em certas plataformas
  - Utilização de algoritmos complexos pode interferir no desempenho

Globalcode

Escalonador RT de baixo custo

THE DEVELOPER'S CONFERENCE

- Objetivos:
  - Garantir *hard real time* no sistema
  - Rodar em um sistema de baixo custo
    - Processador de 8 ou 16 bits, poucos KB de memória

Globalcode

Escalonador RT de baixo custo

THE DEVELOPER'S CONFERENCE

- Objetivos:
  - Garantir *hard real time* no sistema
  - Rodar em um sistema de baixo custo
    - Processador de 8 ou 16 bits, poucos KB de memória

Globalcode

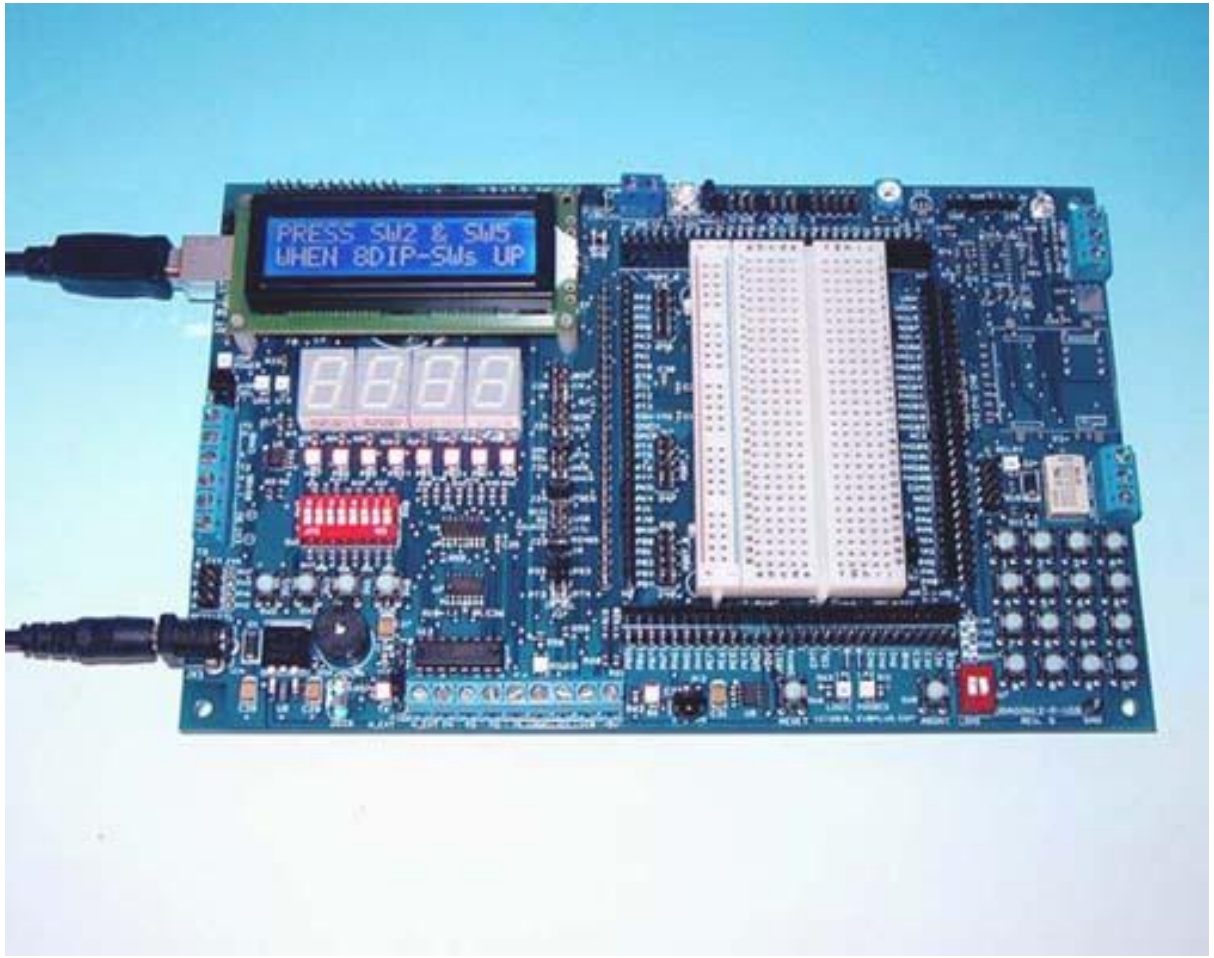
Escalonador RT de baixo custo

THE DEVELOPER'S CONFERENCE

- Hardware utilizado:
  - Wytec Dragon12-Plus-USB
    - MCU Freescale HCS12
    - 256KB RAM
    - Conversores A/D e D/A
    - Display LCD 16x2
    - USB/FT232

Globalcode

Escalonador RT de baixo custo



Escalonador RT de baixo custo

THE DEVELOPER'S CONFERENCE

- Aplicação desenvolvida:
  - Controlador PID digital
    - Utilização dos conversores A/D e D/A
    - Comunicação segura com desktop via USB (com protocolo próprio e CRC)
    - Capacidade de alterar seus parâmetros a pedido do usuário
    - Monitoramento gráfico dos componentes do PID

Globalcode

Escalonador RT de baixo custo

THE DEVELOPER'S CONFERENCE

- Aplicação desenvolvida:
  - Controlador PID digital
    - Utilização dos conversores A/D e D/A
