1

Notas: 1

De acordo com Avizienis e demais autores no artigo "Basic Concepts and Taxonomy of Dependable and Secure Computing", item2.1, assinale se as propriedades listadas abaixo para sistemas de comunicação e computação são fundamentais, secundárias (importantes) ou não citadas no artigo.

desempenho	propriedade fundamental
custo	propriedade fundamental
ubiquidade	não citada
conectividade	não citada
mobilidade	não citada
adaptabilidade	secundária
usabilidade	secundária
gerenciabilidade (manageability)	secundária
dependabilidade	propriedade fundamental
modernidade	não citada
funcionalidade	propriedade fundamental
segurança (security)	propriedade fundamental

Correto
Notas relativas a este envio: 1/1.

2

Notas: 1

Considerando os conceitos básicos apresentados no artigo do Avizienis, associe o conceito ao termo usado no artigo.

O que é esperado que o sistema faça e é descrito na especificação funcional.	função do sistema
Conjunto dos seguintes estados: computação, comunicação, informação armazenada, interconexão e condição física.	estado total
O que o sistema faz para executar o esperado e é descrito por uma sequência de estados.	comportamento do sistema
Composto por um conjunto de componentes interligados, onde cada componente é outro sistema.	estrutura do sistema

Correto
Notas relativas a este envio: 1/1.

3

Notas: 1

Um serviço fornecido por um sistema, no seu papel de provedor de serviço, é:

Escolher uma resposta.

- a. a especificação funcional do sistema ✗
- b. a interface do sistema percebida pelo usuário ou outro sistema ✗
- c. o comportamento do sistema conforme percebido pelo usuário ✓
- d. uma sequência de estados internos do sistema ✗
- e. uma execução correta do sistema ✗

Correto
Notas relativas a este envio: 1/1.

4

Notas: 1

Aqui traduzimos *fault* por falha, *error* por erro e *failure* por defeito. Alguns grupos no Brasil usam outras traduções. De acordo com o artigo do Avizienis e considerando a tradução que usamos, selecione os termos adequados para os espaços vazios:

Um serviço correto é fornecido quando o serviço implementa a função do sistema. Um defeito de serviço é um evento que ocorre quando o serviço fornecido desvia do serviço correto .

Correto
Notas relativas a este envio: 1/1.

5

Notas: 1

Assinale verdadeiro ou falso baseando-se nos conceitos básicos do artigo do Avizienis.

Dizemos que um serviço apresenta defeito apenas quando a especificação não descreve adequadamente a função do sistema.	falso
Um defeito de serviço é uma transição do serviço correto para o serviço incorreto.	verdadeiro
Dizemos que um serviço apresenta defeito apenas quando ele não segue sua especificação funcional.	falso
Dizemos que um serviço apresenta defeito quando ele não segue sua especificação funcional ou quando a especificação não descreve adequadamente a função do sistema.	verdadeiro

Correto
Notas relativas a este envio: 1/1.

6

Notas: 1

Um defeito de serviço significa que ao menos um estado externo do sistema desvia do estado correto do serviço. Esse desvio é chamado de erro .

Correto
Notas relativas a este envio: 1/1.

7

Notas: 1

A causa real ou suposta de um erro é chamada de falha .

Correto
Notas relativas a este envio: 1/1.

8

Notas: 1

Assinale verdadeiro ou falso

Uma falha geralmente primeiro causa um erro no estado de um componente, que é parte do estado interno do sistema, e o estado externo não é imediatamente afetado.	verdadeiro
Alguns erros nunca chegam a provocar um defeito.	verdadeiro
Toda falha causa um defeito de serviço.	falso
Um erro é parte do estado total de um sistema que pode conduzir a um defeito de serviço.	verdadeiro

Correto
Notas relativas a este envio: 1/1.

9

Notas: 1

Uma falha é ativa quando causa um erro; caso contrário é dormente .

Correto
Notas relativas a este envio: 1/1.

10

Notas: 1

A definição original de dependabilidade é:

Escolher uma resposta.

- a. habilidade de evitar falhas permanentes ✗
- b. habilidade de evitar erros no estado interno do sistema ✗
- c. habilidade do sistema de fornecer um serviço no qual se pode justificamente confiar ✓
- d. habilidade de evitar defeitos de serviço que sejam mais frequentes ou mais severos que o aceitável ✗
- e. habilidade do sistema de fornecer um serviço correto ✗

Correto
Notas relativas a este envio: 1/1.

11

Notas: 1

A definição alternativa de dependabilidade é:

Escolher uma

resposta.

- a. habilidade de evitar erros no estado interno do sistema ✗
- b. habilidade de evitar defeitos de serviço que sejam mais frequentes ou mais severos que o aceitável ✓
- c. habilidade do sistema de fornecer um serviço correto ✗
- d. habilidade do sistema de fornecer um serviço no qual se pode justificamente confiar ✗
- e. habilidade de evitar falhas permanentes ✗

Correto
Notas relativas a este envio: 1/1.

12
Notas: 1

Assinale o conceito mais próximo a disponibilidade:

Escolher uma resposta.

- a. Ausência de alterações impróprias no sistema. ✗
- b. Continuidade do serviço correto. ✗
- c. Prontidão para serviço correto. ✓
- d. Facilidade de executar modificações e reparos. ✗
- e. Ausência de consequências catastróficas para o usuário ou ambiente. ✗

Correto
Notas relativas a este envio: 1/1.

13
Notas: 1

Assinale o conceito mais próximo ao atributo conhecido por confiabilidade:

Escolher uma resposta.

- a. Ausência de alterações impróprias no sistema. ✗
- b. Continuidade do serviço correto. ✓
- c. Ausência de consequências catastróficas para o usuário ou ambiente. ✗
- d. Facilidade de executar modificações e reparos. ✗
- e. Prontidão para serviço correto. ✗

Correto
Notas relativas a este envio: 1/1.

14
Notas: 1

Assinale o conceito mais próximo ao atributo conhecido por segurança funcional (safety):

Escolher uma resposta.

- a. Ausência de alterações impróprias no sistema. ✗
- b. Prontidão para serviço correto. ✗
- c. Facilidade de executar modificações e reparos. ✗
- d. Continuidade do serviço correto. ✗
- e. Ausência de consequências catastróficas para o usuário ou ambiente. ✓

Correto
Notas relativas a este envio: 1/1.

15
Notas: 1

Assinale o conceito mais próximo ao atributo conhecido por integridade:

Escolher uma resposta.

- a. Continuidade do serviço correto. ✗
- b. Facilidade de executar modificações e reparos. ✗
- c. Ausência de consequências catastróficas para o usuário ou ambiente. ✗
- d. Ausência de alterações impróprias no sistema. ✓
- e. Prontidão para serviço correto. ✗

Correto
Notas relativas a este envio: 1/1.

16
Notas: 1

Dos atributos listados abaixos, indicar quais são atributos de dependabilidade, quais são atributos de segurança (*security*) e quais são atributos de ambos (dependabilidade e segurança):

disponibilidade	ambos
segurança funcional (safety)	dependabilidade
confidencialidade	segurança (security)
facilidade de manutenção	dependabilidade
integridade	ambos
confiabilidade	dependabilidade

17
Notas: 1

Correto
Notas relativas a este envio: 1/1.

Vários meios foram desenvolvidos para atender os atributos de dependabilidade e segurança computacional (security). Os autores classificam esses meios em quatro categorias principais. Assinale entre as opções abaixo, uma que não corresponda a uma das quatro categorias:

Escolher uma resposta.

- a. tolerância a falhas ✗
- b. detecção de falhas ✓
- c. prevenção de falhas ✗
- d. previsão de falhas ✗
- e. remoção de falhas ✗

Correto
Notas relativas a este envio: 1/1.

18
Notas: 1

O item 3.1 do artigo do Avizienis inicia falando do ciclo de vida de um sistema. Os autores distinguem duas fases. Associe a definição ao nome da fase:

Inclui todas as atividades desde a concepção inicial do sistema até o sistema estar testado e pronto para prover o serviço desejado.	fase de desenvolvimento
Inicia quando o sistema é aceito e começa a fornecer o serviço ao usuário.	fase de uso

Correto
Notas relativas a este envio: 1/1.

19
Notas: 1

Durante a fase de uso na vida de um sistema, o sistema alterna períodos de fornecimento correto de serviço, queda de serviço (*outage*) e desligamento (*shutdown*). Qual a diferença entre queda de serviço e desligamento?

Causado por defeito de serviço.	queda (outage)
Parada intencional e autorizada do serviço.	desligamento (shutdown)
Serviço incorreto ou mesmo nenhum serviço é fornecido.	queda (outage)

Correto
Notas relativas a este envio: 1/1.

20
Notas: 1

Qual o significado do termo manutenção no artigo?

Escolher uma resposta.

- a. apenas modificações na fase de uso do sistema ✗
- b. reparos e modificações na fase de uso do sistema ✓
- c. apenas modificações na fase de desenvolvimento ✗
- d. apenas troca de componentes de hardware ✗
- e. apenas reparo ✗

Correto
Notas relativas a este envio: 1/1.

21
Notas: 1

Responda com números decimais inteiros.

No item em que Avizienis e os demais autores tratam da taxonomia de falhas (3.2.1), os autores classificam as

falhas de acordo com 8 pontos de vista, também chamados de classes de falhas elementares. Se todas as classes pudessem ser combinadas, teríamos 256 diferentes classes de falhas combinadas. Entretanto, os autores identificaram apenas 31 combinações que fazem sentido, por enquanto.

22
Notas: 1

Correto
Notas relativas a este envio: 1/1.

Assinale SIM para as classes de falha elementares e NÃO para os termos que não correspondem as classes de falhas elementares sugeridas por Avizienis e demais autores.

limites	SIM
objetivo	SIM
capacidade	SIM
permanência	NÃO
errata	NÃO
interferência física	SIM
fase	SIM
dimensão	SIM
persistência	SIM
causa	NÃO
intenção	SIM
deterioração física	NÃO

Parcialmente correta
Notas relativas a este envio: 0.83/1.

23
Notas: 1

Todas as combinações razoáveis de classes de falhas foram agrupadas em 3 grupos principais (com intersecção parcial). Quais são esses grupos?

Escolher uma resposta.

- a. falhas internas, externas e correlacionadas ✗
- b. falhas de especificação, de desenvolvimento e de operação ✗
- c. falhas de desenvolvimento, físicas e de interação ✓
- d. falhas físicas, humanas e ambientais ✗
- e. falhas simples, múltiplas simultâneas e múltiplas em cascata ✗

Correto
Notas relativas a este envio: 1/1.

24
Notas: 1

Falhas internas e externas são agrupadas na classe elementar de falhas denominada:

Escolher uma resposta.

- a. persistência ✗
- b. intenção ✗
- c. dimensão ✗
- d. causa ✗
- e. limites ✓

Correto
Notas relativas a este envio: 1/1.

25
Notas: 1

Falhas permanentes e transientes são agrupadas na classe elementar de falhas denominada:

Escolher uma resposta.

- a. dimensão ✗
- b. fase ✗
- c. limites ✗
- d. persistência ✓
- e. causa ✗

Correto
Notas relativas a este envio: 1/1.

26
Notas: 1

Falhas de hardware e de software são agrupadas na classe elementar de falhas denominada:

Escolher uma resposta.

- a. dimensão ✓
- b. limites ✗
- c. persistência ✗
- d. fase ✗
- e. causa ✗

Correto
Notas relativas a este envio: 1/1.

27
Notas: 1

As falhas humanas, ou falhas resultantes da ação de pessoas, são distinguíveis pelo objetivo. Considerando o **objetivo** da interação humana com o sistema, quais são as duas classes básicas de falhas humanas?

Escolher uma resposta.

- a. acidental e incompetência ✗
- b. maliciosa e não-maliciosa ✓
- c. deliberada e não-deliberada ✗
- d. desenvolvimento e operacional ✗
- e. permanente e transiente ✗

Correto
Notas relativas a este envio: 1/1.

28
Notas: 1

As classes combinadas de falhas aparecem agrupadas em grupos parcialmente sobrepostos. Um desses grupos compreende as falhas de interação. Falhas de interação ocorrem durante a fase de uso , portanto elas são todas do tipo operacional . Elas são causadas por elementos do ambiente do sistema em uso e são, então, todas falhas externas . Um exemplo de falhas de interação que consiste em erro na inicializaçao de parâmetros do sistema são as falhas de configuração .

Correto
Notas relativas a este envio: 1/1.

Fechar esta janela Fechar esta janela

1

Notas: 1

Considere a situação em que um sistema foi projetado de acordo com sua especificação e se comporta rigorosamente de acordo com essa especificação. Mesmo assim acontece um desastre inaceitável do ponto de vista do usuário. De acordo com o item 3.3 do artigo do Avizienis, assinale verdadeiro ou falso para as sentenças que seguem:

O sistema apresenta defeito porque não está fornecendo o serviço esperado mesmo seguindo a sua especificação.	Verdadeiro
O sistema não apresenta defeito porque está de acordo com a especificação.	Falso
Um defeito devido a uma falha de especificação é um conceito muito vago e subjetivo e por isso não se deve perder tempo escrevendo especificações.	Falso
O sistema apresenta defeito e a especificação não está adequadamente descrevendo a função do sistema.	Verdadeiro
A especificação tem uma ou mais falhas.	Verdadeiro

Correto

Notas relativas a este envio: 1/1.

2

Notas: 1

Um defeito de serviço é um evento que ocorre quando o serviço fornecido desvia do serviço correto. As diferentes formas nas quais um sistema se desvia de um serviço correto são conhecidas como:

Escolher uma resposta.

- a. inconsistências do serviço ✗
- b. modos de defeito de serviço ✓
- c. erros de serviço ✗
- d. falhas de especificação ✗
- e. severidades de defeito de serviço ✗

Correto

Notas relativas a este envio: 1/1.

3

Notas: 1

De acordo com Avzienis e demais autores, falhas de especificação podem ser de dois tipos. Assinale os dois tipos na lista a seguir:

Escolha pelo menos uma resposta.

- a. ação (commision) ✓
- b. incorreção ✗
- c. inconsequência ✗
- d. inadequação ✗
- e. omissão ✓

Correto

Notas relativas a este envio: 1/1.

4

Notas: 1

Modos de defeito de serviço caracterizam serviço incorreto de acordo com 4 pontos de vista. Assinale o ponto de vista que **não** pertence aos 4 listados no artigo de Avizienis.

Escolher uma resposta.

- a. consistência do defeito ✗
- b. custo do defeito ✓
- c. consequência do defeito ✗
- d. domínio do defeito ✗
- e. detectabilidade do defeito ✗

Correto

Notas relativas a este envio: 1/1.

5

Notas: 1

Considerando o domínio do defeito, quais são os dois tipos de defeitos citados por Avizienis?

Escolher uma resposta.

- a. conteúdo e temporização ✓
- b. parada e errático ✗

c. sinalizado e não sinalizado ✗
d. silencioso e severo ✗
e. consistentes e bizantinas ✗

Correto
Notas relativas a este envio: 1/1.

6
Notas: 1

Considerando o domínio do defeito, quais são as duas classes de defeitos quando tanto a informação quanto a temporização são incorretas?

Escolher uma resposta.

a. defeitos consistentes e bizantinos ✗
b. defeitos silenciosos e severos ✗
c. adiantamento ou atraso ✗
d. defeitos de parada e errático ✓
e. defeitos sinalizados e não sinalizados ✗

Correto
Notas relativas a este envio: 1/1.

7
Notas: 1

Considerando o domínio de defeitos, associe a descrição a classe de defeito:

caso especial de defeito de parada onde nenhum serviço é fornecido, ou no caso de sistemas distribuídos, nenhuma mensagem é enviada	defeito silencioso
o serviço é fornecido mas é incoerente	defeito errático
o estado externo fica constante, ou seja, a atividade do sistema não é mais percebida pelo usuário	defeito de parada

Correto
Notas relativas a este envio: 1/1.

8
Notas: 1

Quais são os dois tipos de defeitos, considerando a detectabilidade do defeito?

Escolher uma resposta.

a. conteúdo e temporização ✗
b. silencioso e severo ✗
c. sinalizado e não sinalizado ✓
d. consistentes e bizantino ✗
e. parada e errático ✗

Correto
Notas relativas a este envio: 1/1.

9
Notas: 1

Considerando o ponto de vista da detectabilidade do defeito, os mecanismos de detecção apresentam dois modos de defeito. Associe as descrições aos termos apropriados:

Sinalização de perda de função quando nenhum defeito efetivamente ocorreu.	Defeito não sinalizado
Nenuma indicação de perda de função quando o defeito ocorre.	Falso alarme

Errado
Notas relativas a este envio: 0/1.

10
Notas: 1

Quais são os dois tipos de defeitos, considerando a consistência do defeito?

Escolher uma resposta.

a. silencioso e severo ✗
b. parada e errático ✗
c. sinalizado e não sinalizado ✗
d. consistentes e bizantinos ✓
e. conteúdo e temporização ✗

Correto

Notas relativas a este envio: 1/1.

11
Notas: 1

Alguns ou todos os usuários percebem o serviço incorreto de maneira diferente, alguns até podem perceber o serviço como correto. Esse tipo de defeito é chamado de:

Escolher uma resposta.

- a. consistente ✗
- b. silencioso ✗
- c. detectável ✗
- d. bizantino ✓
- e. errático ✗

Correto
Notas relativas a este envio: 1/1.

12
Notas: 1

Graduar [as consequências] de defeitos sobre o ambiente do sistema permite definir [a severidade] do defeito. Aos seus [níveis] estão associadas probabilidades máximas de ocorrência.

Correto
Notas relativas a este envio: 1/1.

13
Notas: 1

A quantidade, o nome e a definição dos níveis de severidade de defeitos, assim como os níveis aceitáveis de probabilidades de ocorrência, são relacionados [à aplicação] e envolvem os atributos de dependabilidade e segurança (security) para a aplicação considerada.

Correto
Notas relativas a este envio: 1/1.

14
Notas: 1

Considere os exemplos de critérios para determinação das classes de severidade de defeitos mencionados no artigo do Avizienis. Associe o critério ao atributo:

extensão da corrupção de dados e a capacidade de recuperar essa corrupção	integridade
possibilidade de perdas de vidas humanas	segurança (safety)
duração da interrupção do serviço	disponibilidade

Correto
Notas relativas a este envio: 1/1.

15
Notas: 1

Sistemas que são projetados e implementados de tal forma que apresentam defeitos apenas em modos específicos descritos na especificação de dependabilidade e segurança (*security*) e apenas em uma extensão aceitável são sistemas com controle de defeito (*fail-controlled systems*). Associe a descrição ao termo apropriado:

sistema onde os defeitos são todos (ou numa extensão aceitável) apenas defeitos da classe de parada	fail-halt ou fail-stop
sistema onde os defeitos são todos (ou numa extensão aceitável) apenas defeitos silenciosos	fail-silent
sistema onde os defeitos são todos (ou numa extensão aceitável) apenas defeitos menores (minor failures)	fail-safe

Correto
Notas relativas a este envio: 1/1.

16
Notas: 1

Falhas de desenvolvimento podem levar a defeitos parciais ou completos de desenvolvimento ou permanecer indetectáveis até a fase de uso do sistema. De acordo com Avizienis (item 3.3.2) existem dois aspectos relacionados a defeitos de desenvolvimento, são eles:

Escolher uma resposta.

- a. defeitos de orçamento e defeitos de escalonamento ✓
- b. especificação com erro e alterações na especificação ✗

- c. erros humanos e defeitos de produção ✗
- d. defeitos de ferramentas e interferências do ambiente ✗
- e. tecnologia obsoleta e defeitos de desempenho ✗

Correto
Notas relativas a este envio: 1/1.

17
Notas: 1

Avizienis e os demais autores citam várias causas para defeitos de desenvolvimento (item 3.3.2). Segundo eles, todas as causas citadas são geralmente devidas a:

Escolher uma resposta.

- a. complexidade do sistema a ser desenvolvido subestimada ✓
- b. ausência de especificaçao para o sistema ✗
- c. carência de formação acadêmica em tolerância a falhas ✗
- d. falta de gente para compor uma equipe de desenvolvedores ✗
- e. funcionalidade do sistema desconhecida a priori ✗

Correto
Notas relativas a este envio: 1/1.

18
Notas: 1

Assinale verdadeiro ou falso considerando o item 3.3.3 do artigo do Avizienis e demais autores.

Uma especificação de dependabilidade e segurança (security) é um contrato que estabelece os objetivos de cada atributo: confiabilidade, disponibilidade, safety, confidencialidade, integridade e facilidade de manutenção.	Verdadeiro
Uma classe de falha de especificação de dependabilidade é a escolha injustificada de exagerados níveis de demanda para um ou mais atributos de dependabilidade, o que eleva os custos e pode levar a defeito de desenvolvimento.	Verdadeiro
Não é necessária uma especificação de dependabilidade, os desenvolvedores devem garantir por contrato que falhas não ocorram.	Falso
Uma especificação de dependabilidade e segurança jamais contém falhas.	Falso
Um defeito de dependabilidade ou segurança ocorre quando o sistema sofre defeitos de serviço mais frequentes ou mais severos do que aceitável.	Verdadeiro

Correto
Notas relativas a este envio: 1/1.

19
Notas: 1

Um erro foi definido como parte do estado do sistema que pode conduzir a um defeito, que ocorre quando o erro causa o desvio do serviço correto. A causa de um erro recebe o nome de falha.

