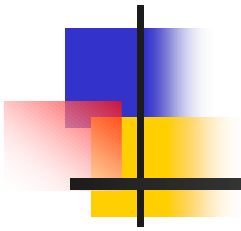


Sistemas Operacionais de Tempo-Real





Tópicos

- O que é computação de tempo-real?
- O que são sistemas operacionais de tempo-real?
- Onde são utilizados RTOSs?
- Sistemas:
 - MicroC/OS-II Kernel;
 - RTOS Linux: RTLinux, uCLinux+RTAI.
- Referências.



O que é computação de tempo-real?

- Computação não-tempo-real:
 - Um programa P é executado com uma entrada I , gerando uma saída O , executando em um tempo T :
 - $O = P(I), T$;
 - Quando menor T , maior performance;
 - Não há, entretanto, imposições quanto ao valor de T .



O que é computação de tempo-real?

- Computação de tempo-real:
 - É a computação “normal”, com uma imposição a mais em T:
 - $O = P(I), T < D;$
 - D é chamado de *deadline*: o programa obrigatoriamente deve ser executado antes do *deadline*;
 - O *deadline* é um tempo crítico de execução;



O que é computação de tempo-real?

- Computação de tempo-real:
 - O *deadline* de cada tarefa deve ser cumprido independente da carga do sistema;
 - O *deadline* varia de tarefa para tarefa:
 - Painel de cotações da bolsa: alguns minutos;
 - Freios automotivos: alguns milissegundos;
 - A execução do programa num tempo maior que D não é apenas indesejada, é inútil!



O que é computação de tempo-real?

- Num sistema em tempo-real, o tempo de execução das tarefas é crítico, não apenas um fator de performance;
- A “corretude” de um programa em tempo-real é baseada não apenas na lógica de sua execução, mas no tempo em que ele o faz;



O que é computação de tempo-real?

- *Soft real-time*: o tempo de execução é crítico, mas possíveis atrasos apenas degradam a qualidade da resposta do sistema:
 - Transmissão de áudio/vídeo *real-time*;
 - Sistemas de bancos de dados integrados (bancos, supermercados, etc.);
 - Sistemas de medição (satélites, microscópios, etc.).



O que é computação de tempo-real?

- *Hard real-time*: o tempo de execução é absolutamente crítico e não tolera atrasos; uma operação que não cumpra o *deadline* é considerada inútil:
 - Equipamentos médicos;
 - Freios automotivos e sistemas de navegação;
 - Programas que jogam xadrez!



O que é computação de tempo-real?

- Em *soft real-time*, o sistema costuma garantir um tempo de execução apenas probabilístico (i.e., “a maioria das tarefas” é executada no prazo);
- Em *hard real-time*, o sistema deve garantir um tempo de execução determinístico (i.e., todas as tarefas são provadas de executar no prazo).



O que são sistemas operacionais de tempo-real?

- Sistemas operacionais de tempo-real (RTOS) são sistemas que auxiliam (mas não garantem) um escalonamento de tarefas que cumpre com *deadlines*;
- RTOSs não são orientados à excelência de performance (*throughput*), mas sim ao cumprimento dos *deadlines*;



O que são sistemas operacionais de tempo-real?

- O elemento chave que diferencia um RTOS de um OS convencional é o escalonador;
- Um exemplo simples: um processador executando uma tarefa sem I/O e sem interrupções:
 - Não há interferência nenhuma na execução do programa: *hard-wired real-time!*



O que são sistemas operacionais de tempo-real?

- Há diversos modelos para um escalonador *real-time*:
 - Orientado a eventos (preemptivo);
 - Orientado a fatias de tempo;
 - Escalonamento separado para tarefas de tempo real e tarefas convencionais;
 - Diminuir o tempo de troca de contexto (*barrel processors*);



O que são sistemas operacionais de tempo-real?

- Mais sobre escalonadores:
 - Recursos em hardware (clocks externos, mascaramentos de interrupção) são geralmente utilizados;
 - Tarefas são divididas em “pedaços” (com uma fração do *deadline*) para facilitar na priorização de *threads*;
 - Estruturas de dados **simples!**



O que são sistemas operacionais de tempo-real?

- Comunicação entre tarefas: mutexes, semáforos, mensagens:
 - Mutex e semáforos: de máxima eficiência, não pode haver deadlock!
 - Geralmente associados à prioridade da tarefa;
 - Filas de mensagens: implementação mais segura, mas mais custosa;
 - Regiões críticas: geralmente executadas em nível de "kernel".



O que são sistemas operacionais de tempo-real?

- Alocação de recursos:
 - Similar a OSs convencionais, com priorização de tarefas;
 - No contexto de acesso exclusivo, utiliza mutexes e semáforos;
 - Quando acessado por interrupções, a prioridade da execução é mínima.



O que são sistemas operacionais de tempo-real?