    - Comunicação segura com desktop via USB (com protocolo próprio e CRC)
    - Capacidade de alterar seus parâmetros a pedido do usuário
    - Monitoramento gráfico dos componentes do PID

OBS: Controladores PID exigem leituras em intervalos constantes de tempo

Globalcode

Escalonador RT de baixo custo

THE DEVELOPER'S CONFERENCE

- Estrutura de um processo

```
typedef struct
{
        volatile unsigned int StackPoint;
        volatile unsigned int StackInit;
        volatile processState Status;
        volatile signed int Time;
        volatile priorityMode Prio;
        volatile procFunc Function;
}process;
```

Globalcode

Escalonador RT de baixo custo

THE DEVELOPER'S CONFERENCE

- Estrutura de um processo

```
typedef struct
{
        volatile unsigned int StackPoint;
        volatile unsigned int StackInit;
        volatile processState Status;
        volatile signed int Time;
        volatile priorityMode Prio;
        volatile procFunc Function;
}process;
```

Globalcode

Escalonador RT de baixo custo

THE DEVELOPER'S CONFERENCE

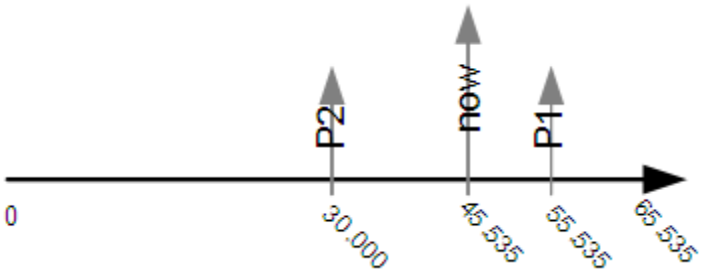
- O escalonador
  - Contagem regressiva para as *deadlines*
  - A cada *tick* do relógio de tempo real, todas as *deadlines* são decrementadas
    - Facilidade para detectar atrasos

Globalcode

Escalonador RT de baixo custo

THE DEVELOPER'S CONFERENCE

- P2 está atrasado ou foi agendado para o futuro?
- Houve overflow no timer?