Correto
Notas relativas a este envio: 1/1.

20
Notas: 1

Escolha o termo apropriado:

quando o erro está presente mas não é sinalizado dizemos que o erro é	latente
quando a presença do erro é indicada por uma mensagem de erro ou por um sinal de erro dizemos que o erro é	detectado

Correto
Notas relativas a este envio: 1/1.

21
Notas: 1

Um erro sempre provoca um defeito de serviço.

Resposta:

- Verdadeiro ✗
- Falso ✓

Correto
Notas relativas a este envio: 1/1.

22
Notas: 1

Considerando erros, associe o termo ao conceito:

23
Notas: 1

são provocados por falhas que causam problemas simultâneos em mais do que um componente	erros múltiplos relacionados
afetam apenas um componente	erro simples

Correto
Notas relativas a este envio: 1/1.

Relacione o termo ao conceito.

aplicação de uma entrada a um componente que faz uma falha dormente se tornar ativa	ativação de falha
processo computacional que faz com que um erro seja sucessivamente transformado em outros erros	propagação de erro

Correto
Notas relativas a este envio: 1/1.

24
Notas: 1

A habilidade de identificar o padrão de ativação de uma falha que causa um ou mais erros, de acordo com Avizienis, é chamada de:

Escolher uma resposta.

- a. identificação do padrão de ativação ✗
- b. reprodutibilidade da ativação da falha ✓
- c. inevitabilidade de ativação de falha ✗
- d. probabilidade de ativar a falha ✗
- e. estímulo de ativação da falha ✗

Correto
Notas relativas a este envio: 1/1.

25
Notas: 1

Falhas podem ser categorizadas de acordo com sua reprodutibilidade de ativação. Falhas cuja ativação é reproduzível são chamadas de:

Escolher uma resposta.

- a. intermitentes ✗
- b. evasivas ou soft ✗
- c. sólidas ou hard ✓
- d. múltiplas ✗
- e. simples ✗

Correto
Notas relativas a este envio: 1/1.

26
Notas: 1

Falhas podem ser categorizadas de acordo com sua reprodutibilidade de ativação. Falhas cuja ativação não é sistematicamente reproduzível são chamadas de:

Escolher uma resposta.

- a. independentes ✗
- b. sólidas ou hard ✗
- c. evasivas ou soft ✓
- d. intermitentes ✗
- e. residuais ✗

Correto
Notas relativas a este envio: 1/1.

Fechar esta janela | Fechar esta janela

1

Notas: 1

De acordo com o artigo “Fighting Bugs: Remove, Retry, Replicate, and Rejuvenate” de Michael Grottke e Kishor S. Trivedi, publicado na IEEE Computer, em fevereiro de 2007, qual a definição de defeito de sistema (*system failure*)?

Escolher uma resposta.

- a. defeito de sistema ocorre quando o comportamento real do sistema leva a prejuízos financeiros ✗
- b. defeito de sistema ocorre quando o sistema trava ✗
- c. defeito de sistema ocorre quando o comportamento real do sistema se desvia do serviço correto ✓
- d. defeito de sistema ocorre quando o comportamento do sistema conduz a perdas de vidas ou danos irremediáveis ao ambiente ✗
- e. defeito de sistema ocorre quando o comportamento real do sistema se desvia da sua especificação ✗

Correto
Notas relativas a este envio: 1/1.

2

Notas: 1

Em que fase do ciclo vida do software o reparo de bugs é mais oneroso (ou seja, apresenta maior custo)?

Escolher uma resposta.

- a. fase de descarte ✗
- b. fase de aquisição ✗
- c. fase de desenvolvimento ✗
- d. fase operacional ✓
- e. fase de teste ✗

Correto
Notas relativas a este envio: 1/1.

3

Notas: 1

De acordo com Grottke e Tivedi, o que pode tornar uma tarefa muito difícil o diagnóstico e o isolamento das falhas que são responsáveis por um defeito observável ?

Escolher uma resposta.

- a. o fato dos testadores não dominarem estratégias de teste dinâmicas ✗
- b. o fato dos desenvolvedores documentarem mal o sistema ✗
- c. o fato de que alguns defeitos não podem ser reproduzidos ✓
- d. o desconhecimento da especificação do sistema por parte dos testadores ✗
- e. o fato dos testadores não dominarem estratégias de teste estáticas ✗

Correto
Notas relativas a este envio: 1/1.

4

Notas: 1

O que são Bohrbugs?

Escolher uma resposta.

- a. falhas que provocam defeitos que são difíceis de serem reproduzidos ✗
- b. falhas que de acordo com Niels Bohr afetam a estrutura dos átomos ✗
- c. falhas que os métodos tradicionais de teste não conseguem detectar ✗
- d. falhas que se manifestam consistentemente sob condições bem definidas ✓

Correto
Notas relativas a este envio: 1/1.

5

Notas: 1

Determine se a sentença a seguir é verdadeira ou falsa:
"Toda a ativação de uma falha de software sempre causa um defeito imediatamente pois a falha é fixa no código."

Resposta:

- Verdadeiro ✗
- Falso ✓

Correto
Notas relativas a este envio: 1/1.

6
Notas: 1

Determine se a afirmação a seguir é verdadeira ou falsa:
"Uma falha de software, que foi ativada pela execução da parte do código onde a falha está localizada, produz uma condição interna no sistema diferente da condição correta. Essa condição interna é conhecida como erro."

Resposta:
- (•) Verdadeiro ✓
- ( ) Falso ✗

Correto
Notas relativas a este envio: 1/1.

7
Notas: 1

Um erro pode se desenvolver em outros erros e assim sucessivamente até que um defeito finalmente ocorra. Essa cadeia de eventos é chamada de:

propagação de erros .

Correto
Notas relativas a este envio: 1/1.

8
Notas: 1

Uma instrução mal codificada na implementação de um algoritmo pode levar a um valor incorreto para uma variável de um programa. O programa pode usar esse valor incorreto para computações posteriores levando a incorreções adicionais. Posteriormente, um desses valores incorretos pode influenciar o comportamento observável do sistema. Considerando o exemplo, associe a descrição ao termo correspondente:

incorreção observável no comportamento do sistema:	defeito
valor incorreto em uma variável:	erro
execução do trecho de programa que contém a instrução mal codificada:	ativação da falha
instrução mal codificada:	falha
incorreções conduzindo a novas incorreções:	propagação de erro

Correto
Notas relativas a este envio: 1/1.

9
Notas: 1

Os autores apresentam **duas** explicações da razão porque o software pode se comportar de forma diferente em situações aparentemente idênticas. São elas:

Escolha pelo menos uma resposta.
- [ ] a. propagação do erro invertida, ou seja, um erro provocando uma falha ✗
- [ ] b. usuários com comportamento caótico e difícil de identificar ✗
- [ ] c. ocorrência falhas randômicas de hardware ✗
- [x] d. longo retardo entre a ativação da falha e a ocorrência do defeito ✓
- [x] e. outros elementos sistema podem influenciar o comportamento da falha numa aplicação específica ✓

Correto
Notas relativas a este envio: 1/1.

10
Notas: 1

Complete os espaços vazios com os termos correspondentes, de acordo com Grottke e Tivedi:

Uma falha pode fazer com que o software exiba um comportamento caótico e até não determinístico com respeito a ocorrência ou não de defeitos. Profissionais da área se referem a falhas com essa propriedade como Mandelbugs . Algumas vezes a literatura também chama essas falhas de Heisenbugs , referindo-se a falhas que trocam seu comportamento quando são observadas ou isoladas.

Correto
Notas relativas a este envio: 1/1.

11
Notas: 1

Segundo Michael Grottke e Kishor S. Trivedi, qual o principal problema que o desenvolvedor ou engenheiro de teste enfrenta com Mandelbugs?

Escolher uma resposta.
- ( ) a. envelhecerem e ficarem obsoletos ✗
- ( ) b. se transformarem em Bohrbugs ✗

c. grande probabilidade de não serem detectados durante o teste ✓

d. grande probabilidade de serem detectados durante o teste e com isso forçarem o programador a corrigir o código ✗

Correto
Notas relativas a este envio: 1/1.

12
Notas: 1

Qual a percentagem de falhas detectadas atribuídas a Mandelbugs após a entrada em operação de um software?

Escolher uma resposta.

a. 15 a 80% ✓

b. 90 a 100% ✗

c. 1 a 8% ✗

d. 30 a 40% ✗

Correto
Notas relativas a este envio: 1/1.

13
Notas: 1

De acordo com Grottke e Trivedi, qual a maneira simples e eficiente de lidar com falhas que aparentemente exibem comportamento não determinístico (Mandelbugs) e que não seria adequada para falhas do tipo Bohrbugs?

Escolher uma resposta.

a. retentativa, por exemplo restart ou reboot ✓

b. atualização de código ✗

c. substituição do software ✗

d. substituição da equipe de desenvolvedores ✗

e. reprojeto a partir da mesma especificação ✗

Correto
Notas relativas a este envio: 1/1.

14
Notas: 1

De acordo com Grottke e Trivedi, a maneira simples e eficiente de lidar com falhas que exibem comportamento não determinístico pode ser melhorada combinando-a com uma técnica conhecida por:

Escolher uma resposta.

a. detecção de falhas ✗

b. isolamento de falhas ✗

c. checkpointing ✓

d. armazenamento estável ✗

e. reboot ✗

Correto
Notas relativas a este envio: 1/1.

15
Notas: 1

De acordo com Michael Grottke e Kishor S. Trivedi, falhas de software são erros humanos que se escondem no código. Assim, por exemplo, diferentes instalações do mesmo sistema operacional executando as mesmas aplicações deveriam conter as mesmas falhas.

Resposta:

Verdadeiro ✓

Falso ✗

Correto
Notas relativas a este envio: 1/1.

16
Notas: 1

Acadêmicos têm discutido que replicação não se aplica a software com as mesmas vantagens obtidas quando aplicada a hardware. Quando se executam comandos em uma segunda instalação para o mesmo software, um usuário vai encontrar as mesmas falhas que levam ao mesmo defeito encontrado na primeira instalação. Grottke e Trivedi concordam com essa argumentação.

Resposta:

Verdadeiro ✗

Falso ✓

Correto
Notas relativas a este envio: 1/1.

17

Notas: 1

Acadêmicos têm discutido que replicação não se aplica a software com as mesmas vantagens obtidas quando aplicada a hardware. Quando se executam comandos em uma segunda instalação para o mesmo software, um usuário vai encontrar as mesmas falhas que levam ao mesmo defeito encontrado na primeira instalação. Grottke e Trivedi afirmam que essa argumentação só é válida se todas as falhas forem Bohrbugs.

Resposta:

- Verdadeiro ✓
- Falso ✗

Correto

Notas relativas a este envio: 1/1.

18

Notas: 1

Em que situações reiniciar um programa antes de ocorrer um defeito pode reduzir a ocorrência futura de defeitos?

Escolher uma resposta.

- a. quando a taxa de ocorrência de defeitos aumenta com o tempo de execução ✓
- b. quando a taxa de ocorrência de defeitos diminui com o tempo de execução ✗
- c. quando a taxa de ocorrência de defeitos se mantém constante em relação ao tempo de execução ✗
- d. nunca, isso é uma piada dos autores ✗

Correto

Notas relativas a este envio: 1/1.

19

Notas: 1

Sistemas de software rodando continuamente durante um longo período de tempo tendem a mostrar um desempenho degradado e uma taxa de defeitos crescente. Esse fenômeno é chamado de:

Escolher uma resposta.

- a. rejuvenescimento de software ✗
- b. obsolescência planejada ✗
- c. envelhecimento de software ✓
- d. fim da vida útil do software ✗
- e. inadequação de plataforma de hardware ✗

Correto

Notas relativas a este envio: 1/1.

20

Notas: 1

Sistemas de software rodando continuamente durante um longo período de tempo tendem a mostrar um desempenho degradado e uma taxa de defeitos crescente. Técnicas preventivas contra esse fenômeno são chamadas de:

Escolher uma resposta.

- a. replicação de software ✗
- b. rejuvenescimento de software ✓
- c. substituição de software ✗
- d. obsolescência planejada ✗
- e. envelhecimento de software ✗

Correto

Notas relativas a este envio: 1/1.

21

Notas: 1

Porque a taxa de ocorrência de defeitos de um programa aumenta com o tempo sem que o código sofra modificações? Grottke e Trivedi citam duas soluções possíveis a esse quebra-cabeça. Qual a **primeira** solução citada no artigo?

Escolher uma resposta.

- a. O tempo total que o sistema opera continuamente pode influenciar a taxa de ativação de bugs relacionados a envelhecimento. ✗
- b. O aquecimento provocado por longa operação contínua faz com que bits da memória que contém o código troquem aleatoriamente gerando alterações no programa sendo executado. ✗
- c. A atenção dos operadores do sistema é influenciada pelo tempo total que o sistema opera continuamente fazendo com que a taxa de falhas de interação aumente sensivelmente. ✗

d. Bugs podem provocar erros que se acumulam com o tempo (como erros e arredondamento em operações aritméticas). ✓

Correto
Notas relativas a este envio: 1/1.

22
Notas: 1

Porque a taxa de ocorrência de defeitos de um programa aumenta com o tempo sem que o código sofra modificações? Grottke e Trivedi citam duas soluções possíveis a esse quebra-cabeça, qual a **segunda** solução citada no artigo?

Escolher uma resposta.

- a. O tempo total que o sistema opera continuamente pode influenciar a taxa de ativação de bugs relacionados a envelhecimento. ✓
- b. O acúmulo de atualizações ao longo da vida útil de um software pode aumentar a taxa de bugs em uma dada instalação. ✗
- c. A atenção dos operadores do sistema é influenciada pelo tempo total que o sistema opera continuamente fazendo com que a taxa de falhas de interação aumente sensivelmente. ✗
- d. Bugs podem provocar erros que se acumulam com o tempo (como erros e arredondamento em operações aritméticas). ✗
- e. O aquecimento provocado por longa operação contínua faz com que bits da memória que contém o código troquem aleatoriamente gerando alterações no programa sendo executado. ✗

Correto
Notas relativas a este envio: 1/1.

23
Notas: 1

Bugs relacionados a envelhecimento são Mandelbugs .

Correto
Notas relativas a este envio: 1/1.

24
Notas: 1

Rejuvenecimento pode limpar condições de erro internas em um sistema, mas tem como consequência:

Escolher uma resposta.

- a. necessidade de monitoramento online ✗
- b. necessitar de um conhecimento completo das falhas internas e suas consequências ✗
- c. maior esforço no desenvolvimento do projeto de software ✗
- d. maior custo ✓
- e. necessidade de reproduzir e isolar bugs ✗

Correto
Notas relativas a este envio: 1/1.

25
Notas: 1

Devido ao custo associado, rejuvenescimento requer otimização do momento de atuação (*optimal timing*). Gottke e Trivedi sugerem duas abordagens para essa otimização. Associe o conceito ao termo:

usa modelos analíticos para capturar a degradação do sistema e necessidade de rejuvenescimento	abordagem baseada em modelos
monitora atributos do sistema que possam mostrar sinais de envelhecimento	abordagem baseada em medidas

Correto
Notas relativas a este envio: 1/1.

Fechar esta janela Fechar esta janela

1

Notas: 1

Considerando o item 5 do artigo "Basic Concepts and Taxonomy of Dependable and Secure Computing", de Avizienis e demais autores, assinale **verdadeiro** para os termos que correspondem aos "meios" que foram mencionados no artigo para alcançar dependabilidade e **falso** caso contrário.

redundância	Falso
remoção de falhas	Verdadeiro
confidencialidade	Falso
disponibilidade	Falso
confiabilidade	Falso
prevenção de falhas	Verdadeiro
tolerância a falhas	Verdadeiro
previsão de falhas	Verdadeiro

Correto
Notas relativas a este envio: 1/1.

2

Notas: 1

Melhorar o processo de desenvolvimento para reduzir o número de falhas introduzidas no sistema durante sua produção é um exemplo de:

Escolher uma resposta.

- Remoção de falhas ✗
- Previsão de falhas ✗
- Tolerância a falhas ✗
- Prevenção de falhas ✓
- Perda de tempo porque falhas são inevitáveis ✗

Errado
Notas relativas a este envio: 0/1.

3

Notas: 1

Tolerância a falhas visa principalmente:

Escolher uma resposta.

- prevenir que falhas ocorram ✗
- aprender a conviver com defeitos ✗
- evitar falhas ✗
- evitar defeitos ✓
- remover falhas ✗

Correto
Notas relativas a este envio: 1/1.

4

Notas: 1

De acordo com Avizienis, tolerância a falhas emprega duas estratégias básicas. São elas:

Escolher uma resposta.

- redundância de hardware e isolamento de falhas ✗
- metodologias para o desenvolvimento de software e regras para o projeto de hardware ✗
- detecção de erro e recuperação do sistema ✓
- compensação de erros e reinicialização ✗
- diagnóstico de falhas e reconfiguração do sistema ✗

Correto
Notas relativas a este envio: 1/1.

5

Notas: 1

Considerando as técnicas de tolerância a falhas, associe o conceito à técnica apropriada.

Transforma um estado que contém um ou mais erros e falhas em um estado sem os erros detectados e sem falhas que possam ser novamente ativadas.	Recuperação

Identifica a presença de um erro.	Detecção de erro
Elimina erros do estado do sistema.	Tratamento de erros
Previne nova ativação de falhas.	Tratamento de falhas

Correto
Notas relativas a este envio: 1/1.

6
Notas: 1

Assinale as técnicas para tratamento de erros:

Escolha pelo menos uma resposta.

- ☐ diagnóstico ✗
- ☑ recuperação por retorno ✓
- ☐ reinicialização ✗
- ☐ isolamento ✗
- ☐ reconfiguração ✗
- ☑ compensação ✓
- ☑ recuperação por avanço ✓

Correto
Notas relativas a este envio: 1/1.

7
Notas: 1

Assinale as técnicas para tratamento de falhas:

Escolha pelo menos uma resposta.

- ☑ reconfiguração ✓
- ☑ reinicialização ✓
- ☐ compensação ✗
- ☐ recuperação por retorno ✗
- ☑ isolamento ✓
- ☑ diagnóstico ✓
- ☐ recuperação por avanço ✗

Correto
Notas relativas a este envio: 1/1.

8
Notas: 1

Associe o conceito à técnica de tratamento de erros adequada.

Leva o sistema a um estado salvo que ocorreu antes da ocorrência do erro:	recuperação por retomo
O estado com erros contém redundância suficiente para permitir que erros sejam mascarados:	compensação
O novo estado é um estado sem erros detectados:	recuperação por avanço

Correto
Notas relativas a este envio: 1/1.

9
Notas: 1

Considere a detecção de erros. Associe o conceito ao termo apropriado.

Ocorre quando o serviço é suspenso e se presta a verificar a presença de erros latentes e falhas dormentes.	detecção preemptiva
Ocorre durante a prestação normal de serviço.	detecção concorrente

Correto
Notas relativas a este envio: 1/1.

10
Notas: 1

Qual dos conceitos abaixo se aplica a diagnóstico?

Escolher uma resposta.

- ○ Verificação, atualização e registro de nova configuração e atualização das tabelas e registros do sistema. ✗
- ◉ Identificação e registro da causa do erro em termos de localização e tipo. ✓

- Chaveamento para componentes extra ou redistribuição de tarefas entre os componentes livres de falhas. ✗
- Exclusão física ou lógica do componente com falha, ou seja, transformação da falhaativa em dormente. ✗

Correto
Notas relativas a este envio: 1/1.

11
Notas: 1

Qual dos conceitos abaixo se aplica a isolamento de falha?

Escolher uma resposta.

- Verificação, atualização e registro de nova configuração e atualização das tabelas e registros do sistema. ✗
- Identificação e registro da causa do erro em termos de localização e tipo. ✗
- Exclusão física ou lógica do componente com falha, ou seja, transformação da falha em dormente. ✓
- Chaveamento para componentes extra ou redistribuição de tarefas entre os componentes livres de falhas. ✗

Correto
Notas relativas a este envio: 1/1.

12
Notas: 1

Qual dos conceitos abaixo se aplica a reconfiguração?

Escolher uma resposta.

- Exclusão física ou lógica do componente com falha, ou seja, transformação da falha em dormente. ✗
- Chaveamento para componentes extra ou redistribuição de tarefas entre os componentes livres de falhas. ✓
- Verificação, atualização e registro de nova configuração e atualização das tabelas e registros do sistema. ✗
- Identificação e registro da causa do erro em termos de localização e tipo. ✗

Correto
Notas relativas a este envio: 1/1.

13
Notas: 1

Qual dos conceitos abaixo se aplica a reinicialização?

Escolher uma resposta.

- Exclusão física ou lógica do componente com falha, ou seja, transformação da falha em dormente. ✗
- Verificação, atualização e registro de nova configuração e atualização das tabelas e registros do sistema. ✓
- Chaveamento para componentes extra ou redistribuição de tarefas entre os componentes livres de falhas. ✗
- Identificação e registro da causa do erro em termos de localização e tipo ✗

Correto
Notas relativas a este envio: 1/1.

14
Notas: 1

Assinale **verdadeiro** ou **falso** de acordo com os conceitos do artigo do Avizienis.

