Conceitos de Confiabilidade

Sumário

- Motivação
- Conceitos introdutórios
- Modelo de sistema
- Impedimentos à confiabilidade
- Realização de sistemas confiáveis
- Conclusões

Motivação
(1/2)

- NENHUM SISTEMA É 100% CORRECTO:
  - erros na especificação, desenho e realização
  - *"time-to-market"* cada vez mais importante (e curto !)
  - imprevisibilidades internas e externas ao sistema
- Nas empresas:
  - dependência de sistemas de informação críticos
  - sistemas são factor de concorrência
  - falhas nos sistemas podem parar o negócio (na aviação podem resultar em perda de vidas humanas !)

Motivação
(2/2)

- O que é importante nos sistemas:
  - o reconhecimento de que podem falhar
  - a compreensão das causas de falha
  - diminuição do impacto de falhas (interno e externo)
  - a comprovação do seu bom funcionamento
- Abordagem sistemática destes problemas é urgente
- Necessidade de acordo na terminologia a utilizar
- Base comum para estudo e discussão

Conceitos introdutórios

(1/3)

- **Confiabilidade** é a qualidade do sistema que nos permite confiar, justificadamente, no serviço oferecido.
- **Confiabilidade** é um conceito global, que se decompõe em vários vectores quantificáveis:
  - fiabilidade (*reliability*)
  - disponibilidade (*availability*)
  - reparabilidade (*maintainability*)
  - segurança contra acidentes *(safety)*
  - segurança contra acesso não autorizado (*security)*

Conceitos introdutórios

(2/3)

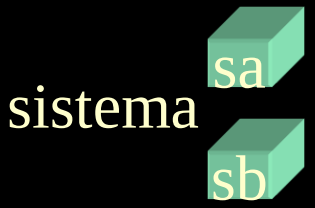
- **fiabilidade** (*reliability*):
  - medida do tempo de funcionamento de um sistema até falhar, ou da probabilidade de não falhar durante o tempo de missão (ex.:MTTF, MTBF, 10^{-5} falhas/hora, 99.9%)
- **disponibilidade** (*availability*):
  - medida do tempo (ou %) em que o sistema está operacional (ex.: MTBF/(MTBF+MTTR), 5000h/ano)
- **reparabilidade** (*maintainability*):
  - medida do tempo de reposição em serviço do sistema (ex.: MTTR)

Conceitos introdutórios
(3/3)

- **segurança contra acidentes durante funcionamento** (*safety*):
  - medida da fiabilidade do sistema relativa a faltas que ocasionem efeitos catastróficos
- **segurança contra acesso não autorizado** (*security*):
  - idem, relativo a faltas contra integridade, confidencialidade e autenticidade

Modelo de sistema

- Cada sistema:
  - obedece a uma especificação
  - oferece serviços ao exterior
  - possui um estado interno de execução
  - pode ser composto por subsistemas internos
  - pode utilizar os serviços de sistemas externos
  - modelo “Caixa Preta” (*“black box”)*



Impedimentos à confiabilidade

(1/4)

- São tudo aquilo que queríamos que não existisse **:-(**
- Os obstáculos são vários ("*the impairments*"):
  - **faltas** (*faults*): são a raiz do mau funcionamento do sistema
    - podem ser vistas quanto à:
      - natureza: acidentais / intencionais
      - origem: fenomenais / no sistema / na criação (ex.: desenho)
      - persistência: permanentes / temporárias / intermitentes
    - exemplos:
      - circuito que se queimou por sobreaquecimento
      - posição de memória sempre com o valor '0'
      - pode resultar em **erro** do sistema !

Impedimentos à confiabilidade

(2/4)

- **erros** (*errors*): são estados inconsistentes em que o sistema foi colocado em resultado de faltas
  - exemplo:
    - escrita de ‘1’ seguida de leitura de ‘0’
    - pode resultar em **falha**
- **falhas** (*failures*): são manifestações para o exterior de erros internos no sistema (desvios do especificado). Podem ser vistas quanto a:
    - domínio: valor / temporal
    - percepção: consistente / inconsistente
    - consequência: benigna / catastrófica

Impedimentos à confiabilidade *(3/4)*

- **falhas** (cont.):
  - exemplo:
    - entrega de resultado indevido
    - quem conhece o “Blue Screen” ? ;-)
  - representa uma falta se não for reconhecida como falha

- Modelo de “cascata” de impedimentos:

Componente A		Componente B	
falta > erro > falha	>	falta > erro > falha	> ...

Impedimentos à confiabilidade

(4/4)

- Exemplo de falha no componente “A” causando uma falta no componente “B”:
  - “A” falha na leitura da temperatura do forno sobreaquecido, e retorna um valor muito baixo (“A” pode ser sensor avariado)
  - “B” lê a temperatura (a falta), verifica que está dentro dos limites (o erro), e não faz nada
  - “B” falhou porque não tomou as acções devidas para baixar a temperatura (falha catastrófica !)
  - entretanto o forno rebenta ! **:-(**

Realização de sistemas confiáveis

(1/5)

- Como desenvolver sistemas que funcionem como esperado ?
- Métodos de desenvolvimento de sistemas confiáveis ou como obter confiança no funcionamento (*"the means"*):
  - prevenção de faltas (*prevention*)
  - tolerância a faltas (*tolerance*)
  - supressão de faltas (*avoidance*)
  - previsão de faltas (*forecasting*)

Realização de sistemas confiáveis

(2/5)

- Como não introduzir faltas ou evitar a sua ocorrência ?

- Prevenção de faltas:
  - auto-teste (*self-check*), assinaturas
  - correcta reutilização de componentes fiáveis
  - especificação rigorosa
  - ambientes e linguagens apropriados
  - protecção de HW

Realização de sistemas confiáveis *(3/5)*

- E se um raio cair vindo do nada ? Funcionará depois ?

- Tolerância a faltas:
  - exemplos: replicação, votadores, transacções, assinaturas e *checkpoints*, temporização/repetição

  - técnicas de processamento de erros:
    - mascaramento de erros (*error masking*)
    - detecção/recuperação (*detection/recovery*)

Realização de sistemas confiáveis *(4/5)*

- E se os “pontos de falta” forem conhecidos ?

- Supressão de faltas:
  - depuração (*debug*)
  - simulação
  - validação

Realização de sistemas confiáveis

(5/5)

- E se tudo para trás tiver falhado, só nos resta prever !

- Previsão de faltas:
  - injecção de faltas em *software*
  - injecção de faltas em *hardware*

Conclusões

- Nenhum sistema é 100% correcto !
- O impossível acontece mesmo [*Murphy's Law*] !
- Sistemas críticos devem ser tolerantes a faltas !
- Confiabilidade deve ser pensada desde o início !
- Credibilidade é importante …
- … e talvez um dia o *Murphy* esteja errado **;-)**

PERGUNTAS ?