Globalcode

Escalonador RT de baixo custo

THE DEVELOPER'S CONFERENCE

- O escalonador
  - Contagem regressiva para as *deadlines*
  - A cada *tick* do relógio de tempo real, todas as *deadlines* são decrementadas
    - Facilidade para detectar atrasos
    - Tempo = 0 => o processo atingiu sua *deadline*
    - Tempo negativo => o processo está atrasado

Globalcode

Escalonador RT de baixo custo

THE DEVELOPER'S CONFERENCE

- O escalonador
  - Pool de processos = vetor estático
    - Estruturas dinâmicas não são seguras
    - Ordenação seria caro
    - Algoritmo do tipo CBS seria caro
    - Sistema simples de prioridade
    - Processo “RT” é *hard real time*, o restante é *soft real time*

Globalcode

Escalonador RT de baixo custo

THE DEVELOPER'S CONFERENCE

```
for (i = 0; i < lastTask; i++){
    if((pool[i].Prio == RTOS) &&
       (pool[i].Status == READY)) {

      return i;
    }
  }
```

Globalcode

Escalonador RT de baixo custo

THE DEVELOPER'S CONFERENCE

```
for(i = 0; i<lastTask; i++){
    if (pool[i].Status == READY) {
      break;
    }
}
next = i;

for (i = (next+1); i < lastTask; i++){
  if((pool[i].Status == READY) &&
      (pool[i].Time < pool[next].Time)) {
     next = i;
  }
}
```

Globalcode

Escalonador RT de baixo custo

THE DEVELOPER'S CONFERENCE

- Testes
  - Para comparação, foi utilizado o Round Robin
  - A cada *tick* do relógio, busca o próximo processo pronto para ser executado
  - Foi mantida a *deadline* no Round Robin para verificar se há atrasos

Globalcode

Escalonador RT de baixo custo

THE DEVELOPER'S CONFERENCE

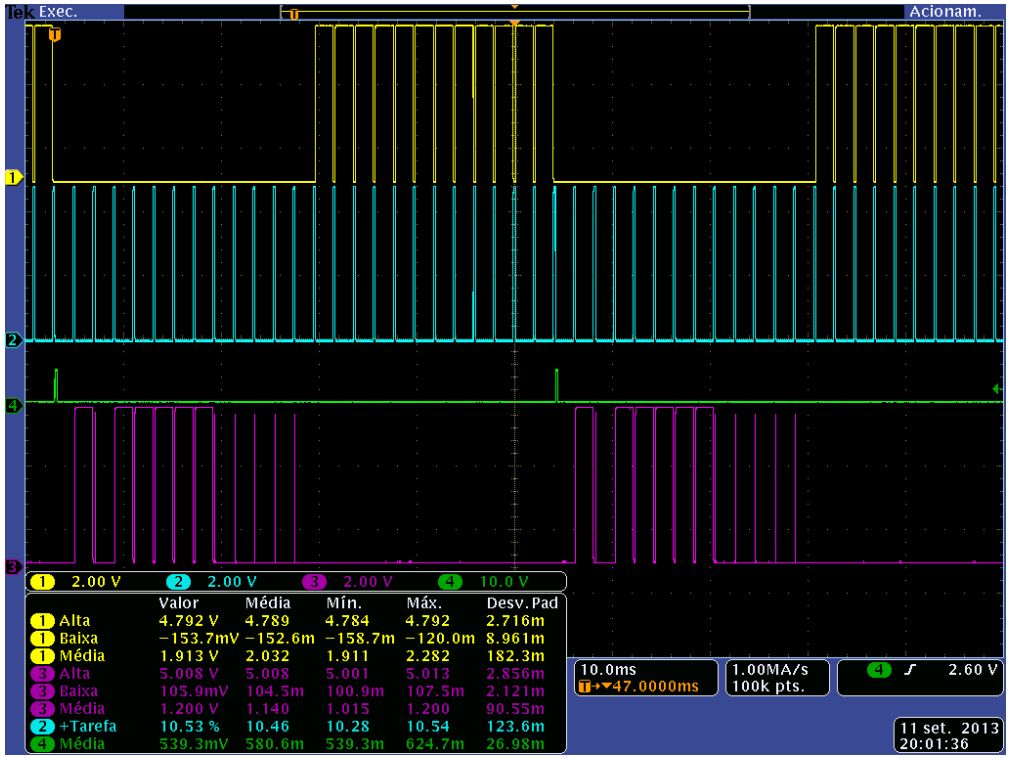
- Testes
  - Utilizou-se um osciloscópio de 4 canais para os testes
  - 4 portas de saída foram monitoradas pelo osciloscópio
  - Cada porta foi acionada em um evento específico:
    1. Execução de um processo comum
    2. Execução do processo prioritário
    3. "Execução" do processo *idle*
    4. Troca de contexto

