

Ponto 2.1: Time-Triggered Systems e Event-Triggered systems

Sistemas de tempo real são aqueles onde a pontualidade da saída é tão importante quanto a sua exatidão. Um sistema de tempo real que não consegue responder no tempo especificado é considerado um sistema falho. Esse falha pode variar de catastrófico, para sistemas de tempo real duros (hard), a inconvenientes que devem ser tratados, para sistemas macis (soft).

Para lidar com esses tipos de falhas, geralmente utilizam-se dois tipos de arquitetura: time-triggered e event-triggered systems. A primeira se baseia na observação periódica dos estados do sistema, em intervalos de tempo fixos. O sistema então processará e responderá dentro destes intervalos de tempo. A segunda arquitetura se baseia na detecção de eventos significativos e a resposta quando eles ocorrem. Eventos significativos são aqueles que foram considerados relevantes no projeto do sistema, enquanto outros tipos de eventos quaisquer não causarão uma resposta.

Apelidando-se de event-triggered systems, que são referidos como ETS daqui para frente, o mecanismo frequentemente utilizado para a realização do ocorrido de um evento ao controlador, é o mecanismo de interrupção. Cada evento significativo gera uma interrupção, pausando o fluxo normal do programa e executando a rotina de resposta ao evento que ocorreu.

Eventos podem ser previstos (tipo P) ou eventos de acaso, chance em inglês (tipo C). Eventos do tipo P podem ser previstos através da utilização de alguma lei conhecida, por exemplo, se se passar uma corrente por um fio, este esquentará e recursos do sistema podem ser reservados previamente para lidar com este eventualidade. Eventos do tipo C, por outro lado,

Não podem ser previstos, como por exemplo o movimento rotativo de alguma molécula. Casos de não podem ser previstos, muitas vezes pode-se calcular as probabilidades de ocorrência destes eventos e dimensionar o sistema a partir delas. Por último, podem ocorrer grandes eventos, como o rompimento de um tanque no sistema de resfriamento de um processo crítico, que causam mudanças em diversas partes do processo, gerando algo chamado de "event showers", ou chuva de eventos. O sistema deve ser capaz de lidar com a informação de vários eventos ao mesmo tempo, lidando primeiro com os de prioridade maior alta e armazenando a ocorrência dos de prioridade mais baixa para lidar com eles depois.

Portanto, em ETS o fluxo de controle no sistema deve ser explícito, com buffers entre o controlador e o sistema de detecção, para que as informações de eventos simultâneos não sejam perdidas. Em relação ao escalonamento (scheduling) de tarefas, este é dinâmico, pois depende das demandas deles do sistema. De existir um escalonamento que atende a todas as demandas possíveis é um problema complexo (NP-Hard) e, mesmo se existir, o escalonamento em si também é um problema complexo. No prática, o que ocorre geralmente, é um uso de prioridades que tentam atender às requisições do sistema. Por isso, geralmente não há grandes mudanças de prioridades.

Assim, apesar de ETS serem relativamente simples de montar, elas podem ter extensas opções gerando que alterarão as regras de tempo. Isso faz com que a escalabilidade desses sistemas seja ~~complexa~~ complexa, pois qualquer alteração requer que todo o sistema seja testado novamente.

Time-triggered systems (TTS), são sistemas que monitoram os estados em tempos fixos e executam as tarefas necessárias para responder a essas condições.

Estes sistemas devem ter escalonamentos planejados, para garantir que suas especificações cumpram os requisitos do processo. Entre esses requisitos podem ser citados uma frequência alta e suficiente para detectar mudanças relevantes nos estados, assim como um tempo de ciclo grande a suficiente para a execução das tarefas necessárias.

Este escalonamento planejado faz com que TTS tenham implementações mais complexas, mas por outro lado, ~~é mais~~ ~~consistente~~ torna o problema de escalonamento muito mais simples. Além disso, estes sistemas não precisam ter negócios tão extensamente quebrados os ETS, e que torna sua escalaabilidade também muito simples.

Na maioria das ~~aplicações~~ explique as vantagens e desvantagens

TTS →

Vantagens

- Mais precisas
- Boa escalabilidade
- Escalonamento de tarefas mais simples
- Não uso de recursos quando a carga do sistema é alta

Desvantagens

- Implementação mais complexa
- Uso muitos de recursos quando a carga do sistema é baixa ou média

PP DE Aplicações

ETS →

- ~~Processamento de Transações (PTT)~~ → Subunidade de processamento de transações
- Vantagens → facilita a utilização, pode ser integrado com outros sistemas
- Implementação mais simples
- Bom uso de recursos quando a carga do sistema é baixa ou média
- Desvantagens →
- menor personalizado
- Portabilidade tem menor garantia
- Escalabilidade ruim
- Escalonamento difícil
- Uso de recursos provavelmente quando a carga aumenta

Obs: o fluxo de controlo em TTS é ~~simples~~ → ~~simples~~

→ diferente de ETS





UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO

PROTOCOLO DE ASSINATURA



O documento acima foi assinado digitalmente com senha eletrônica através do Protocolo Web, conforme Portaria UFES nº 1.269 de 30/08/2018, por
MARIA DA CONCEICAO CARNEIRO DA SILVA - SIAPE 1996445
Departamento de Engenharia Elétrica - DEE/CT
Em 05/05/2025 às 17:15

Para verificar as assinaturas e visualizar o documento original acesse o link: <https://api-lepisma.prod.uks.ufes.br/arquivos-assinados/1122228?tipoArquivo=O>