Um fator que distingue tolerância a falhas de manutenção é que manutenção é realizada por um agente externo.	Verdadeiro
Após a detecção de um erro, sempre deve seguir-se o tratamento do erro e só então, depois disso, o tratamento da falha.	Falso
As técnicas de detecção de erros preemptiva e tratamento de erros, possivelmente acompanhadas de tratamento de falha, são comumente executadas no power up do sistema.	Verdadeiro
Recuperação por avanço (rollforward) e recuperação por retorno (rollback) são mutuamente exclusivas, ou seja, só pode ser usada uma ou outra.	Falso
Falhas intermitentes não necessitam isolamento ou reconfiguração.	Verdadeiro

Correto
Notas relativas a este envio: 1/1.

15

Assinale **verdadeiro** ou **falso** de acordo com o artigo do Avizienis (item 5.2.2).

Notas: 1

As classes de falha que podem realmente ser toleradas em um sistema dependem da hipótese de falha que foi considerada no processo de desenvolvimento do sistema.	Verdadeiro
Um método popular de alcançar tolerância a falhas é realizar múltiplas computações através de múltiplos canais, seja sequencialmente ou concorrentemente.	Verdadeiro
A escolha das técnicas de tolerância a falhas a serem empregadas depende do custo do projeto e dos recursos disponíveis e não da hipótese de falha considerada.	Falso
Para falhas físicas, canais redundantes podem apresentar projeto idêntico pois podemos nos basear na hipótese que componentes de hardware apresentam defeitos independentemente uns dos outros.	Verdadeiro
Para falhas sólidas de projeto, canais redundantes podem apresentar projeto idêntico pois componentes de hardware sempre apresentam defeitos de forma independente uns dos outros.	Falso

Correto
Notas relativas a este envio: 1/1.

16

Notas: 1

A existência de recursos internos ao componente, seja de hardware ou software, para que o componente, além de executar sua função, também possa realizar detecção concorrente de erros está relacionada ao conceito de:

Escolher uma resposta.

- projeto diversitário ✗
- área de confinamento de erro ✗
- componente auto-verificável (self-checking) ✓
- recuperação do retorno (rollback) ✗
- cobertura ✗

Correto
Notas relativas a este envio: 1/1.

17

Notas: 1

Nem todas as técnicas de tolerância a falhas são igualmente efetivas. A medida de eficiência de uma dada técnica de tolerância a falhas é chamada de (escreva o nome em português):

Resposta: Cobertura

Correto
Notas relativas a este envio: 1/1.

18

Notas: 1

As imperfeições de tolerância a falhas, segundo Avizienis e demais autores, são devidas a dois fatores. São eles:

Escolher uma resposta.

- ausência de projeto diversitário e de componentes auto-verificáveis ✗
- deficiência na cobertura do tratamento de erros e falhas e deficiências nas hipóteses sobre falhas ✓
- presença de falhas dependentes e ausência de isolamento de falhas ✗
- problemas com confinamento de falhas e projeto diversitário mal especificado ✗

Correto
Notas relativas a este envio: 1/1.

19

Notas: 1

Avizienis e demais autores do artigo sob análise afirmam que tolerância a falhas é um conceito recursivo. Qual o significado desta afirmação?

Escolher uma resposta.

- a. Conceito recursivo significa neste contexto que os mecanismos de tolerância a falhas não podem ser implementados nem em software nem em hardware pois sempre resta um componente não protegido. ✗
- b. Conceito recursivo significa neste contexto que os mecanismos que implementam tolerância a falhas são auto-suficientes, auto-contidos e não sofrem interferência de falhas. ✗
- c. Conceito recursivo significa neste contexto que os mecanismos que implementam tolerância a falhas também devem ser protegidos contra as falhas que possam afetá-los. ✓

d. Conceito recursivo significa neste contexto que os mecanismos que implementam tolerância a falhas são extremamente complexos e só podem ser implementados em software por linguagens de programação que permitem recursão. ✗

e. Conceito recursivo significa neste contexto que os mecanismos de tolerância a falhas não podem ser implementados em hardware, pois componentes de hardware não são recursivos. ✗

Correto
Notas relativas a este envio: 1/1.

20
Notas: 1

Segundo Avizienis e demais autores, os termos auto-reparo, auto-cura e resiliência são:

Escolher uma resposta.

- relativos a técnicas de tratamento de defeitos ✗
- relativos a técnicas específicas de tratamento de erros ✗
- referentes a técnicas complementares a tolerância a falhas ✗
- sinônimos de dependabilidade ✗
- sinônimos de tolerância a falhas ✓

Correto
Notas relativas a este envio: 1/1.

21
Notas: 1

Um dos quatro meios para atingir dependabilidade é a remoção de falhas. Segundo Avizienis, a remoção de falhas durante a fase de desenvolvimento de um sistema consiste em 3 passos. São eles:

Escolher uma resposta.

- verificação, diagnóstico e correção ✓
- teste dinâmico, teste estático e injeção de falhas ✗
- rollback, rollforward e compensação ✗
- detecção de erro, tratamento de erro e tratamento de falha ✗
- verificaçao estática, verificação dinâmica e execuçao simbólica ✗

Correto
Notas relativas a este envio: 1/1.

22
Notas: 1

Remoção de falhas durante a fase de uso (ou operação) de um sistema compreende duas técnicas. São elas:

Escolher uma resposta.

- verificação e correção ✗
- manutenção corretiva e manutenção preventiva ✓
- isolamento e remoção ✗
- recuperação e compensação ✗
- detecção e tratamento de erro ✗

Correto
Notas relativas a este envio: 1/1.

23
Notas: 1

Realizar uma avaliação do comportamento de um sistema com respeito a ocorrência de falhas e sua ativação é uma tarefa relacionada a:

Escolher uma resposta.

- remoção de falhas ✗
- prevenção de falhas ✗
- previsão de falhas ✓
- tolerância a falhas ✗

Correto
Notas relativas a este envio: 1/1.

24
Notas: 1

Considerando avaliação do comportamento do sistema com respeito a ocorrência de falhas, associe o conceito ao tipo de avaliação:

visa avaliar, em termos de probabilidades, a extensão em que certos atributos são satisfeitos; os atributos são então vistos como "medidas"	avaliação quantitativa ou probabilística

visa identificar, classificar e ordenar os modos de defeito que podem levar a defeitos do sistema	avaliação qualitativa	Correto Notas relativas a este envio: 1/1.	Fechar esta janela Fechar esta janela

1

Notas: 1

No item 2.1, quando argumentam sobre a necessidade de tolerância a falhas de software, Florio e Blondia comentam sobre tolerância a falhas de hardware. Leia o artigo e assinale verdadeiro ou falso segundo as opiniões expressas no texto.

Componentes de hardware são apenas uma das fontes de não confiabilidade em sistemas computacionais, decrescendo em importância conforme a confiabilidade desses componentes aumenta.	Verdadeiro
Durante muitos anos, a pesquisa na área de tolerância a falhas foi concentrada em desenvolver estruturas de hardware engenhosas e eficientes para lidar com falhas.	Verdadeiro
A prevalência de falhas de hardware está aumentando continuamente.	Falso
Durante algum tempo, tolerância a falhas de software foi considerada a única necessária para alcançar a disponibilidade e a integridade de dados demandadas por computadores modernos.	Falso

Correto

Notas relativas a este envio: 1/1.

2

Notas: 1

Florio e Blondia, no item 2.2, justificam prover tolerância a falhas de software no nível da aplicaçao baseados no fato que a maioria dos defeitos experimentados por computadores modernos são devidos a:

Escolher uma resposta.

- a. falhas de hardware ✗
- b. falhas em device drivers e kernel de sistemas operacionais ✗
- c. falhas de interação com o usuário ✗
- d. quedas de serviço ✗
- e. falhas de software incluindo as falhas no nível de aplicação ✓

Correto

Notas relativas a este envio: 1/1.

3

Notas: 1

Um dos argumentos apresentados por Florio e Blondia para justificar tolerância a falhas de software no nível da aplicação é a adoção popular de componentes de software reusáveis. Muitas aplicações orientadas a objetos são construídas com componentes reusáveis. Considerando reuso de componentes, podemos afirmar, de acordo com a opinião dos autores, que:

Escolher uma resposta.

- a. o reuso de componentes diminui os custos de desenvolvimento mas a confiabilidade desses componentes e o seu impacto na confiabilidade da aplicação são frequentemente desconhecidos ✓
- b. o reuso de componentes aumenta os custos de desenvolvimento mas contribui para aumento de confiabilidade porque os componentes já foram exaustivamente testados ✗
- c. o desenvolvedor da aplicação não conhece o desenvolvedor do componente reusável e por essa razão não pode confiar na qualidade do componente ✗
- d. o reuso de componentes aumenta a complexidade do projeto de sistemas e, além disso, a confiabilidade desses componentes é geralmente inadequada ✗

Correto

Notas relativas a este envio: 1/1.

4

Notas: 1

Um dos argumentos apresentados por Florio e Blondia para justificar tolerância a falhas de software no nível da aplicação, que eles consideram como provavelmente o mais convicente, é chamado de argumento fim-a-fim. Considerando esse argumento e a opinião dos autores, assinale verdadeiro e falso.

soluçoes de tolerância a falhas baseadas puramente em hardware ou no sistema operacional não são capazes de prover tolerância a falhas completa e fim-a-fim na aplicação do usuário	verdadeiro
como exemplo é citado que um canal de comunicação extraordinariamente confiável livra a aplicação da obrigação de prover confiabilidade	falso
o argumento implica que todas as tarefas de tolerância a falhas podem ser realizadas no nível da aplicação	falso
frequentemente funções só podem ser completamente e corretamente implementadas com o conhecimento e auxílio da aplicação que se encontra nas extremidades do sistema de suporte	verdadeiro

Correto

Notas relativas a este envio: 1/1.

5

Notas: 1

No item 2.3, os autores listam problemas de projeto relacionados a tolerância a falhas no nível da aplicação. Qual o problema que foi proposto primeiramente por Randell?

Escolher uma resposta.

- a. quais mecanismos de tolerância a falhas devem ser suportados ✗
- b. como reduzir os custos de projeto ✗
- c. como treinar a equipe de desenvolvedores para aplicar tolerância a falhas no nível da aplicação ✗
- d. como incorporar tolerância a falhas no nível da aplicação ✓
- e. como gerenciar o código tolerante a falhas ✗

Correto

Notas relativas a este envio: 1/1.

6

Notas: 1

Randell afirmou ser necessário o uso de técnicas de estruturação apropriadas para a incorporação de tolerância a falhas no nível da aplicação. Soluções inadequadas a esse problema, segundo o artigo de Florio e Blondia, podem provocar alto grau de intrusividade no código .

Correto

Notas relativas a este envio: 1/1.

7

Notas: 1

A falta de estruturaçao adequada, segundo o artigo de Florio e Blondia, resulta na mistura do código funcional com o código não-funcional. Assinale verdadeiro para as consequências ao desenvolvimento do projeto resultantes dessa mistura:

o custo e o tempo de desenvolvimento aumentam	verdadeiro
a complexidade do sistema diminui	falso
o código resultante é mais fácil de ser mantido	falso
o tamanho do código resultante é menor	falso
as tarefas do desenvolvedor ficam significativamente mais complicadas	verdadeiro

Correto

Notas relativas a este envio: 1/1.

8

Notas: 1

Com relação a incorporação de tolerância a falhas no nível da aplicação, Florio e Blondia concluem que:

Escolher uma resposta.

- a. os aspectos de tolerância a falhas devem prevalecer em relação aos aspectos funcionais no código da aplicação ✗
- b. a incorporação neste nível é inadequada pois é impossível conciliar interesses ✗
- c. uma estrutura ideal de sistema deve garantir uma adequada separação entre os interesses funcionais e os relativos a tolerância a falhas no código da aplicação ✓
- d. equipes diferentes de desenvolvedores devem ser encarregados da parte funcional do código e da parte de suporte a tolerância a falhas ✗

Correto

Notas relativas a este envio: 1/1.

9

Notas: 1

Florio e Blondia definem 3 atributos estruturais para ALTF chamados de SC, SA e A. Associe:

separação de interesses funcionais e não funcionais	SC
suporte direto a um grande conjunto de mecanismos de tolerância a falhas	SA
facilidade do código tolerante a falhas se adaptar ao ambiente onde se encontra no momento	A

Correto

Notas relativas a este envio: 1/1.

Notas: 1

Considerando o conceito de redundância como apresentada no item 2.3.1 do artigo de Florio e Blondia, é possível afirmar que:

Redundância requer grande quantidade de recursos extras e portanto implica em alto custo.	Verdadeiro
É impossível acrescentar mais confiabilidade a um canal não confiável aumentando o grau de redundância de informação.	Falso
Tolerância a falhas é, em geral, o resultado de alguma estratégia que efetivamente explora alguma forma de redundância.	Verdadeiro
Redundância consome uma grande quantidade de recursos extras e rapidamente exaure esses recursos.	Verdadeiro

Correto
Notas relativas a este envio: 1/1.

11

Notas: 1

Assinale as 6 classes de métodos, mencionadas no artigo de Florio e Blondia no item 3, que podem ser usadas para prover tolerância a falhas no nível de aplicação em programas de computador.

redundância espacial	não
redundância temporal	não
baseada em monitoração da tarefa de recuperação	sim
programação orientada a aspectos	sim
novas linguagens de programação distribuídas customizadas	sim
tolerância a falhas embarcada	não
baseada em protocolos para metaobjetos	sim
tolerância a falhas em software com múltiplas versões	sim
tolerância a falhas em software de versão única	sim

Correto
Notas relativas a este envio: 1/1.

12

Notas: 1

Qual o requisito chave para o desenvolvimento de sistemas tolerantes a falhas (de acordo com Florio e Blondia, item 3.1)?

Escolher uma resposta.

- a. Recursos replicados em hardware ✗
- b. Replicação temporal ✗
- c. Recursos replicados em hardware ou em software ✓
- d. Recursos replicados em software ✗
- e. Replicação de dados ✗

Correto
Notas relativas a este envio: 1/1.

13

Notas: 1

A replicação é usada como método fundamental para alcançar tolerância a falhas com múltiplas computações. Quais os seus domínios ?

Escolher uma resposta.

- a. Tempo, espaço e informação ✓
- b. Espaço, paralelismo e codificação de dados ✗
- c. Codificação de dados, tempo e concorrência ✗
- d. Tempo, concorrência e paralelismo ✗

Correto
Notas relativas a este envio: 1/1.

14

Notas: 1

Considerando os domínios de replicação de software, associe:

Repetição da computação	tempo

Adoção de múltiplas versões de software	informação	
Adoção de múltiplos canais de hardware	espaço	

Correto
Notas relativas a este envio: 1/1.

15
Notas: 1

Considerando os domínios de replicação de software (item 3.1 do artigo sob análise), associe o significado aos termos na notação nT/mH/pS:

número de execuções	n
número de programas	p
número de canais de hardware	m

Correto
Notas relativas a este envio: 1/1.

16
Notas: 1

Um sistema 1T/1H/1S é chamado de:

Escolher uma resposta.

- a. intolerante a falhas ✗
- b. tolerante a falhas ✗
- c. replicado ✗
- d. duplex ✗
- e. simplex ✓

Correto
Notas relativas a este envio: 1/1.

17
Notas: 1

Tolerância a falhas em versão única de software implica em:

Escolher uma resposta.

- a. Embutir em um sistema simplex (ou seja, sem réplicas) técnicas de detecção e recuperação de erros. ✓
- b. Usar externamente aos programas aplicativos as técnicas de ações atômicas, pontos de recuperação e recuperação por retorno e tratamento de exceções. ✗
- c. Construir programas aplicativos sem erros. ✗
- d. Construir programas que não possam ser copiados nem replicados. ✗
- e. Executar software em ambientes que inibam múltiplos processos e múltiplos usuários simultâneos. ✗

Correto
Notas relativas a este envio: 1/1.

18
Notas: 1

A adoção de tolerância a falhas em versão única de software apresenta como desvantagens:

Escolher uma resposta.

- a. necessidade de desenvolver bibliotecas e frameworks para suprir os recursos de tolerância a falhas ✗
- b. aumento do tamanho da aplicação com consequente perda de transparência, da facilidade de manutençao e da portabilidade, acompanhados do aumento do tempo de desenvolvimento e dos custos ✓
- c. obrigação do uso estrito de processos de engenharia de software ✗
- d. necessidade de optar entre o sistema EFTOS e o sistema SwiFT ✗

Correto
Notas relativas a este envio: 1/1.

19
Notas: 1

Uma solução parcial, segundo os autores, para a perda de portabilidade e aumento dos custos associados a embutir estratégias de tolerância a falhas em software de versão única seria:

Escolher uma resposta.

- a. o uso de bibliotecas e frameworks criados seguindo criteriosamente processos de engenharia de software ✓
- b. uso eficiente de linguagens de baixo nível ✗

c. o uso de estratégias como ações atômicas, recuperação por retorno e tratamento de exceções ✗

d. obrigação do uso estrito de processos de engenharia de software ✗

e. o uso do sistema EFTOS e do sistema SwiFT ✗

Correto
Notas relativas a este envio: 1/1.

20
Notas: 1

Considerando o sistema SwiFT, assinale as sentenças verdeiras e falsas.

SwiFT é o nome de um computador tolerante a falhas que foi popular na década de 90	Falso
SwiFT inclui uma série de componentes reutilizáveis de software adequados para incluir técnicas de tolerância a falhas em software de versão única	Verdadeiro
SwiFT introduziu apenas um pequeno overhead em custo na maioria dos caso	Verdadeiro
SwiFT foi usado com sucesso para aumentar a tolerância a falhas em sistemas de software onde falhas residuais estavam presentes	Verdadeiro
SwiFT é o nome de um projeto que visava construir componentes de hardware para facilitar a execução de software tolerante a falhas	Falso

Correto
Notas relativas a este envio: 1/1.

21
Notas: 1

Na conclusão do item 3.1.1, Florio e Blondia, após reforçarem uma série de desvantagens da abordagem de versão única, alertam para uma característica importante dessa abordagem. Qual é ela?

Escolher uma resposta.

a. apresenta suficiente adaptabilidade ✗

b. não apresenta impacto nos custos de desenvolvimento e manutenção ✗

c. não apresenta qualquer restrição quanto a classe de aplicação que pode se beneficiar da abordagem ✓

d. apresenta um alto grau de portabilidade e facilidade de manutenção ✗

Correto
Notas relativas a este envio: 1/1.

22
Notas: 1

Na conclusão do item 3.1.1, Florio e Blondia, após reforçarem uma série de desvantagens, falam das vantagens de utilizar, em software de versão única, ferramentas que permitem ao usuário tratar com átomos de tolerância a falhas sem se preocupar com sua implementação. Em relação a essas ferramentas podemos afirmar que:

Escolher uma resposta.

a. introduzem grande overhead no desempenho para apenas um pequeno aumento nos atributos de dependabilidade ✗

b. permitem ao projetista reusar peças de software sofisticadas e amplamente testadas ✓

c. só podem ser usadas em programas escritos em C ou C++ ✗

d. prejudicam a dependabilidade do sistema ✗

e. não apresentam qualquer impacto no custo ou no desempenho da aplicação ✗

Correto
Notas relativas a este envio: 1/1.

Terminar revisão

1

Notas: 1

O item 3.1.2 do artigo de Florio e Blondia trata de tolerância a falhas em software multi-versão (MV). Esse tipo de abordagem requer um número mínimo de versões do mesmo programa, que é igual a:

Escolher uma resposta.

- a. sT/yH/nS ✗
- b. 3 ✗
- c. qualquer número ímpar para não haver empate na votação ✗
- d. 2 ✓
- e. 1 ✗

Correto

Notas relativas a este envio: 1/1.

2

Notas: 1

No item 3.1.2, Florio e Blondia afirmam que a abordagem MV exige N versões de software com certa particularidade importante. Qual é:

Escolher uma resposta.

- a. Todas as versões devem ser idênticas. ✗
- b. Cada uma das N versões deve ter sido projetada e desenvolvida por um time diferente e independente. ✓
- c. Todas as N versões devem apresentar o mesmo desempenho. ✗
- d. O número N de versões deve ser ímpar para não haver empate na votação ✗
- e. Todas as N versões devem executar sincronizadamente sobre canais idênticos de hardware. ✗

Correto

Notas relativas a este envio: 1/1.

3

Notas: 1

A abordagem MV tem por objetivo:

Escolher uma resposta.

- a. aumentar a testabilidade do software ✗
- b. facilitar a detecção de erros ✗
- c. reduzir os efeitos de falhas de projeto devido a erros humanos cometidos durante o projeto ✓
- d. facilitar a manutenção do software ✗
- e. tolerar falhas transitórias de hardware ✗

Correto

Notas relativas a este envio: 1/1.

4

Notas: 1

A abordagem MV é subdividida, segundo Florio e Blondia, em duas abordagens principais. Assinale **sim** para as abordagens abaixo que correspondem às citadas no artigo.

ações atômicas distribuídas	não
programação distribuída	não
tratamento de exceções	não
programação concorrente	não
bloco de recuperação	sim
programação N-versões	sim

Correto

Notas relativas a este envio: 1/1.

5

Notas: 1

A abordagem conhecida com o nome de blocos de recuperação pode ser implementada da forma NT/1H/NS ou 1T/NH/NS, sendo a primeira a mais frequente. Conhecendo essas expressões podemos afirmar que:

Escolher uma resposta.