Globalcode

Escalonador RT de baixo custo

THE DEVELOPER'S CONFERENCE

- Testes



Globalcode

Escalonador RT de baixo custo

THE DEVELOPER'S CONFERENCE

- Testes
  - Os processos “comuns” criados para teste consomem cerca de 4,68% do tempo de processamento
  - O processo idle é um processo “vazio”, ativo quando a CPU está ociosa
  - O processo prioritário foi agendado para ser executado a cada 25 trocas de contexto
  - Cada algoritmo foi testado com e sem prioridade com quantidade variável de processos (1 a 25)

Globalcode

Escalonador RT de baixo custo

THE DEVELOPER'S CONFERENCE

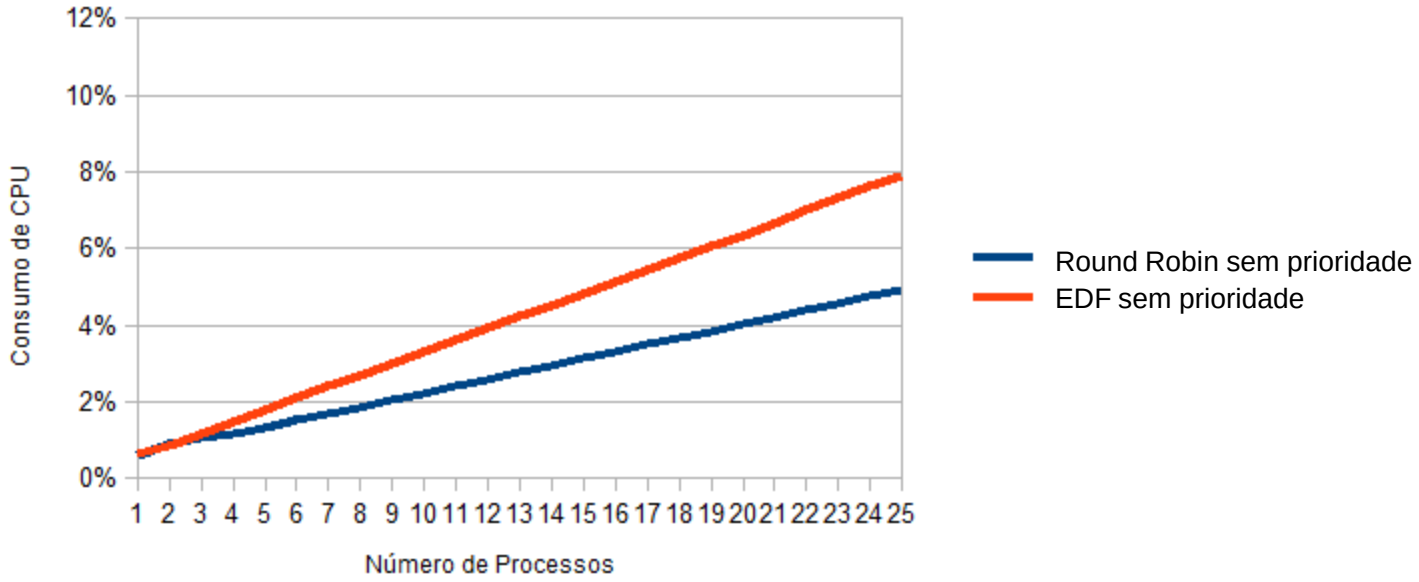
- Testes
  - Fazendo a medição no osciloscópio:
    - Obtem-se o valor máximo de tensão (em torno de 5V)
    - Obtem-se o valor médio de um determinado período
    - Dividindo-se o médio pelo máximo tem-se a proporção de tempo que a porta estava em nível lógico 1 no período