- Alocação de memória:
 - Ainda mais crítica que em sistemas convencionais;
 - Não tolera fragmentação (RTOSs podem rodar anos a fio sem reiniciar!);
 - Não tolera atrasos (nada de buscar nós em listas ligadas!);
 - Uso de blocos fixos é bastante comum.



Onde são utilizados RTOSs?

- Operações em tempo-real são utilizadas em vários âmbitos;
- Há software desenvolvido para várias arquiteturas;
- Há várias arquiteturas desenvolvidas para vários usos;
- Há vários usos!



Onde são utilizados RTOSs?

- Coisas simples: jogos eletrônicos, videoconferências, sistemas integrados de informação;
- Coisas sérias: freios automotivos, airbags, sistemas de navegação em aviões, equipamentos médicos;
- Coisas **MUITO** sérias: usinas nucleares, satélites, programas que jogam xadrez!



Onde são utilizados RTOSs?

- No seu computador, a BIOS possui interrupções que checam a temperatura do processador: resposta em tempo crítico;
- Na sua câmera digital, há circuito que codifica áudio/vídeo com um mínimo de latência: é tempo-real?



MicroC/OS-II Kernel

- Sistema operacional *open-source*, usado largamente em sistemas embarcados;
- Segunda versão, primeira versão de 1992;
- Mantido pela Micrium Inc.;
- Escalonador preemptivo, escrito totalmente em ANSI C.



MicroC/OS-II Kernel

- Pode ser compilado para várias arquiteturas:
 - Altera Nios II
 - ARM
 - Atmel
 - Freescale/Motorola
 - PowerPC
 - Intel x86, etc.



MicroC/OS-II Kernel

- Chamadas não seguem o padrão POSIX:
 - OSMutexPost(), OSQAccept(), OSTaskCreate();
- Já possui várias bibliotecas desenvolvidas:
 - Compressão de dados (zip, bz2);
 - Pilhas de protocolos (TCP/IP).



RTOS Linux

- Basicamente duas versões de Linux implementando RTOS:
 - RTLinux (*Real-time Linux*);
 - RTAI (*Real-time Application Interface*): associado a um kernel como o uCLinux;
- Ambas são extensões do *kernel* possibilitando escalonamento *hard real-time* de tarefas;



RTOS Linux

- RTLinux: desenvolvido nos EUA, possui uma versão gratuita (RTLinux Free) e uma versão paga (RTLinux Pro);
- Um *kernel* “paralelo” (*real-time core*) é utilizado para gerenciar tarefas de tempo-real;
- É patenteado (não é exatamente “livre”).



RTOS Linux

- RTAI: *patch* para o *kernel* que provê interface de tempo-real para tarefas;
- Determinístico, preemptivo, POSIX-*compliant*;
- Pode ser compilado como um módulo para o *kernel*;
- Utiliza ADEOS (*Adaptive Domain Environment for OS*) como HAL;



RTOS Linux

- ADEOS é um *nanokernel*: providencia uma microinterface entre um (ou mais) *kernels* e o hardware subjacente;
- A RTAI usa a ADEOS para prover serviços de “alto nível” para gerência de tarefas de tempo-real:
 - O uso de HALs para “adaptar” um *kernel* normal em um *kernel real-time* é comum!



RTOS Linux

- uCLinux: *kernel/linux* para sistemas embarcados;
- uCLinux+RTAI = RTOS:
 - Muito usado como alternativa ao RTLinux e ao MicroC/OS-II;
 - Permite tarefas de tempo-real juntamente com tarefas convencionais.



Outros RTOS

- Xenomai (Linux, semelhante à RTAI);
- MaRTE OS (*Minimal Real-Time Operating System for Embedded Apps*);
- Helium: HC/CPUS08 *Core* (Freescale), menos de 900 bytes!
- PhoenixRTOS, RTEMS, RTMOS, FreeRTOS, pSOS, Fusion RTOS, Nucleus RTOS (Mentor Graphics),



Referências

- Labrosse, J. J. "MicroC/OS-II Kernel", Segunda Edição, CMP Books, 2002;
- Barr, M. "Multitasking Alternatives and the Perils of Preemption", Embedded Systems Design, 2006;
- Labrosse, J.; Barr, M. "Introduction to Preemptive Multitasking", Embedded Systems Programming, 2003;
- Wikipedia, termo "Real-time Computing", acesso em 2007;



Referências

- Sites interessantes:

- <http://www.netrino.com/Articles/RTOSAlternatives/index.php>
- <http://www.netrino.com/Articles/RTOSes/index.php>
- Por que **não** usar um RTOS:
<http://www.embedded.com/2000/0009/0009feat4.htm>



Lista de Fabricantes

- Express Logic (ThreadX);
- Red Hat (eCos);
- LynuxWorks (LynxOS);
- Micrium Inc. (MicroC/OS-II);
- Mentor Graphics (Nucleus);
- Wind River (pSOS, RTLinux, VxWorks);
- ...e GNU ☺ (RTAI, Xenomai, etc.).