- a. N se refere ao número de recuperações necessárias para garantir o correto funcionamento dos módulos, também chamados de blocos, do programa. ✗

b. A abordagem opera com 1 programa executado N vezes em série e N vezes em paralelo. ✗

c. A abordagem opera com N programas encadeados em série formando um único bloco de execução para fins de recuperação. ✗

d. A abordagem opera com N programas executados em paralelo no mesmo hardware. ✗

e. A abordagem opera com N programas executados ou em série no mesmo hardware ou em paralelo sobre N canais de hardware. ✓

Correto
Notas relativas a este envio: 1/1.

6
Notas: 1

Na abordagem conhecida com o nome de blocos de recuperação, cada um dos N componentes funcionais, programas ou versões é chamado alternativa. Assinale a opção mais próxima do conceito da abordagem.

Escolher uma resposta.

a. Cada alternativa colabora com as demais executando parte da computação; as diferentes partes são então combinadas após passar por um teste de aceitação. ✗

b. Cada alternativa executa toda computação;os resultados são então comparados por um teste de aceitação que escolhe o melhor. ✗

c. Cada alternativa executa toda computação; os resultados são então avaliados por um teste de aceitação que calcula a média entre os resultados para compor o resultado final. ✗

d. A cada execução é escolhida uma nova alternativa para que os programas não acumulem erros residuais. ✗

e. O resultado de cada alternativa, começando pela primária, é avaliado por um teste de aceitação; caso não seja aceito, o resultado da próxima alternativa é avaliada e assim sucessivamente até que um dos resultados passe no teste de aceitação. ✓

Correto
Notas relativas a este envio: 1/1.

7
Notas: 1

A eficácia da abordagem de blocos de recuperação reside na cobertura do mecanismo de detecção de erros adotado, portanto o componente mais crucial da abordagem é:

Escolher uma resposta.

a. a alternativa primária ✗

b. a cache de recuperação ✗

c. o teste de aceitação ✓

d. o chaveamento entre as alternativas ✗

e. o suporte para o mecanismo de recuperação ✗

Correto
Notas relativas a este envio: 1/1.

8
Notas: 1

Florio e Blondia analisando blocos de recuperação apontam alguns pontos positivos e negativos. Assinale verdadeiro se a opção corresponde a uma afirmação encontrada no artigo e falso se não corresponde.

A abordagem de blocos de recuperação foi validada com sucesso através de experimentos estatísticos e modelagem matemática.	verdadeiro
Não existem registros que a abordagem tenha sido usada com sucesso em algum campo de aplicação.	falso
Blocos de recuperação não são uma abordagem satisfatória considerando instrusividade no código, portabilidade e facilidade de manutenção.	verdadeiro
A abordagem de blocos de recuperação consiste em uma estratégia rígida sem possibilidade de configuração offline e com baixa adaptabilidade.	verdadeiro

Correto
Notas relativas a este envio: 1/1.

9
Notas: 1

Sistemas de programação N-versões (NVP) são baseados em:

Escolher uma resposta.

a. redundância e recuperação ✗

b. N versões do mesmo programa executados sequencialmente ✗

c. concorrência e teste de aceitação ✗

d. execucão paralela e replicação de código ✗

e. redundância e consenso ✓

Correto
Notas relativas a este envio: 1/1.

10
Notas: 1

O autor que introduziu o conceito de programação N-versões foi:

Escolher uma resposta.

a. Laprie ✗

b. Randell ✗

c. Florio ✗

d. Blondia ✗

e. Avizienis ✓

Correto
Notas relativas a este envio: 1/1.

11
Notas: 1

Na abordagem conhecida como programação N-versões, cada um dos N componentes funcionais é chamado versão. O autor da abordagem define que as versões devem ser geradas de forma independente . O número N de versões deve ser maior que 1 . As versões funcionalmente equivalentes devem ser desenvolvidas a partir da mesma especificação .

Correto
Notas relativas a este envio: 1/1.

12
Notas: 1

Na abordagem conhecida como programação N-versões, cada um dos N componentes funcionais é diferente dos demais e é chamado versão. Assinale a opção mais próxima do conceito da abordagem.

Escolher uma resposta.

a. Cada versão executa toda computação; os resultados são então processados por um algoritmo genérico de decisão e é escolhido o resultado final por consenso ou maioria. ✓

b. O resultado de cada versão, começando pela primária, é avaliado por um teste de consenso; caso não seja aceito, o resultado da próxima versão é avaliado e assim sucessivamente até que um dos resultados passe no teste. ✗

c. Cada versão colabora com as demais executando parte da computação; as diferentes partes são então combinadas após passar por consenso. ✗

d. A cada execução é escolhida uma nova versão para que os programas não acumulem erros residuais. ✗

e. Cada versão executa toda computação; os resultados são então avaliados por um teste de aceitação que escolhe o mais correto com o resultado final. ✗

Correto
Notas relativas a este envio: 1/1.

13
Notas: 1

A abordagem conhecida como programação N-versões é baseada em uma conjectura fundamental, qual seja:

Escolher uma resposta.

a. projetos independentes se traduzem em defeitos randômicos dos componentes, ou seja, independência estatística dos defeitos ✓

b. usar N cópias do mesmo programa ou N versões diferentes não influencia a tolerância a falhas de um sistema ✗

c. as N versões diferentes executadas em paralelo replicam as falhas residuais de software que se encontram dormentes do código fonte ✗

d. projetos independentes resultam em defeitos correlacionados de componentes, que rapidamente exaurem a redundância disponível ✗

Correto
Notas relativas a este envio: 1/1.

14
Notas: 1

Considerando a abordagem de programação N-versões, assinale verdadeiro ou falso de acordo como texto do artigo de Florio e Blondia.

Vários experimentos e estudos teóricos mostram que nem sempre é correto assumir que a geração das versões de forma independente reduza significativamente os defeitos correlacionados.	Verdadeiro

De acordo com Avizienis, a geração das versões de forma independente reduz significativamente os defeitos correlacionados.	Verdadeiro	Correto Notas relativas a este envio: 1/1.	**15** Notas: 1

Comparando as duas abordagens, blocos de recuperação e programação N-versões, selecione os recursos adequados.

Mecanismo usado para seleção de resposta na abordagem conhecida como programação N-versões:	consenso ou algoritmo de votação
Mecanismo usado para seleção de resposta na abordagem conhecida como blocos de recuperação:	teste de aceitação

Correto
Notas relativas a este envio: 1/1.

16
Notas: 1

Comparando as duas abordagens, blocos de recuperação e programação N-versões, em relação ao mecanismo usado para seleção de resultado, podemos afirmar que a primeira abordagem usa um mecanismo específico dependente da aplicação enquanto a segunda usa um mecanismo genérico independente da aplicação .

Correto
Notas relativas a este envio: 1/1.

17
Notas: 1

Comparando blocos de recuperação e programação N-versões quanto às saídas, podemos afirmar que a primeira abordagem admite diferentes saídas corretas das versões enquanto a segunda abordagem permite uma única saída correta .

Correto
Notas relativas a este envio: 1/1.

18
Notas: 1

Assinale verdadeiro ou falso:

Programação N-versões não é adequada a sistemas de tempo-real.	Verdadeiro
A aplicabilidade da programação N-versões está restrita a um pequeno número de linguagens de programação como Java e C++.	Verdadeiro
Programação N-versões foi adotada com sucesso em várias áreas de aplicação incluindo aplicações de segurança crítica espaciais e aéreas.	Falso
O componente conhecido como EE (executivo do ambiente de programação N-versões) é simples, mas dependente da aplicação.	Falso

Parcialmente correta
Notas relativas a este envio: 0.25/1.

19
Notas: 1

Assinale verdadeiro ou falso:

A programação N-versões nas arquiteturas 1T/NH/NS e NT/1H/NS tem uma função de custo que cresce linearmente com o número de versões.	Falso
Portabilidade da programação N-versões é restrita pela portabilidade do executivo do ambiente N-versões e pela portabilidade de cada uma das versões.	Verdadeiro
A adoção de programação N-versões sempre implica em penabilidades em relação a manutenibilidade e portabilidade.	Verdadeiro
A adoção de programação N-versões implica em um aumento substancial no custo de desenvolvimento e manutenção.	Verdadeiro

Correto
Notas relativas a este envio: 1/1.

20
Notas: 1

Segundo Florio e Blondia, o método MOPs (item 3.2 do artigo) permite, em alguns casos, alcançar graus adequados de:

Escolher uma resposta.

- a. desempenho e dependabilidade ✗

b. facilidade na implementação de técnicas de tolerância a falhas em qualquer linguagem de programação ✗

c. flexibilidade, transparência e separação de interesses funcionais e não funcionais ✓

d. correção e desempenho ✗

e. adaptabilidade e portabilidade ✗

Correto
Notas relativas a este envio: 1/1.

21
Notas: 1

Segundo Florio e Blondia, a ideia básica do método MOPs é:

Escolher uma resposta.

a. abrir a implementação do compilador de uma linguagem de alto nível de tal forma que o programador possa adotar semânticas diferentes e customizadas, ajustando assim a linguagem para as necessidades do usuário e do ambiente ✗

b. abrir a implementação do executivo de execução (runtime executive) de uma linguagem orientada a objetos de tal forma que o programador possa adotar semânticas diferentes e customizadas, ajustando assim a linguagem para as necessidades do usuário e do ambiente ✓

c. abrir a implementação dos protocolos de comunicação inter-objetos de forma a inserir comando específicos e padronizados de tolerância a falhas que os objetos são obrigados a executar ✗

d. abrir a implementação do sistema operacional de tal forma que o programador possa desenvolver novas chamadas do sistema (sys call), ajustando assim o sistema operacional para as necessidades do usuário e do ambiente ✗

e. adotar uma linguagem orientada a objetos como C++ ou Java e protocolos de comunicação certificados pelos maiores fabricantes para facilitar a interoperabilidade ✗

Correto
Notas relativas a este envio: 1/1.

22
Notas: 1

Usando MOPs o programador pode modificar o comportamento de características fundamentais como invocação de métodos, criação e destruição de objetos. Para fins de implementação de técnicas de tolerância a falhas, essa possibilidade de modificação permite:

Escolher uma resposta.

a. a gerência transparente de redundância espacial e temporal ✓

b. a verificação automática da correção de programas ✗

c. a substituição dinâmica de programas errados do usuário por programas corretos do sistema ✗

d. o aumento significativo do desempenho ✗

Correto
Notas relativas a este envio: 1/1.

23
Notas: 1

O conceito chave de suporte a MOPs (item 3.2 do artigo de Florio e Blondia) é:

Escolher uma resposta.

a. reflexão computacional ✓

b. Java ✗

c. objetos reais ✗

d. runtime executive ✗

e. orientação a aspectos ✗

Correto
Notas relativas a este envio: 1/1.

24
Notas: 1

MOPs oferecem ao programador de metanível uma representação de um sistema como um conjunto de elementos que representam e refletem propriedades de objetos reais. Esses elementos são chamados de:

Escolher uma resposta.

a. objetos confiáveis ✗

b. objetos tolerantes a falhas ✗

c. metaobjetos ✓

d. objetos de tempo real ✗

e. objetos funcionais ✗

Correto
Notas relativas a este envio: 1/1.

25
Notas: 1

De acordo com Florio e Blondia, item 3.2, preencha as lacunas:

O programador dos recursos de tolerância a falhas define uma quantidade de protocolos de metaobjetos e os associa com invocações de métodos ou outros casos gramaticais . A cada vez que o programa funcional entra um certo caso gramatical o/a protocolo correspondente é transparentemente executado .

Correto
Notas relativas a este envio: 1/1.

26
Notas: 1

Segundo Florio e Blondia, a abordagem baseada em MOPs parece constituir uma técnica promissora para a adoção efetiva, coerente e transparente de mecanismos e técnicas de TF. Assinale verdadeiro ou falso:

Nenhuma evidência experimental ou analítica permite estimar a praticidade e generalidade da abordagem MOPs.	Verdadeiro
Não há dúvidas que MOPs representa uma solução prática para a efetiva integração na aplicação de usuários da maioria dos mecanismos de TF existentes.	Falso
Alguns estudos na década de 90 comprovam a eficiência de MOPs em alguns casos.	Verdadeiro

Correto
Notas relativas a este envio: 1/1.

Fechar esta janela Fechar esta janela

1

Notas: 1

Qual o conceito de dependabilidade de acordo com Avizienis? Assinale a alternativa mais próxima

Escolher uma resposta.

- a. Não ocorrência de consequências catastróficas ao computador ou ao usuário. ✗
- b. Desempenho disponível para execução do serviço solicitado mesmo na presença de falhas. ✗
- c. Nível de qualidade provida pelo serviço fornecido pelo sistema ✗
- d. Confiança justificadamente depositada no serviço fornecido pelo sistema. ✓
- e. Integridade, confidencialidade e segurança do sistema durante fornecimento de serviço crítico. ✗

Correto
Notas relativas a este envio: 1/1.

2

Notas: 1

Duplicação de módulos de hardware e comparação das saídas como técnica de detecção apresenta como desvantagem:

Escolher uma resposta.

- a. Baixa cobertura na detecção de falhas ✗
- b. Alta latência na sinalização do erro. ✗
- c. Comparação de custo elevado e baixa confiabilidade. ✗
- d. Transparência em relação ao software. ✗
- e. Dificuldade na sincronização dos módulos. ✓

Correto
Notas relativas a este envio: 1/1.

3

Notas: 1

Qual dos conceitos abaixo se aplica a diagnóstico?

Escolher uma resposta.

- a. Chaveamento para componentes extra ou redistribuição de tarefas entre os componentes livres de falhas. ✗
- b. Exclusão física ou lógica do componente com falha, ou seja, transformação da falha em dormente. ✗
- c. Verificação, atualização e registro de nova configuração e atualização das tabelas e registros do sistema. ✗
- d. Identificação e registro da causa do erro em termos de localização e tipo ✓

Correto
Notas relativas a este envio: 1/1.

4

Notas: 1

Em que situações, de acordo com Grottke e Trivedi, reiniciar um programa antes de ocorrer um defeito pode reduzir a ocorrência futura de defeitos?

Escolher uma resposta.

- a. quando a taxa de ocorrência de defeitos diminui com o tempo de execução ✗
- b. nunca, isso é uma piada dos autores ✗
- c. quando a taxa de ocorrência de defeitos aumenta com o tempo de execução ✓
- d. quando a taxa de ocorrência de defeitos se mantém constante em relação ao tempo de execução ✗

Correto
Notas relativas a este envio: 1/1.

5

Notas: 1

Comparando os códigos i-de-2i e de Berger podemos afirmar que:

Escolher uma resposta.

- a. Ambos possuem capacidade de detectar falhas múltiplas de qualquer tipo. ✗
- b. Ambos podem detectar falhas múltiplas unidirecionais mas o segundo requer menos redundância que o primeiro. ✓
- c. Apenas o código de Berger pode detectar falhas múltiplas unidirecionais. ✗
- d. Ambos são códigos não separáveis. ✗

e. Ambos podem detectar falhas múltiplas unidirecionais mas o primeiro é mais eficiente que o segundo. ✗

Correto
Notas relativas a este envio: 1/1.

6
Notas: 1

Em relação à redundância ativa e à passiva, é correto afirmar que:

Escolher uma resposta.

- a. A redundância passiva combinada com a ativa é a técnica preferida para sistemas de transações pois combina baixo custo com alto desempenho. ✗
- b. Redundância passiva é usada para mascarar e localizar falhas em sistemas de missão crítica . ✗
- c. A redundância ativa envolve detecção e posterior mascaramento da falha e seu isolamento. ✗
- d. A redundância passiva é inadequada para sistemas críticos de tempo real pois consome muito tempo na recuperação prejudicando o desempenho. ✗
- e. A redundância ativa exige o emprego eficiente de uma estratégia de detecção de erros. ✓

Correto
Notas relativas a este envio: 1/1.

7
Notas: 1

Assinale a alternativa correta.

Escolher uma resposta.

- a. MTTR mede o tempo máximo de reparo de um sistema computacional. ✗
- b. MTTF é inversamente proporcional a taxa de defeitos. ✓
- c. Como o MTTR é muito menor que o MTTF, ele é considerado desprezível para a disponibilidade de um sistema. ✗
- d. MTTF e MTBF são medidas estatísticas não relacionadas. ✗
- e. MTTF mede o tempo médio entre defeitos de um sistema passível de reparo. ✗

Correto
Notas relativas a este envio: 1/1.

8
Notas: 1

Selecione a alternativa correta.

Escolher uma resposta.

- a. Diversidade se aplica apenas ao software. ✗
- b. A programação n-versões é eficiente apenas quando as versões não apresentam correlação. ✓
- c. Diversidade implica o uso de componentes idênticos e votação replicada. ✗
- d. Na programação n-versões, resultados incorretos das versões são descartados por um teste de aceitação. ✗
- e. Diversidade apresenta baixo custo de desenvolvimento e permite prova formal da confiabilidade alcançada. ✗

Correto
Notas relativas a este envio: 1/1.

9
Notas: 1

Selecione a alternativa correta.

Escolher uma resposta.

- a. Em um sistema NMR, quanto maior o N maior a confiabilidade e o desempenho inicial. ✗
- b. TMR é usado para missões curtas porque sua confiabilidade fica menor do que a de um sistema simplex após um certo tempo de operação. ✓
- c. NMR é usado para aumentar o desempenho em sistemas de alta confiabilidade. ✗
- d. TMR é adequado para mascarar falhas de projeto de hardware. ✗
- e. Como TMR opera com 3 componentes em paralelo, pode suportar falhas em até dois componentes. ✗

Correto
Notas relativas a este envio: 1/1.

10

Notas: 1

Segundo Avizienis, a classe de falhas de desenvolvimento evasivas e a classe de falhas físicas transientes podem ser agrupadas juntas em uma classe denominada:

Escolher uma resposta.

- a. falhas indeterminadas ✗
- b. erros soft ✗
- c. falhas múltiplas ✗
- d. falhas intermitentes ✓
- e. falhas independentes ✗

Correto
Notas relativas a este envio: 1/1.

11

Notas: 1

Considere os meios sugeridos para alcançar dependabilidade. Dentro deste contexto, quais as ações mais próximas ao conceito de previsão de falhas?

Escolher uma resposta.

- a. Verificar presença de falhas e removê-las. ✗
- b. Fornecer serviço esperado mesmo na presença de falhas ✗
- c. Corrigir as conseqüências da ocorrência de defeitos. ✗
- d. Evitar introduzir falhas durante projeto e construção do sistema. ✗
- e. Estimar presença de falhas e suas conseqüências. ✓

Errado
Notas relativas a este envio: 0/1.

12

Notas: 1

A abordagem conhecida com o nome de blocos de recuperação pode ser implementada da forma NT/1H/NS ou 1T/NH/NS, sendo a primeira a mais frequente. Conhecendo essas expressões, conforme Florio e Blondia, podemos afirmar que:

Escolher uma resposta.

- a. A abordagem opera com N programas executados ou em série no mesmo hardware ou em paralelo sobre N canais de hardware. ✓
- b. A abordagem opera com 1 programa executado N vezes em série e N vezes em paralelo. ✗
- c. A abordagem opera com N programas encadeados em série formando um único bloco de execução para fins de recuperação. ✗
- d. N se refere ao número de recuperações necessárias para garantir o correto funcionamento dos módulos, também chamados de blocos, do programa. ✗
- e. A abordagem opera com N programas executados em paralelo no mesmo hardware. ✗

Errado
Notas relativas a este envio: 0/1.

13

Notas: 1

Segundo Florio e Blondia, a abordagem MV (múltiplas versões) tem por objetivo:

Escolher uma resposta.

- a. reduzir os efeitos de falhas de projeto devido a erros humanos cometidos durante o projeto ✓
- b. tolerar falhas transitórias de hardware ✗
- c. facilitar a manutenção do software ✗
- d. facilitar a detecção de erros ✗
- e. aumentar a testabilidade do software ✗

Correto
Notas relativas a este envio: 1/1.

14

Notas: 1

Sobre as estratégias de recuperação por retorno (rollback) e recuperação por avanço (rollforward) podemos afirmar que:

Escolher uma resposta.

- a. Apenas rollforward substitui o estado incorreto por um novo estado nunca ocorrido durante a execução sem erros do sistema. ✓

b. Ambas são independentes da aplicação, mas apenas rollback substitui o estado incorreto por um estado novo nunca ocorrido durante a execução do sistema. ✗

c. Ambas ocorrem após a detecção de erros e são estratégias ideais para sistemas de tempo real devido ao baixo custo e facilidade de implementação. ✗

d. Apenas rollforward é baseada em checkpoints periódicos do estado do sistema, mas apresenta alto custo de implementação. ✗

e. Ambas se baseiam em checkpoints, mas são mutuamente excludentes em um mesmo sistema. ✗

Correto
Notas relativas a este envio: 1/1.

15
Notas: 1

Assinale a alternativa correta.

Escolher uma resposta.

a. Uma falha sempre provoca um erro, mas nem todo erro provoca um defeito. ✗

b. Um defeito é consequência de um erro que não foi corrigido ou mascarado. ✓

c. Falhas são possíveis de serem evitadas com projeto adequado e uso de componentes de qualidade. ✗

d. Um sistema com falha forçosamente vai apresentar um defeito. ✗

e. Um sistema com manutenção cuidadosa e regular está livre de falhas. ✗

Correto
Notas relativas a este envio: 1/1.

16
Notas: 1

Considerando redundância, assinale a alternativa falsa.