Globalcode

Escalonador RT de baixo custo

THE DEVELOPER'S CONFERENCE

- Resultados

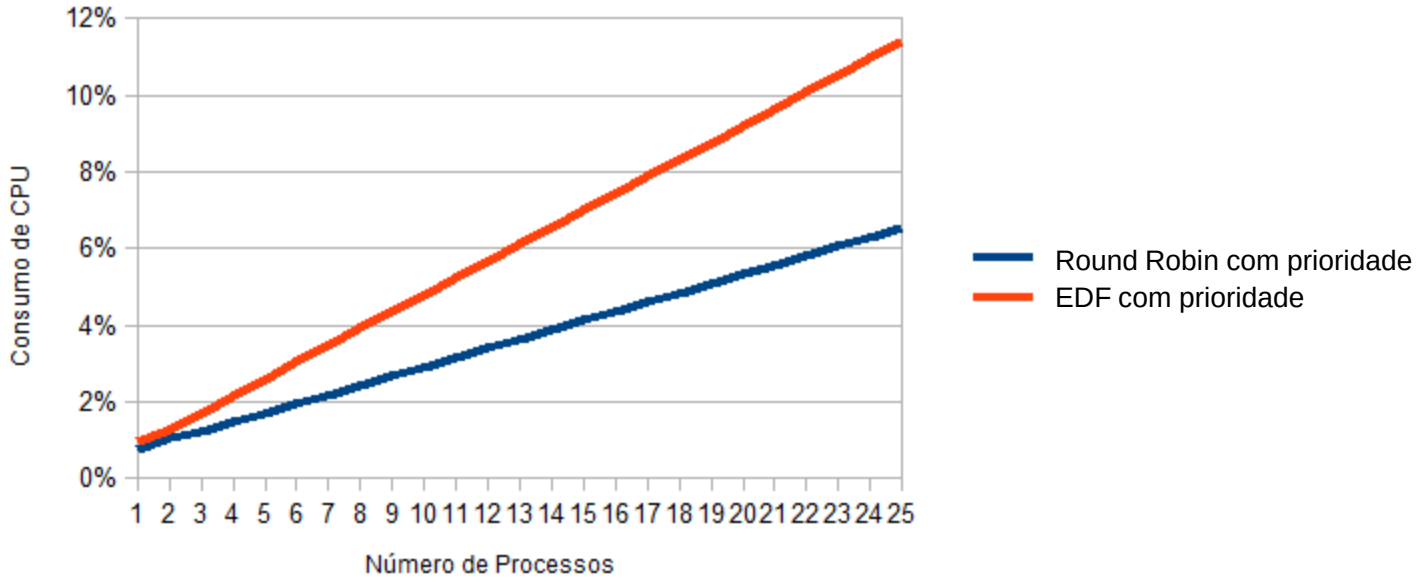


Globalcode

Escalonador RT de baixo custo

THE DEVELOPER'S CONFERENCE

- Resultados



Globalcode

Escalonador RT de baixo custo

THE DEVELOPER'S CONFERENCE

- Resultados

Falhas sem o sistema de prioridade

Escalonador	100% de Consumo	< 100% de Consumo
EDF	62.5%	0%
RR	75%	8.3%

Globalcode

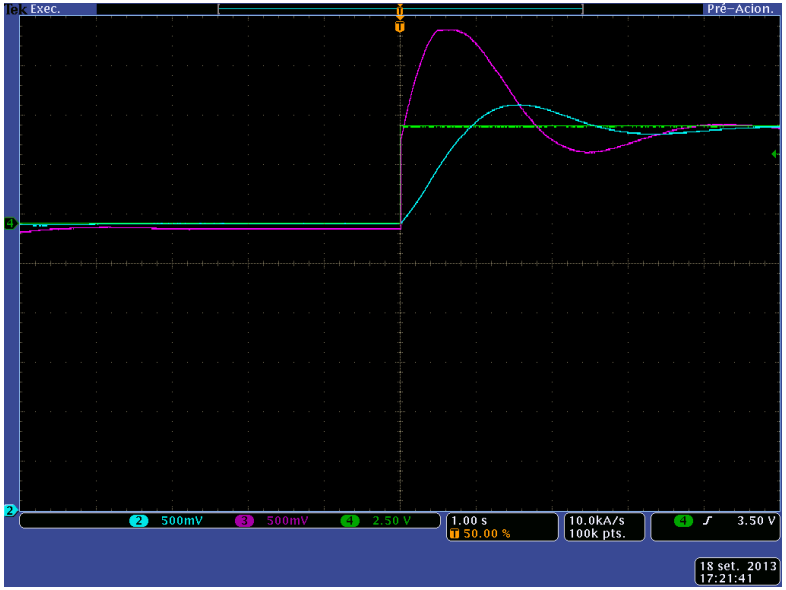
Escalonador RT de baixo custo

THE DEVELOPER'S CONFERENCE

- Resultados

Controle PID

Osciloscópio x Aplicativo



Globalcode

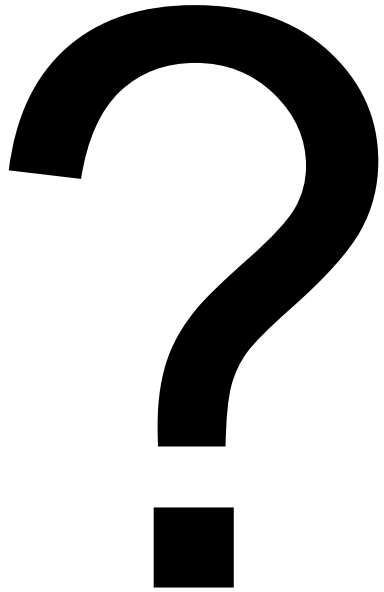
Considerações Finais

THE DEVELOPER'S CONFERENCE

- O EDF não falha quando bem utilizado ☺
- O EDF facilita o trabalho em sistemas com restrições temporais ☺
- O desenvolvedor deve ser cuidadoso para cumprir os requisitos do EDF ☹
- O *overhead* de processamento é significativo e aumenta ao se tratar falhas e erros ☹

Globalcode

Dúvidas



THE DEVELOPER'S CONFERENCE

Globalcode

Contato

THE DEVELOPER'S CONFERENCE

- Twitter: @rafaelmmoreira
- E-mail: rafaelmmoreira@gmail.com

Globalcode

Referências e recomendações

THE DEVELOPER'S CONFERENCE

- Evidence SRL. SCHED_DEADLINE. Março 2014. http://www.evidence.eu.com/sched_deadline.html