Escolher uma resposta.

a. Redundância sempre implica em aumento de custo (medido em desempenho, em volume, em energia consumida, etc...). ✗

b. Redundância temporal serve a para detecção de falhas transitórias e em certos casos também para permanentes. ✗

c. A redundância de hardware implica no uso de componentes extras de hardware. ✗

d. A redundância nem sempre está relacionada ao uso de componentes de hardware idênticos replicados. ✗

e. O uso de redundância sempre conduz ao aumento da confiabilidade de um sistema. ✓

Errado
Notas relativas a este envio: 0/1.

17
Notas: 1

Grottke e Trivedi classificam falhas de software em dois tipos: Bohrbugs e Mandelbugs. Assinale a alternativa correta:

Escolher uma resposta.

a. Bohrbugs podem ser detectados por revisão do código e Maldelbugs por testes dinâmicos. ✗

b. Mandelbugs são falhas de hardware que se manifestam como erros no software durante a fase operacional do sistema e por essa razão não são detectáveis durante a fase de testes. ✗

c. Mandelbugs são falhas que se manifestam consistentemente sob condições bem definidas, ou seja são reproduzíveis. ✗

d. Mandelbugs são falhas dificilmente reproduzíveis que geralmente provocam comportamento não determinístico no software. ✓

e. Falhas relacionadas ao envelhecimento de software (software-aging) são Bohrbugs. ✗

Correto
Notas relativas a este envio: 1/1.

18
Notas: 1

Dos atributos listados abaixo, indicar quais são atributos de dependabilidade, quais são atributos de segurança (*security*) e quais são atributos de ambos (dependabilidade e segurança) segundo artigo do Avizienis [2004]:

segurança funcional (safety)	dependabilidade
confidencialidade	segurança (security)
facilidade de manutenção	dependabilidade
confiabilidade	dependabilidade

integridade | ambos

Correto
Notas relativas a este envio: 1/1.

19
Notas: 1

Considere a seguinte situação: uma forte radiação provocou a troca de um bit no registrador apontador de instruções de um controlador embarcado; uma instrução foi trazida da posição de memória correspondente a esse endereço; essa instrução alterou a orientação de um satélite de comunicações e interrompeu ligações telefônicas de milhões de pessoas. Várias pessoas perderam dinheiro, pois estavam comparando ações naquele momento. Em relação ao sistema computacional podemos afirmar que:

Escolher uma resposta.

- a. A troca de bits no registrador corresponde à falha; o endereço corrompido de memória ao erro; alteração da orientação do satélite com a perda das conexões telefônicas ao defeito. ✓
- b. A troca de bits no registrador corresponde à falha; a alteração da orientação do satélite ao erro e a perda de dinheiro em ações ao defeito. ✗
- c. O endereço corrompido de memória corresponde à falha; a interrupção de ligações telefônicas ao erro; a perda de dinheiro em ações ao defeito. ✗
- d. A forte radiação corresponde à falha; o conteúdo corrompido de memória ao erro; a interrupção de ligações telefônicas ao defeito. ✗

Errado
Notas relativas a este envio: 0/1.

20
Notas: 1

Indique o código de detecção ou correção de erros adequado a cada situação:

Transferências de pacotes por redes de dados com grande probabilidade de falhas em rajada (bursts).	CRC
Meio ótico como CD ou DVD.	Reed-Solomon
Memória com predominância de falhas simples e sem possibilidade de nova aquisição de dados.	Paridade sobreposta
Transferências por barramentos paralelos com maior probabilidade de falhas que afetam bits adjacentes.	Paridade entrelaçada

Correto
Notas relativas a este envio: 1/1.

21
Notas: 1

Defeitos podem ser subdivididos em 4 pontos de vista, de acordo com Avizienis. Assinale a qual ponto de vista corresponde o exemplo fornecido:

serviço incorreto é percebido da mesma forma por todos os usuários	consistência
o conteúdo da informação recebida do serviço desvia do esperado	operação
perdas são detectadas e sinalizadas pelo sistema	dectabilidade
os modos de defeitos são ordenados por grau de severidade	domínio

Parcialmente correta
Notas relativas a este envio: 0.5/1.

22
Notas: 1

Considerando a curva que relaciona taxa de defeitos ao tempo de uso de componentes de hardware a partir da sua fabricação, responda as questões que seguem.

Fase onde é usual se colocar componentes em situação de stress para que os fracos logo apresentem defeitos e possam ser descartados:	mortalidade infantil
Fase usual de operação de componentes:	vida útil
Fase em que os componentes apresentam alta taxa de defeitos após um certo tempo em uso:	envelhecimento

Correto
Notas relativas a este envio: 1/1.

Fechar esta janela | Fechar esta janela

1

Notas: 1

Responda de acordo com a opinião do Iyer e demais autores:

Foram tradicionalmente associadas à corrupção de conteúdo de memórias: falhas transientes . Esse fenômeno foi inicialmente reportado na década de 50 em aplicações operando em situações adversas como na proximidade de testes nucleares e no espaço .

Correto

Notas relativas a este envio: 1/1.

2

Notas: 1

Desde a década de 70 sabe-se que circuitos de memória densos como DRAM e SRAM são suscetíveis a erros soft causados por partículas alfa do encapsulamento do circuito e raios cósmicos .

Correto

Notas relativas a este envio: 1/1.

3

Notas: 1

Apesar de dados disponíveis em pesquisas da Texas Instruments e da Intel, Iyers e demais autores afirmam que, com a crescente integração de dispositivos em um chip, a taxa de erros soft no nível do circuito devida aos elementos sequenciais tem:

Escolher uma resposta.

- a. claramente aumentado ✓
- b. diminuído apenas levemente ✗
- c. significativamente diminuído ✗
- d. permanecido constante ✗

Correto

Notas relativas a este envio: 1/1.

4

Notas: 1

Através dos anos, três tipos principais de medidas têm sido aplicadas para proteger os sistemas de erros soft. Marque com um **X** as três medidas mencionadas por Iyer e seus co-autores e marque com **-** as demais.

técnicas no nível do circuito	X
técnicas no nível funcional e comportamental	-
técnicas no nível de arquitetura	X
técnicas no nível elétrico e eletro-magnético	-
técnicas no nível de partículas sub-atômicas	-
técnicas no nível da aplicação	-
técnicas no nível lógico	X

Correto

Notas relativas a este envio: 1/1.

5

Notas: 1

Entre as técnicas no nível de circuitos, é mencionado o uso de práticas de projeto conservativas. De acordo com os autores, essas técnicas compreendem:

Escolher uma resposta.

- a. diminuir a probabilidade que uma partícula incidente possa alterar a saída do dispositivo ✗
- b. aumentar o tamanho do transistor ✗
- c. aumentar o tamanho dos circuitos ✗
- d. fortalecer o laço de feedback em cicuitos MOSFET ✗
- e. usar componentes de alta confiabilidade e excluir estilos de componentes sensíveis a radiação ✓

Errado

Notas relativas a este envio: 0/1.

6

Notas: 1

Assinale verdadeiro ou falso de acordo com a opinião de Iyer e demais autores:

7
Notas: 1

No nível lógico de hardware, as técnicas para proteger sistemas contra erros soft são relacionadas a detecção e recuperação de erros no circuito combinacional e nos flip-flops.	Verdadeiro
No nível lógico de hardware, as técnicas para proteger sistemas contra erros soft são fortemente dependentes de flip-flops redundantes.	Falso
No nível lógico de hardware, as técnicas para proteger sistemas contra erros soft são todas baseadas na geração de paridade na saída do circuito.	Falso

Correto
Notas relativas a este envio: 1/1.

Assinale verdadeiro ou falso de acordo com a opinião de Iyer e demais autores:

No nível de arquitetura, as técnicas para proteger sistemas contra erros soft incluem replicação de unidades funcionais e execução de aplicações em hardware redundante.	Verdadeiro
No nível de arquitetura, as técnicas para proteger sistemas contra erros soft incluem circuitos auto-verificadores.	Falso
No nível de arquitetura, as técnicas para proteger sistemas contra erros soft incluem verificação formal dos pipelines.	Falso

Correto
Notas relativas a este envio: 1/1.

8
Notas: 1

De acordo com Iyer, as abordagens de suporte a confiabilidade adotam três formas de redundância de maneira isolada ou combinada. São elas: temporal, hardware, informação . Para proteger memórias, caches e outros elementos de armazenamento, os projetistas geralmente aplicam redundância de informação .

Correto
Notas relativas a este envio: 1/1.

9
Notas: 1

Técnicas que exploram redundância de hardware são sensíveis a defeitos de modo comum ou correlacionados. Esses defeitos são causados por:

Escolher uma resposta.

- a. múltiplas falhas de mesma origem externa que afetam múltiplas unidades redundantes de hardware e provocam erros em cascata ✗
- b. uma falha que afeta o componente em série ao circuito redundante e assim compromete todo o funcionamento do circuito, anulando as vantagens da redundância. ✗
- c. múltiplas falhas de projeto ou concepção do circuito ✗
- d. uma falha que afeta múltiplas unidades redundantes de hardware da mesma maneira tornando o erro indetectável ✓

Correto
Notas relativas a este envio: 1/1.

10
Notas: 1

Considerando as técnicas implementadas em hardware para alcançar tolerância a falhas em comparação com técnicas implementadas em software, os autores afirmam que:

Escolher uma resposta.

- a. as técnicas implementadas em hardware são específicas para uma dada aplicação ✗
- b. as técnicas implementadas em hardware apresentam como vantagem a baixa latência de detecção de erros mas não são conscientes da aplicação ✓
- c. as técnicas implementadas em hardware apresentam uma grande latência de detecção de erros e provocam grande perda de desempenho ✗
- d. as técnicas implementadas em hardware são tão eficientes quanto as técnicas implementadas em software em relação a latência de detecção de erros, mas as primeiras interferem em maior grau no desempenho da aplicação ✗

Correto
Notas relativas a este envio: 1/1.

11

Os autores afirmam que vários estudos independentes baseados em injeção de falhas mostraram que apenas uma pequena percentagem de erros no nível do hardware se propaga e são observáveis no nível da aplicação. Eles

Notas: 1

atribuem esse fenômeno a dois fatores. Assinale X para um fator mencionado no artigo e - quando não foi mencionado no artigo.

hardware desenvolvido especificamente para um alvo	-
detecção e recuperação em baixo nível	X
mascaramento natural no sistema	X
baixa latência de detecção de erros	-
o hardware não é consciente da aplicação	-

Correto
Notas relativas a este envio: 1/1.

12

Notas: 1

Localize-se na pagina 20 do artigo do Iyer - item *Logic-level soft-error resilience*. Considerando flip-flops e lógica convencional, qual o recurso no nível lógico é mais vulnerável a erros soft?

Escolher uma resposta.

- a. lógica combinacional e flip-flops são igualmente vulneráveis ✗
- b. flip-flops ✓
- c. lógica combinacional ✗
- d. erros soft não afetam o hardware ✗

Correto
Notas relativas a este envio: 1/1.

13

Notas: 1

Iyer e seus co-autores citam duas técnicas como as principais para a proteção de flip-flops e latches contra erros soft. São elas:

Escolher uma resposta.

- a. manter o clock em zero e programar todos os elementos de baixo nível em C ✗
- b. fortalecer a lógica combinacional ✗
- c. reforçar circuitos ou flip-flops e desenvolver resiliência interna a erros soft ✓
- d. fortalecer o SEU e evitar usar células de armazenamento ✗
- e. construir todos os circuitos ou com entradas replicadas ou com saídas replicadas ✗

Correto
Notas relativas a este envio: 1/1.

14

Notas: 1

Iyer cita uma técnica interessante proposta por Nicolaidis para tratar erros soft em lógica combinacional usando latches sombras (shadow latch). Assinale verdadeiro ou falso.

Dependendo se o objetivo for detecção de erro ou mascaramento podem ser usados um latch sombra (e comparação) ou dois latches sombras (e votação) além do latch principal.	Verdadeiro
Na técnica proposta por Nicolaidis o latch sombra funciona atrasado em relação ao latch principal; desta forma, com grande probabilidade, apenas um dos latches é afetado por uma falha transiente.	Verdadeiro
A técnica proposta por Nicolaidis não usa redundância.	Falso

Correto
Notas relativas a este envio: 1/1.

15

Notas: 1

Localize-se na página 22 do artigo do Iyer (Error protection in current-generation processors). Os autores afirmam que o mecanismo mais usado para proteção contra erros-soft nos processadores atuais é:

Escolher uma resposta.

- a. projeto diversitário ✗
- b. redundância de informação ✓
- c. redundância de software ✗
- d. redundância temporal ✗
- e. redundância espacial ✗

Correto
Notas relativas a este envio: 1/1.

16
Notas: 1

Os códigos genericamente conhecidos como EDC ou ECC permitem que certas classes de erros em bits possam ser detectadas ou corrigidas. Assinale Verdadeiro ou Falso.

Aplicar ECC para registradores individuais pode requerer um quantidade significativa de overhead e aumentar o retardo do caminho crítico.	Verdadeiro
A proteção de memória por ECC é relativamente eficiente pois o custo da lógica de codificação pode ser amortizada por toda a memória	Verdadeiro
Códigos conhecidos como EDC resolvem todos os problemas de detecção e correção de bits isolados na memória e registradores.	Falso

Correto
Notas relativas a este envio: 1/1.

17
Notas: 1

Qual a vantagem do Chipkill em relação a códigos ECC para proteção de memória?

Escolher uma resposta.

- a. Chipkill se aplica eficientemente a registradores internos do microprocessador enquanto ECC só serve para memórias. ✗
- b. Chipkill é uma estratégia que isola (mata) um chip suspeito de estar com erros e, por isso, é mais eficiente que ECC uma vez que ECC ainda tenta corrigir um bit com erro, prejudicando o desempenho do sistema neste processo. ✗
- c. Chipkill pode tratar tratar múltiplos erros em um chip de memória, incluindo erros que afetam todos os bits de dados no chip, enquanto ECC só detecta e corrige erros em um bit. ✓
- d. Nenhuma vantagem, Chipkill é apenas mais um nome para códigos ECC. ✗

Correto
Notas relativas a este envio: 1/1.

18
Notas: 1

Um estudo da IBM com 10 mil servidores em um período de 3 anos determinou o número de paradas dos servidores devido a defeitos em memórias protegidas com paridade, ECC e Chipkill. Os resultados foram francamente favoráveis ao uso de Chipkill. Qual a razão que Iyer menciona como provável para esse fato?

Escolher uma resposta.

- a. 3 anos é um tempo de vida muito longo para um servidor e portanto os resultados apurados não são conclusivos ✗
- b. Memórias envelhecem muito rápido e é melhor matar os chips antes que envelheçam. ✗
- c. Erros que afetam múltiplos bits na memória constituem uma percentagem significativa dos erros de memória. ✓
- d. O estudo foi conduzido pela IBM e o código Chipkill é patenteado pelo IBM, os resultados podem ter sido forjados. ✗

Correto
Notas relativas a este envio: 1/1.

19
Notas: 1

Na época da publicação do artigo, as arquiteturas predominantes de microprocessadores eram: Intel P6, AMD Hammer, Intel Itanium, IBM S/390 G5 e IBM Power4. Naturalmente, após 5 anos vários outros chips apareceram derivados dessas arquiteturas, com inúmeras inovações e melhoramentos. Mas a tabela 2 do artigo nos fornece uma ótima visão dos recursos de tolerância a falhas aplicados nesses processadores.

Responda de acordo com a tabela, associando o recurso ao chip. Considere os registradores internos:

IBM Power4	paridade
AMD Hammer	sem proteção
Intel P6	paridade
IBM S/390 G5	ECC
Intel Itanium	sem proteção

Correto
Notas relativas a este envio: 1/1.

20
Notas: 1

Na época da publicação do artigo, microprocessadores predominantes eram: Intel P6, AMD Hammer, Intel Itanium, IBM S/390 G5 e IBM Power4. A tabela 2 do artigo nos fornece uma ótima visão dos recursos de tolerância a falhas

nesses processadores.

21
Notas: 1

Responda de acordo com a tabela, associando o recurso ao chip. Considere a cache de nível 1:

AMD Hammer, cache de dados	ECC
IBM S/390 G5	paridade
AMD Hammer, cache de instruçoes	paridade
Intel Itanium	paridade
IBM Power4	paridade
Intel P6	paridade

Correto
Notas relativas a este envio: 1/1.

Na época da publicação do artigo, microprocessadores predominantes eram: Intel P6, AMD Hammer, Intel Itanium, IBM S/390 G5 e IBM Power4. Responda de acordo com a tabela 2 do artigo, associando o recurso ao chip. Considere os barramentos internos:

Intel P6, barramento da cache de nível 2 a CPU	ECC
AMD Hammer	sem proteção
IBM Power4, barramento de endereços e controle	paridade
Intel Itanium	sem proteção
IBM Power4, barramento de dados	ECC
IBM S/390 G5	sem proteção

Correto
Notas relativas a este envio: 1/1.

22
Notas: 1

Na época da publicação do artigo, microprocessadores predominantes eram: Intel P6, AMD Hammer, Intel Itanium, IBM S/390 G5 e IBM Power4. Responda de acordo com a tabela 2 do artigo, indicando com **X** os microprocessadores que implementam internamente uma arquitetura de verificação da máquina - machine check architecture (MCA) - com o objetivo de detectar e corrigir erros na lógica do processador.

IBM S/390 G5	-
Intel Itanium	X
AMD Hammer	X
IBM Power4	-
Intel P6	X

Correto
Notas relativas a este envio: 1/1.

23
Notas: 1

Responda de acordo com a tabela 2 do artigo que lista as técnicas de tolerância a falhas usadas nas famílias de microprocessadores: Intel P6, AMD Hammer, Intel Itanium, IBM S/390 G5 e IBM Power4.

A velha estratégia de duplicação e comparação pode ser implementada internamente ao chip, por exemplo duplicando unidades de instrução e execução e comparando-as a cada clock, ou externamente ao chip, usando 2 chips idênticos em uma configuração mestre e escravo.

Assinale qual dos chips listados a seguir é construído usando duplicação e comparação interna:

Escolher uma resposta.

- a. Intel P6 ✗
- b. Intel Itanium ✗
- c. AMD Hammer ✗
- d. IBM S/390 G5 ✓
- e. IBM Power4 ✗

Correto
Notas relativas a este envio: 1/1.

Notas: 1

Responda de acordo com a tabela 2 do artigo que lista as técnicas de tolerância a falhas usadas nas famílias de microprocessadores populares em 2005.

A conhecida estratégia de duplicação e comparação pode ser implementada internamente ao chip, por exemplo duplicando unidades de instrução e execução e comparando-as a cada clock, ou externamente ao chip, usando 2 chips idênticos em uma configuração mestre e escravo.

Assinale qual dos chips listados a seguir é construído usando duplicação e comparação externa, também conhecida como FRC:

Escolher uma resposta.

- a. IBM S/390 G5 ✗
- b. Intel P6 ✓
- c. Intel Itanium ✗
- d. IBM Power4 ✗
- e. AMD Hammer ✗

Correto

Notas relativas a este envio: 1/1.

Fechar esta janela Fechar esta janela

Técnicas de proteção no nível arquitetural

Revisão da tentativa 1

Terminar revisão

Iniciado em	sábado, 9 outubro 2010, 22:13
Completado em	sábado, 9 outubro 2010, 22:21
Tempo empregado	8 minutos 21 segundos
Notas	17/18
Nota	**94.44** de um máximo de 100(**94**%)

1

Notas: 1

Localize-se no item "*Architectural error protection techniques*", na página 23 do artigo do Iyer e demais autores. Neste item é mencionado que vários trabalhos acadêmicos propuseram técnicas baseadas em redundância para tratar falhas transientes em microprocessadores.

A primeira técnica mencionada é SRTR (*simultaneously and redundantly threaded processors with recovery*). Como SRTR provê tolerância a falhas e recuperação?

Escolher uma resposta.

- a. Pela execução de duas cópias do mesmo programa em processadores com multithreading simultânea. ✓
- b. A técnica denominada SRTR não foi mencionada no artigo. ✗
- c. Executando o programa em uma thread de um processador com capacidade de multithreading e simultaneamente executando um programa especial capaz de verificar condições de erro do primeiro programa na outra thread do mesmo processador. ✗
- d. Pela execução de duas cópias do mesmo programa em dois processadores idênticos com capacidade de FRC. ✗
- e. Executando duas ou mais vezes o mesmo programa sequencialmente em um único processador com memória interna especial para armazenar os resultados intermediários. ✗

Correto

Notas relativas a este envio: 1/1.

2

Notas: 1

Ainda no item "*Architectural error protection techniques*", do artigo do Iyer e demais autores, a segunda técnica mencionada é DIVA (Dynamic Instruction Verification Architecture). Como esta técnica provê tolerância a falhas?

Escolher uma resposta.

- a. Pela execução de duas cópias do mesmo programa em dois processadores idênticos com capacidade de FRC. ✗
- b. Um processador extra, simples e com execução seguindo o ordenamento natural de instruções é integrado na mesma pastilha do processador complexo com a função de verificar suas saídas e disparar uma recuperação quando descobrir uma inconsistência. ✓
- c. A técnica DIVA não foi mencionada no artigo. ✗
- d. Um processador verificador externo ao chip atua sobre o processador complexo super-escalar forçando uma ação de reconfiguração sempre que descobrir uma inconsistência na saída. ✗
- e. Pela execução de duas cópias do mesmo programa em processadores com multithreading simultânea. ✗

Correto

Notas relativas a este envio: 1/1.

3

Notas: 1

Considere a tabela sobre técnicas de confiabilidade no nível arquitetural mostrada na página 24 (tabela 3) do artigo do Iyer. Quanto aos méritos listados abaixo, assinale se correspondem a DIVA, a SRTR, a ambas ou a nenhuma delas.

Utiliza hardware estepe (redundante) em processadores multithread.	SRTR
Redundância suprida pela técnica serve para aumentar o desempenho.	nenhum
Se o verificador for simples, pode ser verificado formalmente.	DIVA

Resultados intermediários ficam disponíveis no circuito para detecção de erros.	ambas	Correto

Notas relativas a este envio: 1/1.

4

Notas: 1

Considere a tabela sobre técnicas de confiabilidade no nível arquitetural mostrada na página 24 (tabela 3) do artigo do Iyer. Quanto às desvantagens listadas abaixo, assinale se correspondem a DIVA, a SRTR, a ambas ou a nenhuma delas.

Projetar um verificador para um core superescalar não é uma tarefa trivial.	DIVA
O ciclo de execução fica aumentado.	SRTR
Detecção de erros é limitada a esfera de replicação que inclui apenas o estágio de execução de instruções.	ambas
A aplicação não pode usar diretamente o hardware redundante (seja o verificador ou a outra thread).	nenhuma

Parcialmente correta

Notas relativas a este envio: 0.5/1.

5

Notas: 1

Examine a tabela sobre técnicas de confiabilidade no nível arquitetural mostrada na página 24 (tabela 3). Além de DIVA e SRTR são apresentadas duas novas técnicas (instruções duplicadas em processadores superescalares; uso dual de caminhos de dados superescalares).

Considerando essas duas novas técnicas, assinale com X qual (ou quais) delas apresenta(m) uma estratégia de redundância que pode ser aplicada em microprocessadores COTS.

ambas	-
uso dual de caminhos de dados superescalares	-
instruções duplicadas em processadores superescalares	X
nenhuma	-

Correto

Notas relativas a este envio: 1/1.

6

Notas: 1

Considere a tabela 3, sobre técnicas de confiabilidade no nível arquitetural. Além de DIVA e SRTR são apresentadas duas outras técnicas. Ambas são baseadas na duplicação de instruções. Assinale em qual (ou em quais) essa duplicação ocorre por software e/ou por hardware.

uso dual de caminhos de dados superescalares	por software e por hardware
instruções duplicadas em processadores superescalares	software (pelo compilador)

Parcialmente correta

Notas relativas a este envio: 0.5/1.

7

Notas: 1

Considere a tabela sobre técnicas de confiabilidade no nível arquitetural (tabela 3). Examine as duas técnicas ("instruções duplicadas em processadores superescalares" e "uso dual de caminhos de dados superescalares") para responder essa questão. Assinale com X qual (ou quais) apresenta(m) como desvantagem uma grande penalidade em desempenho e uma grande latência de detecção de erro.

nenhuma apresenta tal desvantagem	-
ambas	-
instruções duplicadas em processadores superescalares	X
uso dual de caminhos de dados superescalares	-

Correto

Notas relativas a este envio: 1/1.

8

Notas: 1

Considere ainda todas as quatro técnicas mencionadas na tabela 3. Associe uma categoria de redundância a cada uma das técnicas, seguindo a opinião dos autores expressa no artigo.

DIVA	redundância de hardware
instruções duplicadas em processadores escalares	redundância temporal em software
SRTR	redundância temporal em hardware

uso dual de caminho de dados superescalar	redundância de hardware e temporal	Correto

Notas relativas a este envio: 1/1.

9

Notas: 1

Localize-se no item: verificação em tempo de execução consciente da aplicação (página 23). Os autores afirmam que as técnicas baseadas em hardware para proteção contra erros apresentam problemas. Assinale quais são esses problemas de acordo com o texto:

Escolher uma resposta.

- a. baixa cobertura de falhas pelos mecanismos em hardware e ausência de cobertura de bugs do software ✗
- b. perda de controle na ordem de execução das instruções e baixa cobertura de falhas permanentes ✗
- c. alto custo em termos de área ocupada, tempo gasto no processamento e energia consumida ✓
- d. interferência na semântica da aplicação e alta latência na sinalização do erro ✗

Correto

Notas relativas a este envio: 1/1.

10

Notas: 1

Localize-se no item: verificação em tempo de execução consciente da aplicação. Considere a duplicação de software para fins de detecção e recuperação de erros em duas threads de hardware em processadores multithread. Assinale verdadeiro ou falso segundo as afirmações dos autores:

o desempenho do sistema é penalizado pelo retardo no clock provocado pela hardware adicional	Verdadeiro
o desempenho aumenta pois o software está sendo executado duas vezes mais rápido nas threads paralelas	Falso
o desempenho do sistema é penalizado pela necessidade de sincronizar threads duplicadas	Verdadeiro

Correto

Notas relativas a este envio: 1/1.

11

Notas: 1

Considere detecção e recuperação de erros através do software. Assinale verdadeiro ou falso segundo as afirmações dos autores:

o sistema é penalizado pela alta latência de detecção de erros	Verdadeiro
a detecção de erros implementada em software ou hardware apresenta latências de detecção equivalentes	Falso
a complexidade das aplicações é maior em sistemas que usam soluções de detecção em software	Verdadeiro

Correto

Notas relativas a este envio: 1/1.

12

Notas: 1

Um dos focos do artigo e a base para as propostas dos autores discutidas no final do artigo é o fato de que as técnicas de tolerância a falhas implementadas no nível arquitetural não são conscientes da aplicação executando no sistema. Considerando a opinião dos autores expressa no texto, quais seriam as consequências desta falta de consciência (assinale verdadeiro ou falso):

a falta de consciência é inconveniente pois o processador não se adapta para cumprir as necessidades de dependabilidade da aplicação	Verdadeiro
a falta de consciência é indesejável pois a aplicação não pode selecionar as técnicas disponíveis de TF mais adequadas as suas necessidades	Verdadeiro
a falta de consciência é benéfica pois torna as estrategias de tolerância a falhas do microprocessador transparentes para a aplicação	Falso

Correto

Notas relativas a este envio: 1/1.

13

Notas: 1

Os autores afirmam que 3 estudos de injeção de falhas nos níveis inferiores do hardware em diferentes processadores mostraram que apenas 20% a 4% dessas falhas se manifestaram nos níveis superiores (nível de pinos ou nível de software) como defeito.

Resposta: Verdadeiro ✓

Falso ✗

Correto

Notas relativas a este envio: 1/1.

14

Notas: 1

Sobre o significado das expressões: *chip-level replication* e *on-chip replication* usada por Iyer podemos afirmar que:

Escolher uma resposta.

- a. chip-level replication significa replicação de hardware enquanto on-chip replication significa replicação do software e execução sequencial (redundância temporal) ✗
- b. os termos são sinônimos e significam que a replicação é feita no hardware ✗
- c. chip-level replication implica no uso 2 ou mais chips e on-chip replication significa replicação de recursos de hardware interna ao chip ✓
- d. chip-level replication implica em posicionar dois ou mais chips um ao lado do outro enquanto on-chip replication significa posicionar um ou mais chips um sobre o outro economizando espaço na placa ✗

Correto

Notas relativas a este envio: 1/1.

15

Notas: 1

De acordo com Iyer e demais autores, em relação ao processador do sistema HP NonStop Himalaya, é correto afirmar que:

chip-level duplication mostrou maior nível de desempenho do que on-chip replication	Falso
chip-level duplication mostrou menor nível de dependabilidade e desempenho do que on-chip replication	Falso
chip-level duplication mostrou maior nível de dependabilidade do que on-chip replication	Verdadeiro

Correto

Notas relativas a este envio: 1/1.

16

Notas: 1

De acordo com Iyer, sistemas IBM como o G5 usam um processador backup *(chip-level replication)* em adição a *on-chip replication* das unidades instrução e de execução.

Resposta:

- Verdadeiro ✓
- Falso ✗

Correto

Notas relativas a este envio: 1/1.

17

Notas: 1

Segundo Iyer, uma das razões pelas quais *chip-level duplication* mostra maiores níveis de dependabilidade do que *on-chip replication* é relacionado ao fato que a maioria dos erros no nível do processador não provoca erros na aplicação.

Resposta:

- Verdadeiro ✓
- Falso ✗

Correto

Notas relativas a este envio: 1/1.

18

Notas: 1

Segundo Iyer, estudos experimentais em caracterização quantitativa comprovam os níveis de confiabilidade que os processadores podem alcançar com *on-chip replication.*

Resposta:

- Verdadeiro ✗
- Falso ✓

Correto

Notas relativas a este envio: 1/1.

Terminar revisão

Segurança funcional crítica

Revisão da tentativa 1

Terminar revisão

Iniciado em	sábado, 9 outubro 2010, 22:22
Completado em	sábado, 9 outubro 2010, 22:28
Tempo empregado	5 minutos 58 segundos
Notas	24/24
Nota	**100** de um máximo de 100(**100**%)

1

Notas: 1

William Dunn, no artigo Designing Safety-Critical Computer Systems, publicado na Computer em 2003, cita várias exemplos de aplicações de segurança crítica. Assinale com X aplicações de segurança crítica citadas pelo autor.

sistemas de suporte a vida em hospitais	X
controles de vôo de aeronaves	X
processamento químico e de petróleo	X
base de dados de hóspedes em um hotel	-
editores de texto	-
controle de mercadorias em um supermercado	-

Correto

Notas relativas a este envio: 1/1.

2

Notas: 1

Segundo William R. Dunn no artigo Designing Safety-Critical Computer Systems, publicado na Computer em 2003, qual é a principal preocupação relacionada ao uso de sistemas baseados em computadores?

Escolher uma resposta.

- A. A sua ubiquidade. ✗
- B. A ineficiência destes sistemas, se comparados a sistemas eletro-mecânicos tradicionais. ✗
- C. A vida útil do sistema, que por força do constante avanço da tecnologia, tente a encurtar muito rapidamente. ✗
- D. O custo inerente à manutenção destes sistemas. ✗
- E. Eles podem eventualmente falhar e causar grandes danos, como já foi observado no passado. ✓

Correto

Notas relativas a este envio: 1/1.

3

Notas: 1

Dunn cita um grave acidente que ocorreu devido ao uso do Therac25. O que era o Therac25?

Escolher uma resposta.

- A. Era um sistema de computador terapêutico destinado a curar pacientes, mas devido à falta de evidências de sua eficácia, foi forçado a sair do mercado. ✗
- B. Era um sistema de gestão hospitalar muito popular nos anos 80, principalmente nos Estados Unidos. ✗
- C. Era um sistema de computador terapêutico destinado a curar pacientes, mas que inadvertidamente matou e mutilou muitas pessoas antes de ser forçado a sair do mercado. ✓
- D. Era um processador tolerante a falhas especialmente projetado para sistemas terapêuticos, que ajudou na evolução da medicina computacional. ✗

Correto

Notas relativas a este envio: 1/1.

4

Notas: 1

Na prática, muitos conceitos e arquiteturas vistas como seguras por seus desenvolvedores falharam, colocando pessoas, propriedades ou o ambiente em risco. Segundo William Dunn, isto aconteceu principalmente porque seus autores ou usuários... (marque as 3 razões principais):

menos uma resposta.

- ☑ A. ignoraram pontos únicos de falha, que fizeram o conceito de segurança irreal quando posto em prática. ✓
- ☐ B. não estavam dispostos a pagar o custo adicional relacionado aos conceitos e estruturas de segurança ✗
- ☐ C. não sabiam implementar as possíveis soluções seguras adequadas às suas necessidades. ✗
- ☑ D. esqueceram de considerar o sistema principal no qual o conceito de segurança a ser aplicado era para ser incorporado. ✓
- ☐ E. não se preocuparam em monitorar as possíveis falhas. ✗
- ☑ F. tinham uma compreensão incompleta do que faz um sistema ser "seguro". ✓

5

Notas: 1

Correto

Notas relativas a este envio: 1/1.

Segundo o William Dunn, rever as definições e conceitos fundamentais de segurança de sistemas já implementados nos fornece uma estrutura para tratar os problemas relacionados à área. O que pode auxiliar, segundo o autor, a verificar se o projeto destes sistemas será seguro?

Escolher uma resposta.

- ○ A. Testar continuadamente por um longo período os sistemas, antes de coloca-los em funcionamento. ✗
- ○ B. Controlar sua temperatura em períodos longos de funcionamento. ✗
- ○ C. Utilizar técnicas de mapeamento dinâmico de pontos críticos no projeto quando este está em uso. ✗
- ◉ D. Explorar o projeto sistemático de sistemas de computadores seguros. ✓

Correto

Notas relativas a este envio: 1/1.

6

Notas: 1

Localize-se no item "definindo segurança". Qual é o significado do termo *mishap* segundo o Departamento de Defesa dos EUA?

Escolher uma resposta.

- ○ A. Consequência de um acidente ou infortúnio imprevisto. ✗
- ○ B. Evento resultante diretamente de uma falha humana, que resulta em morte, lesão, doença ocupacional, danos e perda de equipamento ou bens, ou danos ao meio ambiente. ✗
- ○ C. Gravidade de um acidente determinado em valores financeiros. ✗
- ◉ D. Evento imprevisto ou uma série de eventos que resultam em morte, lesão, doença ocupacional, danos e perda de equipamentos ou bens, ou danos ao meio ambiente. ✓

Correto

Notas relativas a este envio: 1/1.

7

Notas: 1

Qual é o significado de *mishap risk* (risco de acidente) segundo Willian Dunn?

Escolher uma resposta.

- ○ A. É uma avaliação da probabilidade de ocorrência de um acidente desconsiderando a sua severidade. ✗
- ○ B. É a avaliação do impacto de um acidente em termos de duas preocupações principais: danos a vidas humanas e danos financeiros. ✗
- ◉ C. É a avaliação do impacto de um acidente em termos de duas preocupações principais: a sua gravidade potencial e a probabilidade de sua ocorrência. ✓
- ○ D. É a avaliação do risco estatístico de perda financeira relacionado a ocorrência de um acidente. ✗

Correto

Notas relativas a este envio: 1/1.

8

Notas: 1

Considerando as ideias e conceitos expressas por William Dunn, responda se as sentenças são verdadeiras ou falsas.

Em algumas aplicações, a redução de riscos de acidentes domina completamente os custos do sistema.	Verdadeiro
Sistemas tais como automóveis, aviões e as usinas nucleares não são absolutamente (ou seja, 100%) seguros.	Verdadeiro

Dada a atual tecnologia, é possível eliminar os riscos de acidentes em sistemas de segurança críticos.	Falso
Na criação de um sistema seguro, minimizar os custos de redução de riscos de acidentes (mishap risks) nos obriga a um comprometimento do sistema, na medida em que gastamos recursos. Mas pode-se fazer isto somente até um nível considerado aceitável em geral.	Verdadeiro

Correto
Notas relativas a este envio: 1/1.

9
Notas: 1

Localize-se no item "riscos aceitáveis de acidentes". O que é considerado um risco aceitável para um determinado tipo de acidente por parte do grande público, se acordo com Willian Dunn?

Escolher uma resposta.

- A. Risco aceitável para um acidente é determinado pela ausência de acidentes daquele tipo até o momento. ✗
- B. Só é considerado aceitável um risco zero de acidentes. ✗
- C. O grande público não se preocupa com aspectos técnicos e mostra disposição de aceitar qualquer risco. ✗
- D. Basicamente, enquanto os acidentes ocorrem muito infrequentemente seu risco é considerado aceitável. ✓
- E. Enquanto há um risco, este nunca será considerado aceitável. ✗

Correto
Notas relativas a este envio: 1/1.

10
Notas: 1

Segundo as estatísticas, qual é a taxa aceitável de risco de acidentes comuns?

Escolher uma resposta.

- 10^-2 a 10^-10 acidentes por dia. ✗
- 10^2 a 10^10 acidentes por hora. ✗
- Não existem dados considerados confiáveis para determinar tal taxa de acidentes. ✗
- 10^-2 a 10^-10 acidentes por hora. ✓

Correto
Notas relativas a este envio: 1/1.

11
Notas: 1

Dunn afirma que apesar dos resultados trágicos dos acidentes de trânsito, a maioria das pessoas se sente segura dirigindo um carro. Qual seria a razão para essa sensação de segurança?

Escolher uma resposta.

- a. a falta de formação tecnológica das pessoas ✗
- b. a raridade da ocorrência de acidentes ✓
- c. a desinformação da maioria das pessoas a respeito da frequência dos acidentes de carro ✗
- d. ao custo de um acidente de carro ser relativamente baixo comparado com um acidente na aviação ✗

Correto
Notas relativas a este envio: 1/1.

12
Notas: 1

Considerando o conceito de risco aceitável, um projeto com uma chance de 10% de acidente catastrófico por hora de operação é um sistema seguro?

Escolher uma resposta.

- A. Depende do propósito da aplicação. ✗
- B. Sim, se está dentro do planejado pelos projetistas. ✗
- C. Não, deve ser no mínimo de 1%. ✗
- D. Não, é uma taxa de acidentes altíssima. ✓

Correto
Notas relativas a este envio: 1/1.

13
Notas: 1

Quem decide o que é um risco aceitável?

Escolher uma

- A. os consumidores de produtos tecnológicos ✗

B. o governo americano e a IEC através de padrões internacionais ✗

C. a própria empresa que implementa e comercializa a solução segura ✗

D. as associações industriais, sociedades profissionais de segurança e institutos relacionados a segurança que assumem o consenso do público em geral sobre riscos aceitáveis ✓

E. os projetistas de sistemas baseados no seu conhecimento técnico e julgamento ético ✗

Correto
Notas relativas a este envio: 1/1.

14
Notas: 1

Mil-STD-882 D e IEC 61508 são:

Escolher uma resposta.

a. relatos de acidentes governamentais e industriais repectivamente ✗

b. normas de segurança ✓

c. leis internacionais que regulam aspectos de segurança ✗

d. textos motivacionais para alertar governos e indústrias acerca dos cuidados com segurança ✗

Correto
Notas relativas a este envio: 1/1.

15
Notas: 1

Tipicamente qualquer computador, seja um controlador de vôo de uma aeronave, um robô industrial ou uma máquina de radiação terapêutica, é composto, segundo Dunn, de cinco componentes primários. Relacione os componentes primários de um sistema de computação com sua função.

Entidade física que o sistema monitora e controla, muitas vezes chamada de planta ou processo.	Aplicação
Componente que converte uma medida física em um sinal elétrico correspondente para a entrada no computador.	Sensor
Componente que converte um sinal elétrico de saída do computador para uma ação física correspondente que controla a função de um aplicativo.	Atuador
Humano responsável por monitorar e ativar em tempo real o sistema	Operador
Composto de hardware e software que monitora e controla a aplicação em tempo real.	Computador

Correto
Notas relativas a este envio: 1/1.

16
Notas: 1

Dunn cita alguns exemplos para componentes de um sistema de computação. Associe os exemplos aos nomes dos componentes.

acelerômetro	sensor
motor	atuador
válvula	atuador
controlador lógico programável	computador
transdutor de pressão	sensor

Correto
Notas relativas a este envio: 1/1.

17
Notas: 1

Dunn cita alguns modelos de computador que monitoram e controlam aplicações em tempo real. Assinale entre os modelos abaixo um exemplo que não forma um sistema de computação.

Escolher uma resposta.

A. CLP - Controlador lógico-programável ✗

B. Computador de bordo em aviões. ✗

C. Controlador em placa única. ✗

D. Instrumentos de campo. ✗

E. Conjunto de sensores. ✓

Correto
Notas relativas a este envio: 1/1.

18
Notas: 1

Localize-se no item "análise de hazards" do artigo de Willian Dunn. O que se pode afirmar quanto a sistemas básicos de computadores?

Escolher uma resposta.

- A. É impossível ter um sistema básico que apresente uma taxa de risco de acidentes em um nível aceitável ✗
- B. Com a tecnologia atual, sistemas básicos já são suficientemente seguros para qualquer tipo de aplicação. ✗
- C. Projetos seguros são geralmente muito complexos, e estes elementos básicos que compõe o sistema têm pouca ou nenhuma influência na taxa de risco de acidentes. ✗
- D. Como sistemas básicos não empregam recursos de segurança, provavelmente irão apresentar um nível muito elevado de risco de acidente. ✓

Correto
Notas relativas a este envio: 1/1.

19
Notas: 1

Quando um sistema de computação básico apresentar um nível inaceitável de risco de acidente, qual a solução apontada por Dunn.

Escolher uma resposta.

- a. A solução exige encontrar uma solução que não necessite o uso de computadores. ✗
- b. A solução exige modificar a aplicação para que ela não exiba um nível tão alto de risco. ✗
- c. A solução exige adotar uma rede complexa de sistemas básicos. ✗
- d. A solução exige modificar o operador, os sensores, os atuadores e o computador para criar um novo sistema que estará dentro do nível aceitável de risco. ✓

Correto
Notas relativas a este envio: 1/1.

20
Notas: 1

Quando um sistema básico exibe um nível inaceitável de risco, o projeto de uma solução inicia com qual questão?

Escolher uma resposta.

- A. Como pode este sistema de computação básico apresentar defeito e causar um acidente? ✓
- B. Quais as consequências de um possível acidente causado por este sistema básico? ✗
- C. Quão custosa será a solução para o problema? ✗
- D. De que maneira é possível minimizar os danos em caso de acidente por falha no sistema? ✗

Correto
Notas relativas a este envio: 1/1.