- THANG, L. T. Comparing real-time scheduling on the Linux kernel and an RTOS. *Embedded.com.* Abril 2012. http://www.embedded.com/design/operating-systems/4371651/Comparing-the-real-time-scheduling-policies-of-the-Linux-kernel-and-an-RTOS-

Globalcode

Referências e recomendações

THE DEVELOPER'S CONFERENCE

- SOUSA, P. B.; FERREIRA, L. L. Implementing a new real-time scheduling policy for Linux. *Embedded.com.* Julho 2010. http://www.embedded.com/design/operating-systems/4204929/Real-Time-Linux-Scheduling-Part-1 http://www.embedded.com/design/operating-systems/4204971/Real-Time-Linux-Scheduling-Part-2 http://www.embedded.com/design/operating-systems/4204980/Real-Time-Linux-Scheduling-Part-3

Globalcode

Referências e recomendações

THE DEVELOPER'S CONFERENCE

- ROSTEDT, S. Intro to Real-Time Linux for Embedded Developers. *Linux.com.* Março 2013. Entrevista concedida a Libby Clark. http://www.linux.com/news/featured-blogs/200-libby-clark/710319-intro-to-real-time-linux-for-embedded-developers

Globalcode

Referências e recomendações

THE DEVELOPER'S CONFERENCE

- Documentação oficial SCHED_DEADLINE: https://git.kernel.org/cgit/linux/kernel/git/torvalds/linux.git/plain/Documentation/scheduler/sched-deadline.txt?id=712e5e34aef449ab680b35c0d9016f59b0a4494c
- E não poderia faltar a Wikipedia ;) http://en.wikipedia.org/wiki/Earliest_deadline_first_scheduling

  http://en.wikipedia.org/wiki/SCHED_DEADLINE

Globalcode