21
Notas: 1

O elemento chave que conecta um defeito em um sistema básico de computação a um acidente subsequente é chamado de:

Escolher uma resposta.

- a. hazard ✓
- b. failure ✗
- c. mishap ✗
- d. error ✗
- e. mishap risk ✗

Correto
Notas relativas a este envio: 1/1.

22
Notas: 1

De acordo com Willian Dunn, no contexto de segurança crítica, *hazard* é qualquer condição real ou potencial que pode causar:

Escolher uma resposta.

- A. erros não detectáveis no equipamento; falhas difíceis de serem localizadas, isoladas e/ou compensadas; alta latência de falha; defeitos de causa comum ✗
- B. perdas financeiras; danos a propriedade; perdas de equipamentos e de credibilidade no mercado de prestação de serviços ✗

C. insatisfação dos operadores de um sistema básico de segurança; stress ocupacional; falhas humanas involuntárias provocadas por fadiga do operador ✗

D. lesões, doença ou morte de pessoas; danos ao meio ambiente; danos a ou perdas de sistemas, equipamentos ou propriedade ✓

Correto
Notas relativas a este envio: 1/1.

23
Notas: 1

Segundo o autor, são exemplos de *hazards* contidos na aplicação: (escolha 3 exemplos)

Escolha pelo menos uma resposta.

- ☐ A. bit-flip na memória do sistema básico de computação ✗
- ☐ B. leitura incorreta de informação de temperatura ✗
- ☑ C. perda de controle de vôo ✓
- ☑ D. esfriamento do núcleo em uma central nuclear ✓
- ☑ E. presença de material tóxico ou gás natural ✓

Correto
Notas relativas a este envio: 1/1.

24
Notas: 1

para finalizar a leitura do item "análise de hazards", enumere a sequência de ações a serem tomadas para um projeto de um sistema seguro.

Análise do operador, sensor, computador e atuador.	2
Análise de modos de defeitos de cada componente, para determinar todas as suas fontes de defeitos incluindo defeitos de hardware, problemas de fabricação, falhas de programação, stress ambiental, erros de projeto e erros na manutenção.	3
Com base nas informações coletadas, estabelecimento de uma ligação entre todos os modos possíveis de defeitos e os acidentes.	4
Inicio do projeto real, levando em consideração as análises anteriores.	5
Identificação dos hazards na aplicação do sistema.	1

Correto
Notas relativas a este envio: 1/1.

Terminar revisão

1

Notas: 1

Localize-se no item "um exemplo simples" no artigo de William Dunn. No exemplo apresentado, um sistema de aquecimento de água, identifique o que é considerado *hazard*.

Escolher uma resposta.

- a. a explosão do tanque ✗
- b. o sobreaquecimento da água ✓
- c. as queimaduras provocadas em um operador humano ✗
- d. o sensor de temperatura estragado ✗
- e. a unidade de aquecimento que permanece permanentemente ligada ✗

Correto

Notas relativas a este envio: 1/1.

2

Notas: 1

No exemplo simples, assinale *hazard* (perigo), *mishap* (acidente, sinistro) e defeito para os eventos descritos abaixo:

a interface de hardware do computador comanda permanentemente "manter aquecimento" para a unidade de aquecimento de água	defeito
sensor de temperatura marcando temperatura abaixo da temperatura real da água	defeito
a explosão do tanque de água causada pelo sobreaquecimento da água além do ponto de ebulição	mishap
sobreaquecimento da água	hazard
operador sofre queimaduras por ter aberto uma saída de água estando a água sobreaquecida	mishap

Correto

Notas relativas a este envio: 1/1.

3

Notas: 1

Considerando que um sistema pode apresentar um alto risco de acidente (*mishap*), Dunn cita 3 formas de diminuir o risco. Assinale as 3 formas:

Escolha pelo menos uma resposta.

- a. usar ferramentas de desenvolvimento certificadas ✗
- b. aumentar a qualidade e a confiabilidade dos componentes ✓
- c. escolher uma boa equipe de desenvolvedores ✗
- d. incorporar dispositivos de segurança internos ✓
- e. incorporar dispositivos de segurança externos ✓

Correto

Notas relativas a este envio: 1/1.

4

Notas: 1

Melhorar a confiabilidade visa a:

Escolher uma resposta.

- a. aumentar a vida útil de um componente, o que, por sua vez, aumenta a vida útil do sistema e com isso reduz os custos associados à manutenção ✗
- b. zerar a probabilidade de ocorrência de defeitos ✗
- c. eliminar as fontes que potencialmente causam defeitos de componentes ✗
- d. reduzir a probabilidade de defeito de componente, o que, por sua vez, reduz a probabilidade de acidente (mishap) ✓

Correto

Notas relativas a este envio: 1/1.

5

Notas: 1

De acordo com Dunn, existe uma abordagem muito usada e eficiente para aumentar a confiabilidade de um sistema. Determine qual seria essa abordagem:

Escolher uma resposta.

- a. evitar problemas que resultam de condições ambientais impróprias ✗
- b. reprojeto ✗
- c. aplicação de medidas que aumentem a qualidade do sistema ✗

d. uso de componentes redundantes de software e hardware ✓

e. certificação do sistema pela IEC 61508 ✗

Correto
Notas relativas a este envio: 1/1.

6
Notas: 1

No artigo são mencionadas fontes responsáveis por defeito de componentes que são, nas palavras do autor, mais evasivas, como por exemplo: erros humanos, inadequação de projeto e deficiências nos procedimentos. A norma IEC 61508 inclui essas fontes de defeitos em uma categoria geral descrita como:

Escolher uma resposta.

a. defeitos evasivos ✗

b. defeitos sistemáticos ✓

c. defeitos elusivos ✗

d. defeitos randômicos ✗

e. defeitos maliciosos ✗

Correto
Notas relativas a este envio: 1/1.

7
Notas: 1

No artigo são mencionadas fontes de defeito de componentes mais evasivas, como: erros humanos, inadequação de projeto e deficiências nos procedimentos. Para evitá-los ou eliminá-los, a norma IEC 61508 recomenda:

Escolher uma resposta.

a. abordagens orientadas a redundância ✗

b. abordagens baseadas em automação de projeto sem interferência humana ✗

c. evitar usar componentes de software ✗

d. abordagens orientadas a qualidade ✓

e. reprojeto do sistema ✗

Correto
Notas relativas a este envio: 1/1.

8
Notas: 1

Segundo Willian Dunn, quando são necessários dispositivos internos de segurança?

Escolher uma resposta.

a. quando se deseja reduzir os custos além de reduzir os riscos ✗

b. quando o projeto requer passos adicionais para reduzir os riscos abaixo do nível de risco alcançado pelas medidas de qualidade e confiabilidade ✓

c. quando o projeto foi mal especificado e ninguém conhece como ele vai se comportar em operação real ✗

d. quando redundância se torna economicamente inviável ✗

e. quando o projeto usou componentes de baixa confiabilidade ✗

Correto
Notas relativas a este envio: 1/1.

9
Notas: 1

Segundo Willian Dunn, os dispositivos internos de segurança, além de reduzir os efeitos de falhas de hardware e software, também fornecem:

Escolher uma resposta.

a. uma solução de baixo custo para evitar acidentes ✗

b. uma barreira que evita a ocorrência de falhas de software e hardware ✗

c. uma barreira contra defeitos sistemáticos ✓

d. uma barreira contra operadores humanos ✗

Correto
Notas relativas a este envio: 1/1.

10
Notas: 1

A última linha de defesa contra defeitos que podem resultar em acidentes (*mishaps*) é provida por dispositivos que atuam quando a aplicação experimenta um evento perigoso. Esses dispositivos são conhecidos por:

Escolher uma resposta.

a. dispositivos de alta confiabilidade ✗

b. dispositivos redundantes ✗

c. dispositivos especializados em segurança ✗

d. dispositivos externos de segurança ✓

e. dispositos internos de segurança ✗

Correto
Notas relativas a este envio: 1/1.

11
Notas: 1

Para alcançar efetiva redução de risco de acidentes, os projetistas costumam:

Escolher uma resposta.

a. usar as 3 formas de redução de risco de forma balanceada e evitando negligenciar alguma parte do sistema ✓

b. enfatizar dispositivos externos de segurança pois são mais baratos e eficientes que os demais recursos de segurança ✗

c. enfatizar dispositivos internos de segurança pois são mais baratos e eficientes que os demais recursos de segurança ✗

d. enfatizar qualidade e confiabilidade dos componentes de hardware e software ✗

e. evitar defeitos sistemáticos ✗

Correto
Notas relativas a este envio: 1/1.

12
Notas: 1

Considerando o exemplo do aquecedor de água, apresentado por Dunn, associe o recurso usado no exemplo com uma das 3 estratégias de redução de riscos:
1 - melhorar a confiabilidade e confiabilidade dos componentes
2 - incorporar dispositivos internos de segurança
3 - incorporar dispositivos externos de segurança

válvula de alívio de temperatura e pressão	3
componentes com confiabilidade superior aos usados em equipamentos convencionais	1
uso de uma chave de limitação de alta temperatura	2
programa que detecta defeito em vários componentes e então atua sobre uma chave de potência para desligar o equipamento	2
cuidados para garantir integridade estrutural do tanque de água	1

Correto
Notas relativas a este envio: 1/1.

13
Notas: 1

A figura 4 do artigo do Dunn mostra dispositivos adicionais de segurança e algumas técnicas usados para redução de risco encontradas em aplicações da vida-real. Associe o nome do recurso à sua forma de atuação:

inibe as saídas do atuador (effector) forçando o sistema para um estado seguro	circuito de parada de emergência
inibe ação do atuador (effector) a menos que alguma condição física externa seja satisfeita	intertravamento
detecta defeitos em sensores individuais	redundância de informação
detecta defeitos em atuadores realimentando a saída do atuador no computador para verificar se a saída corresponde ao comando enviado	wraparound
inibe a ação do atuador (effector) se o computador (ou o software) apresentar defeito e parar de enviar pulsos indicando que está em operação	watchdog timer

Correto
Notas relativas a este envio: 1/1.

14
Notas: 1

Considere os conceitos de sistemas *fail-safe* e sistemas *fail-operate*. Associe:

sistemas que para se manter em operação tolerando falhas aplicam redundância, como por exemplo, backups ou replicação de componentes	fail-operate
sistemas que apresentam um estado seguro, geralmente não operacional, que pode ser alcançado modificando as saídas de atuadores (effectors) após detecção de falha	fail-safe

sistemas que devem permanecer em operação mesmo após detecção de falha em um ou mais de seus componentes	fail-operate	

Correto
Notas relativas a este envio: 1/1.

15
Notas: 1

Considere os conceitos de sistemas *fail-safe* e sistemas *fail-operate*. Associe os exemplos ao termo:

a maioria das aplicações de controle da vida-real	fail-operate
controle de uma ferramenta industrial de corte (guilhotina)	fail-safe
sistemas de controle fly-by-wire	fail-operate

Parcialmente correta
Notas relativas a este envio: 0.67/1.

16
Notas: 1

Relacionado ao texto sobre sistemas *fail-operate*, assinale verdadeiro e falso de acordo com as afirmações do autor.

Dunn afirma que sistemas fail-operate usam duas abordagens de tolerância a falhas: backup e replicação de componentes.	Verdadeiro
O uso de backups evita a degradação de desempenho em sistemas fail-operate, como no caso do Airbus A320.	Falso
Redundância de componentes é um conceito simples de fácil implementação.	Falso
Os processos de detecção de falhas, isolamento e reconfiguração podem se tornar complexos aumentando muito os custos de desenvolvimento.	Verdadeiro

Correto
Notas relativas a este envio: 1/1.

17
Notas: 1

Dunn cita três técnicas analíticas que são usadas para determinar se os procedimentos aplicados para reduzir o risco de acidentes em um sistema alcançaram o nível de risco adequado. Marque com X as siglas que correspondem a essas três técnicas analíticas.

SIL	-
MIL	-
RA	X
FMEA	X
FTA	X
IEC	-

Correto
Notas relativas a este envio: 1/1.

18
Notas: 1

O projetista examina cada componente no sistema, considera como o componente pode apresentar defeito e então determina o efeito que cada defeito de componente pode provocar no sistema. O nome da técnica analítica sendo empregada é:

Escolher uma resposta.

- a. RA ✗
- b. FMEA ✓
- c. FTA ✗
- d. SIL ✗
- e. cadeia de Markov ✗
- f. IEC ✗

Correto
Notas relativas a este envio: 1/1.

19
Notas: 1

A técnica analítica conhecida por análise de modos de defeito e efeitos procura principalmente verificar se existe algum ponto único de defeito no sistema que possa anular os benefícios da aplicação dos procedimentos de redução de riscos.

Correto
Notas relativas a este envio: 1/1.

20

Notas: 1

O projetista inicia com um acidente (*mishap*) identificável e trabalha retroativamente identificando todos os componentes que podem causar tal acidente e todos os dispositivos de segurança que podem evitá-lo. O nome da técnica analítica sendo empregada é:

Escolher uma resposta.

- a. FMEA ✗
- b. IEC ✗
- c. SIL ✗
- d. cadeia de Markov ✗
- e. RA ✗
- f. FTA ✓

Correto

Notas relativas a este envio: 1/1.

21

Notas: 1

Entre os técnicas analíticas citadas por Dunn, identifique os métodos qualitativos e os quantitativos.

RA	quantitativo
FMEA	qualitativo
FTA	qualitativo

Correto

Notas relativas a este envio: 1/1.

22

Notas: 1

O analista deve determinar a probabilidade de defeito de cada componente na árvore de falhas. Os componentes são: de hardware, de software e operador. O resultado é numérico e geralmente fornecido como probabilidade de defeito por hora. O nome da técnica analítica sendo empregada é:

Escolher uma resposta.

- a. FMEA ✗
- b. SIL ✗
- c. cadeia de Markov ✗
- d. IEC ✗
- e. RA ✓
- f. FTA ✗

Correto

Notas relativas a este envio: 1/1.

23

Notas: 1

Assinale verdadeiro ou falso:

Dunn afirma que é óbvio que as preocupações do projetista sobre a segurança funcional (safety) de um sistema crítico se encerram quando termina o projeto e a implementação do sistema.	Falso
Dunn afirma que um acompanhamento rigoroso deve ser mantido durante toda a vida operacional de um sistema crítico para garantir que o risco de acidentes permaneça no, ou fique abaixo do, nível de risco obtido no projeto original.	Verdadeiro
Para alcançar redução de risco, é exigido que todos os componentes de um sistema sejam considerados: hardware e software, sensores, atuadores, operador e a aplicação.	Verdadeiro

Correto

Notas relativas a este envio: 1/1.

Fechar esta janela | Fechar esta janela

1

Notas: 1

Ken Birman e demais autores no artigo (How the Hidden Hand Shapes the Market for Software Reliability) defendem:

Escolher uma resposta.

- a. que produtos visando confiabilidade falharam em ser bem sucedidos no mercado devido a deficiências dos próprios produtos ✗
- b. que os problemas de confiabilidade, que são endêmicos em sistemas distribuídos modernos, são culpa dos vendedores desses sistemas ✗
- c. que os problemas de confiabilidade em sistemas distribuídos modernos se devem muito mais a problemas de mercado do que aos produtos e vendedores ✓
- d. que Adam Smith já havia previsto que o mercado de comercialização de software iria sofrer prejuízos na primeira década do novo milênio devido a pirataria (hidden hand) ✗
- e. que os usuários não estão interessados em confiabilidade de sistemas distribuídos devido ao custo desses sistemas ✗

Correto

Notas relativas a este envio: 1/1.

2

Notas: 1

Como exemplo de um fracasso de mercado, os Ken Birman e demais autores citam:

Escolher uma resposta.

- a. DCE ✗
- b. CORBA ✗
- c. Xerox PARC ✗
- d. RPC ✓
- e. padrões para computação cliente-servidor ✗

Errado

Notas relativas a este envio: 0/1.

3

Notas: 1

Assinale verdadeiro ou falso de acordo com as opiniões dos autores:

para ter impacto, os pesquisadores devem demonstrar soluções no contexto de plataformas realísticas mas não devem se preocupar com custos pois são soluções inovadoras	Falso
todos os trabalhos acadêmicos devem ser julgados pelo seu valor comercial	Falso
trabalhos acadêmicos que queiram influenciar a indústria (de software) devem ir além da validação acadêmica tradicional, mesmo quando a avaliação é excepcionalmente rigorosa	Verdadeiro

Correto

Notas relativas a este envio: 1/1.

4

Notas: 1

Birman e demais autores afirmam que nas últimas 3 décadas a comunidade atuante em confiabilidade em sistemas distribuídos tem oferecido uma enorme quantidade de soluções para problemas de confiabilidade do mundo real. Os exemplos citados de soluções são (marque com X):

data mining	-
linguagens para programação paralela e distribuída	-
multicast confiável	X
redes de alta velocidade	-
teoria de computação distribuída	X
transações e mecanismos de atomicidade relacionados	X

Correto

Notas relativas a este envio: 1/1.

5

Notas: 1

Com a experiência em projetos acadêmicos e industriais, Birmam e demais autores afirmam ter alcançado a compreensão que a aceitação de mercado das tecnologias confiáveis apresenta alguma relação com a tecnologia em si, mas muito mais ainda com (assinale com X as opções que correspondem as afirmações dos autores):

compatibilidade com práticas padrões	X
desempenho demonstrado pela tecnologia na solução de problemas corriqueiros	-
impacto, no custo total, da construção, implantação e operação da solução como um todo	X
demanda real do usuário	-
credibilidade da visão e do processo a longo prazo	X

6

Notas: 1

Correto
Notas relativas a este envio: 1/1.

Os autores afirmam que os primeiros dias de sistemas de armazenamento de dados eram caracterizados pela exploração de um número considerável de novos paradigmas e metodologias de projeto. Neste cenário, duas inovações se destacaram, segundo os autores. São elas:

Escolha pelo menos uma resposta.

- ☐ a. sistemas de arquivos distribuídos ✗
- ☐ b. objetos distribuídos ✗
- ☐ c. sistemas de arquivo journaling ✗
- ☑ d. bancos de dados relacionais ✓
- ☑ e. modelo de computação transacional ✓

Correto
Notas relativas a este envio: 1/1.

7

Notas: 1

Seguindo a opinião de Birman e demais autores, assinale verdadeiro ou falso:

Por razões de desempenho, plataformas de base de dados necessitam altos níveis de concorrência.	Verdadeiro
Transações, e outras noções de atomicidade, oferecem ao programador uma forma de escrever programas como se cada aplicação executasse sozinha em um sistema.	Verdadeiro
Uma premissa para aplicar transações é que dados estejam armazenados em objetos persistentes identificáveis pelo sistema.	Verdadeiro
Programadores não vêm dificuldades em escrever código concorrente correto.	Falso

Correto
Notas relativas a este envio: 1/1.

8

Notas: 1

Segundo Birman e demais autores, uma solução simples e eficiente de tolerância a falhas relacionada a transações é:

Escolher uma resposta.

- ○ a. checkpointing ✗
- ○ b. códigos de correção como ECC ou chipkill ✗
- ○ c. redundância de dados ✗
- ◉ d. aborto ✓
- ○ e. atomicidade ✗

Correto
Notas relativas a este envio: 1/1.

9

Notas: 1

Assinale verdadeiro e falso seguindo a opinião de Birman e demais autores expressa no artigo.

A escalabilidade de servidores transacionais não representa nenhum problema.	Falso
Todos os programas admitem a separação entre código e dados necessária a sistemas de transações.	Falso
Tentativas de importar o modelo transacional para um âmbito mais vasto de computação distribuída demonstraram a versatilidade desta abordagem.	Falso
Transações têm se mostrado um sucesso de mercado fenomenal.	Verdadeiro
Transações são o modelo predominante de programação para aplicações onde é praticável a separação de dados e código.	Verdadeiro

Correto
Notas relativas a este envio: 1/1.

10
Notas: 1

O modelo de transações aninhadas geralmente termina com *commit* de duas ou 3 fases. A respeito deste mecanismo, Birman e demais autores afirmam que (assinale verdadeiro ou falso)

O mercado, com efeito, rejeitou o uso de transações que acessam múltiplos servidores.	Verdadeiro
Produtos comerciais que usam commit de duas ou 3 fases são eficientes, flexíveis e fáceis de serem aplicados.	Falso
Commit de duas ou 3 fases foi usado amplamente em vários produtos comerciais (pelo menos durante algum tempo).	Verdadeiro
Commit de duas ou 3 fases é caro.	Verdadeiro
Padrões para transações em WebServices não permitem o uso de transações complexas aninhadas.	Falso

Correto
Notas relativas a este envio: 1/1.

11
Notas: 1

Birman e demais autores citam 3 exemplos de aplicações que tradicionalmente não tem sido consideradas adequadas para sistemas de transações. Marque com X os 3 exemplos citados pelos autores:

Serviços críticos quanto ao tempo.	X
Programas distribuídos que cooperam indiretamente através de uma base de dados compartilhada.	-
Programas distribuídos que cooperam diretamente através da passagem de mensagens.	X
Aplicações leves com estruturas de dados na memória.	X
Aplicações concorrentes com acesso a dados comuns armazenados em uma base de dados.	-

Correto
Notas relativas a este envio: 1/1.

12
Notas: 1

Birman e demais autores, no item que trata de multicast confiável, pede aos leitores imaginarem um grupo de processos que colabora para replicar dados. A forma de um processo atualizar os dados é:

Escolher uma resposta.

- a. solicitar o serviço de escrita no dado a um servidor central por meio de uma mensagem e esperar a resposta do servidor; o servidor central se encarrega de replicar os dados na sua memória estável ✗
- b. fazer a atualização localmente e enviar o resultado da operação a um servidor central que se encarrega de comunicar o resultado aos demais membros do grupo ✗
- c. difundir uma mensagem de atualização a todos os membros operacionais do grupo (ou a um quorum deles), sendo que a difusão se encarrega de descobrir quais os membros atuais do grupo e ordenar todos os pedidos de atualização, ou seja, colocar os pedidos em uma ordem previamente acertada ✓
- d. garantir que mais ninguém quer fazer uma atualização sobre o mesmo dado para não haver conflitos, atualizar o dado e enviar o resultado para todos os demais membros do grupo ✗
- e. partir do princípio otimista que nenhum outro processo vai querer atualizar os mesmos dados e, de tempos e em tempos, difundir as atualizações já realizadas aos outros membros do grupo ✗

Correto
Notas relativas a este envio: 1/1.

13
Notas: 1

Membership do grupo significa:

Escolher uma resposta.

- a. todos os processos que enxergam um grupo em um dado momento, pertencendo ou não a ele ✗
- b. apenas os processo que abandonaram o grupo, seja voluntariamente ou por terem sido excluídos após detecção de uma falha ✗

- c. todos os processos que pertencem a um grupo em um dado momento ✓
- d. todos os processos que se uniram ao grupo desde a última atualização de dados replicados ✗
- e. todos os clientes de um serviço fornecido por um grupo de processos ✗

Errado
Notas relativas a este envio: 0/1.

14
Notas: 1

Joins e *leaves* são operações:

Escolher uma resposta.

- a. para troca de mensagens entre os membros em um grupo, que os processos usam quando querem atualizar dados replicados ✗
- b. sobre dados mantidos em um grupo, que os processos usam quando querem escrever ou ler os dados ✗
- c. de transação sobre uma base de dados ✗
- d. para troca de mensagens entre os membros em um grupo, que os processos usam quando querem se comunicar por multicast ✗
- e. sobre a visão de grupo, que os processos usam quando querem entrar ou sair de um grupo ✓

Correto
Notas relativas a este envio: 1/1.

15
Notas: 1

Os autores criaram um termo para representar a evolução do estado de um grupo vista por todos os seus membros como uma sequência de mudanças de membership, atualizações de dados replicados, defeitos, etc. O termo é:

Escolher uma resposta.

- a. group view ✗
- b. membership ✗
- c. transações atômicas ✗
- d. sincronia virtual ✓
- e. multicast confiável ✗
- f. sincronia serial ✗

Correto
Notas relativas a este envio: 1/1.

16
Notas: 1

Na sincronia virtual a execução parece síncrona assim como a execução transacional parece serial .

Correto
Notas relativas a este envio: 1/1.

17
Notas: 1

Ainda considerando as contribuições de Birman para a área de *multicast* confiável, podemos afirmar que:

Sincronia virtual oferece um modelo forte e torna relativamente fácil replicar dados em aplicações que não se adequam ao modelo transacional.	Verdadeiro
Sincronia virtual foi um enorme sucesso no mercado.	Falso
Sincronia virtual foi usada como base para vários produtos comerciais como por exemplo IBM Websphere e Microsoft Windows Clustering.	Verdadeiro
As garantias do modelo de sincronia virtual não podem ser reduzidos a expressões matemáticas e portanto não é possível a prova de teoremas mostrando que protocolos de comunicação implementam o modelo.	Falso

Correto
Notas relativas a este envio: 1/1.

18
Notas: 1

Birman afirma que a sincronia virtual foi acompanhada de outra tecnologia, cujo autor foi Leslie Lamport, que também garantia que os processos em um sistema viam eventos idênticos em uma ordem idêntica e que sofreu de problemas similares no mercado. Seu nome é:

Escolher uma resposta.

- a. generais bizantinos ✗

b. timestamps ✗

c. Paxos ✓

d. consenso probabilístico ✗

e. FLP ✗

Correto
Notas relativas a este envio: 1/1.

19
Notas: 1

Considerando os dois modelos considerados irmãos no texto, assinale verdadeiro e falso de acordo com a comparação feita pelos autores:

Paxos alcançou maior impacto comercial que a sincronia virtual	Falso
Paxos é mais rápido que as implementações disponíveis de sincronia virtual	Falso
Paxos garante propriedades de confiabilidade mais fortes que sincronia virtual	Verdadeiro
Paxos escala pior que as implementações disponíveis de sincronia virtual	Verdadeiro
o custo de Paxos é menor que o da sincronia virtual	Falso

Correto
Notas relativas a este envio: 1/1.

20
Notas: 1

No item dedicado à teoria de computação distribuída, Birman et al. discutem o sucesso de propostas teóricas para a solução de problemas da área. Considere a opinião dos autores apresentada no artigo e assinale verdadeiro ou falso para as sentenças a seguir.

A teoria é largamente ignorada pelos profissionais da área.	Verdadeiro
A área gerou milhares de artigos incluindo alguns resultados considerados clássicos.	Verdadeiro
Surpreendentemente, a teoria mostrou pouco impacto direto.	Verdadeiro
A teoria é vista como tendo sido desenvolvida fortemente conectada a problemas práticos.	Falso

Correto
Notas relativas a este envio: 1/1.

21
Notas: 1

Uma das duas possíveis causas para o fracasso da aceitação da teoria de computação distribuída entre os profissionais da área é explicada por Briman et al. no item dedicado ao modelo de computação assíncrona. Sobre esse modelo podemos afirmar que (assinale verdadeiro ou falso):

a simplicidade deveria oferecer um contexto em que soluções pudessem ser desenvolvidas e posteriormente portadas para o mundo real	Verdadeiro
o modelo foi introduzido para explicar da forma mais simples possível um sistema de comunicação	Verdadeiro
o objetivo era oferecer um modelo teórico para derivar propriedades idealizadas sem qualquer compromisso com o mundo real	Falso

Correto
Notas relativas a este envio: 1/1.

22
Notas: 1

Assinale com X as características do modelo assíncrono:

uma mensagem entre dois processos em operação será entregue eventualmente (ou seja em tempo finito) no destino mas não existe forma de dizer que a entrega ocorre prontamente	X
não há noção de tempo	X
processos interagem através de uma abstração de uma memória comum	-
processos que param de operar avisam aos demais processos no sistema	-
processos se comunicam através da passagem de mensagens	X
processos com defeito param silenciosamente	X
falhas são detectadas por time-out	-

Correto
Notas relativas a este envio: 1/1.

23

Notas: 1

Em relação ao modelo assíncrono, assinale verdadeiro ou falso:

Não se encontram no mundo real sistemas que utilizam estas técnicas, portanto, o modelo não possui qualquer valor prático.	Falso
O modelo assíncrono rendeu técnicas práticas para resolução de problemas tais como rastreamento de causalidade potencial, detecção de deadlocks, coordenação da criação de checkpoints para evitar a efeito dominó.	Verdadeiro
Todos os sistemas do mundo real são assíncronos, o que mostra o valor do modelo e de suas contribuições práticas	Falso

Correto

Notas relativas a este envio: 1/1.

24

Notas: 1

Os autores afirmam que o maior sucesso do modelo assíncrono também é associado a um grande fracasso de mercado. Eles se referem ao consenso assíncrono. Como os autores enunciam o consenso assíncrono?

Escolher uma resposta.

- a. estratégia de detecção de falhas temporais em sistemas assíncronos ✗
- b. acordo sobre uma propriedade em um sistema assíncrono ✓
- c. algoritmo especial para eleição de líder em sistemas assíncronos ✗
- d. transformação de um sistema assíncrono em um sistema síncrono ✗
- e. forma de distinguir colapsos (crashes) reais de desconexões transientes de rede ✗

Correto

Notas relativas a este envio: 1/1.

25

Notas: 1

Em um trabalho teórico de extrema importância de Fischer, Lynch e Paterson, foi enunciado o problema da impossibilidade de consenso (posteriormente passou a ser conhecido como impossibilidade FLP). O que Birman menciona sobre a impossibilidade FLP no artigo?

Escolher uma resposta.

- a. Fisher et al. demonstraram que o consenso assíncrono é impossível na presença de falhas, a menos que um sistema possa distinguir com precisão falhas de colapso (crashes) reais de desconexões transitórias de rede ✓
- b. Fisher e al. demonstraram que o consenso é impossível na presença de qualquer tipo de falha em qualquer tipo de sistema tanto assíncrono como síncrono. ✗
- c. Birman demonstrou que o consenso é impossível na presença de falhas bizantinas a menos que um sistema pode distinguir com precisão entre falhas reais e falhas provocadas por intrusos na rede. ✗
- d. Fisher e al. demonstraram que o consenso é impossível na presença de falhas bizantinas a menos que o sistema seja assíncrono. ✗

Correto

Notas relativas a este envio: 1/1.

26

Notas: 1

Considerando um determinado grupo de processos tentando chegar a um consenso através de troca de mensagens em um sistema assíncrono, o que significa "impossível" na impossibilidade FLP?

Escolher uma resposta.

- a. o problema de alcançar consenso nunca pode ser resolvido ✗
- b. não é possível saber de antemão se o problema de alcançar consenso pode ser resolvido ou não ✗
- c. o problema de alcançar consenso algum dia será resolvido para o grupo de processos envolvido, mas não é possível esperar eternamente ✗
- d. o problema de consenso nem sempre pode ser resolvido ✓

Correto

Notas relativas a este envio: 1/1.

27

Notas: 1

De acordo com Birman e al. FLP é de enorme importância para a comunidade teórica. Na verdade, o resultado é a descoberta mais importante até hoje nesta área. No entanto, a essa "impossibilidade" tem sido uma fonte de confusão entre os profissionais. Assinale verdadeiro ou falso de acordo com o texto:

As pessoas têm discutido se faz sentido provar a impossibilidade de alguma coisa	Verdadeiro

em um modelo (o modelo assíncrono) que de qualquer forma é super simplificado.		Parcialmente correta Notas relativas a este envio: 0.75/1.
É impossível construir um sistema que seja robusto contra defeitos.	Falso	
É fácil resolver consenso como um algoritmo expresso sobre multicast confiável.	Verdadeiro	
FLP simplesmente nos diz que é muito fácil evitar certos padrões de retardo infrequentes no mundo real.	Verdadeiro	

Fechar esta janela

Fechar esta janela

1

Notas: 1

Retorne ao artigo "How the Hidden Hand Shapes the Market for Software Reliability" de Ken Birman, Coimbatore Chandersekaran, Danny Dolev e Robbert van Renesse, 2006. A segunda das duas possíveis causas para o fracasso da aceitação da teoria de computação distribuída entre os profissionais da área é explicada por Briman et al. no item dedicado ao modelo de computação síncrona. Sobre o modelo de computação síncrona podemos afirmar que (assinale verdadeiro ou falso):

o modelo síncrono força o sistema de comunicação de uma forma irreal	Verdadeiro
alguns processos enviam mensagens em rodadas enquanto outros ignoram as rodadas e se comportam de forma assíncrona	Falso
relógios são perfeitamente sincronizados	Verdadeiro
colapsos (crashes) são facilmente detectáveis	Verdadeiro
o modelo admite falhas bizantinas	Verdadeiro

Correto

Notas relativas a este envio: 1/1.

2

Notas: 1

Assinale com X as características do modelo síncrono:

não há noção de tempo	-
colapsos são fáceis de serem detectados	X
processos interagem através de uma abstração de uma memória comum	X
processos se comunicam através da troca de mensagens	-
o modelo inclui falhas maliciosas	X
falhas podem ser detectadas por time-out	X
existe um relógio global	X

Parcialmente correta

Notas relativas a este envio: 0.57/1.

3

Notas: 1

No contexto de sistemas distribuídos, o que são defeitos bizantinos?

Escolher uma resposta.

- a. Defeitos bizantinos correspondem a casos em que um determinado número de processos não apresenta defeitos de colapso, mas defeitos caracterizados pelo funcionamento arbitrário, malicioso e de maneira coordenada que presume o conhecimento perfeito do estado geral do sistema. ✓
- b. Defeitos bizantinos correspondem a casos em que todos processos do sistema apresentam defeitos coordenados de colapso, forçando o funcionamento arbitrário de um servidor central para restaurar o sistema e a excluir os processos maliciosos. ✗
- c. Defeitos bizantinos correspondem a casos em que um processo malicioso apresenta defeitos sucessivos de colapso, forçando um funcionamento arbitrário e de maneira coordenada dos demais processos para excluí-lo do sistema. ✗
- d. Um sistema sofre um defeito bizantino quando está sob ataque de intrusos maliciosos que agem de forma coordenada para derrubar (colapsar) um sistema. ✗

Correto

Notas relativas a este envio: 1/1.

4

Notas: 1

Em relação ao modelo síncrono, Birman et al. afirmam que um corpo substancial de teoria existe para este tipo de sistema, incluindo algumas impossibilidades e alguns algoritmos. Alguns desses algoritmos servem para a solução dos problemas dos generais bizantinos enunciado por Lamport em 1982. Sobre esse corpo de teoria, os autores afirmam que:

por muitos anos e apesar dos fascinantes resultados práticos, o Acordo Bizantino não foi aplicado em sistemas reais pois o tratamento matemático dado ao problema tornava difícil sua compreensão	Falso
mesmo para os problemas em que o modelo bizantino faz sentido, os aspectos irrealistas do modelo síncrono impedem que os usuários apliquem esses resultados diretamente	Verdadeiro
por muitos anos, o Acordo Bizantino foi de interesse puramente teórico: apenas um resultado matemático fascinante	Verdadeiro

Correto

Notas relativas a este envio: 1/1.

5

Notas: 1

Recentemente (2002) o Acordo Bizantino, que foi enunciado em 1982, experimentou um renascimento ilustrativo da tese central do artigo de Birman e demais autores que estamos analisando. Sobre este renascimento, Birman afirma que:

Escolher uma resposta.

- a. O modelo síncrono não é adequado para tratar da prevalência de vírus e spywares em sistemas de rede atuais. ✗
- b. Os trabalhos de Castro e outros autores serviram para demonstrar que o acordo bizantino não tem aplicação prática. ✗
- c. o modelo bizantino foi reformulado em configurações realistas de rede e com isso se resolveu o problema de construir servidores ultra-defensáveis contra vírus e intrusos ✓
- d. Castro e Liskov chegaram a conclusão que servidores de rede nao podem tolerar ataques de intrusos e vírus ✗

Correto

Notas relativas a este envio: 1/1.

6

Notas: 1

Birman, no artigo, afirmar que "dada a prevalência de vírus e spyware, a nova abordagem introduzida por Castro e Liskov ao Acordo Bizantino está encontrando algum interesse comercial, embora em um pequeno mercado. Ao revisitar o problema em um contexto mais realista, a pesquisa de Castro e Liskov demonstrou o seu valor prático."

Resposta:

- Verdadeiro ✓
- Falso ✗

Correto

Notas relativas a este envio: 1/1.

7

Notas: 1

No item IV, Birman e demais autores discutem as razões do fracasso de mercado das tecnologias para sistemas distribuídos confiáveis. A experiência dos autores com plataformas multicast sugere algumas razões para esse fracasso. Qual a **primeira** razão listada:

Escolher uma resposta.

- a. a apresentação do multicast foi muito primitiva (ou seja, de baixo nível) ✓
- b. produtos multicast substituíam sistemas operacionais populares, com os quais os programadores já estavam familiarizados ✗
- c. desenvolvedores modernos preferem trabalhar com recursos de baixo nível de rede, multicast era apresentado com um nível de abstração muito alto ✗
- d. multicast deveria ter sido apresentado para os pesquisadores como uma tecnologia de rede ✗

Correto

Notas relativas a este envio: 1/1.

8

Notas: 1

Entre as razões listadas para o insucesso das soluções de multicast no mercado, a **segunda** razão se relaciona à dificuldade de usar os produtos em ambientes reais. O exemplo citado de dificuldade de uso se refere a:

Escolher uma resposta.

- a. limitações de escalabilidade ✓
- b. um único processo não tinha consciência do grupo ao qual pertencia ✗
- c. surpresa dos usuários ✗
- d. excesso de membros nos grupos ✗
- e. colapso frequente de desempenho ✗

Correto

Notas relativas a este envio: 1/1.

9

Notas: 1

Entre as razões listadas para o insucesso das soluções de multicast no mercado, a **terceira** razão mencionada é:

Escolher uma resposta.

- a. soluções multicast apresentam muitas limitações que são inaceitáveis aos usuários ✗
- b. soluções de multicast avançaram mais rápido que os padrões aceitos no mercado ✗

- c. soluções multicast não acompanharam os avanços na área como padrões de orientação a objetos e arquiteturas orientadas a serviço ✓
- d. soluções multicast ficaram vinculadas unicamente a CORBA que não foi bem sucedida no mercado ✗
- e. soluções multicast ficaram vinculadas unicamente a réplicas de dados em servidores idênticos e determinísticos ✗

Correto
Notas relativas a este envio: 1/1.

10
Notas: 1

Entre as razões listadas para o insucesso das soluções de multicast no mercado, a **quarta** razão mencionada se relaciona ao tamanho do mercado. De acordo com os autores esse mercado é:

Escolher uma resposta.

- a. de pequeno tamanho pois os data centers geralmente não contam com um grande número de servidores ✓
- b. grande, pois mesmo sendo pequeno o número de servidores, o número de clientes é enorme e todos os clientes deveriam pagar licenças de uso de soluções multicast ✗
- c. de tamanho considerável pois todas as instalações de computadores contam com pelo menos um servidor ✗
- d. enorme, graças ao surgimento de clouds ✗
- e. de tamanho global pois todo mundo é cliente de soluções de rede ✗

Correto
Notas relativas a este envio: 1/1.

11
Notas: 1

Em relação a **quinta** razão do insucesso de mercado de soluções de multicast, os autores afirmam que:

Esses produtos não têm mostrado, com ao menos um caso significativo, que os desenvolvedores que os usam ganham benefícios econômicos diretos em relação aos desenvolvedores que não os usam.	Verdadeiro
Um caso significativo pode ser construído, mas esta não foi uma prioridade para a comunidade de pesquisadores.	Verdadeiro
Um caso significativo que mostra os benefícios econômicos das soluçoes de multicast não pode ser construído.	Falso

Correto
Notas relativas a este envio: 1/1.

12
Notas: 1

A **sexta** razão do insucesso de mercado de soluções de multicast, os autores atribuem ao alto custo dos produtos. Em relação ao custo podemos afirmar que:

os compradores compreendem as razões do alto custo e consideram que o custo é real e vale a pena para alcançar a confiabilidade necessária	Falso
não há razão para soluções de multicast não custarem uma pequena fração do que custa um sistema operacional ou um banco de dados	Falso
se os vendedores de soluções multicast exigiram um preço alto de licença, que possa dar lucro aos seus negócios, os compradores vão se recusar a comprar os produtos	Verdadeiro
multicast não é um tecnologia simples, os produtos são caros para desenvolver e manter	Verdadeiro
do ponto de vista do comprador, um framework para replicação de dados é algo muito útil, mas não é nem remotamente tão útil como um sistema operacional ou um sistema de banco de dados	Verdadeiro

Correto
Notas relativas a este envio: 1/1.

13
Notas: 1

Entre as razões listadas para o insucesso das soluções de multicast no mercado, a **sétima** razão mencionada se relaciona ao fato das soluções de multicast demandarem serviços adicionais de desenvolvimento. De acordo com os autores esse fato é:

Escolher uma resposta.

- a. não necessariamente mau, muitas companhias fazem negócio com esses serviços ✓
- b. ótimo, pois vai propiciar a criação de inúmeros empregos ✗
- c. inevitável, quanto mais investimento, menos completo o produto ✗

d. péssimo, soluções deveriam vir completas e embaladas em celofane ✗

Correto
Notas relativas a este envio: 1/1.

14
Notas: 1

Resumindo sua discussão sobre o insucesso de soluções de multicast no mercado, Birman e os demais autores do artigo afirmam que:

a associação com os principais fornecedores de plataformas já ocorreu e permite maior permanência às soluções de multicast	Falso
construir aplicações escaláveis envolve replicar dados de tal forma que consultas possam ser balanceadas em múltiplos servidores	Verdadeiro
a crescente popularidade de clusters e data centers vai favorecer produtos dos fornecedores de plataformas	Verdadeiro
criar demanda por soluções de replicação abre as portas para tecnologias que também promovem tolerância a falhas, segurança (security) e QoS	Verdadeiro

Correto
Notas relativas a este envio: 1/1.

15
Notas: 1

Ainda finalizando sua discussão sobre o insucesso de soluções de multicast no mercado, Birman e os demais autores do artigo afirmam que notaram desincentivo a considerar confiabilidade uma mercadoria (commodity) por parte dos vendedores de plataforma. Em relação a essa questão, e seguindo a opinião dos autores, assinale Verdadeiro ou Falso, para sentenças a seguir:

as plataformas são suficientemente robustas e não precisam de soluções de tolerância a falhas em software	Falso
os fornecedores de plataformas anunciam que se os usuários executarem suas aplicações nos seus produtos eles terão a robustez necessária e que de outra forma não seria possível	Verdadeiro
se os usuários podem construir aplicações robustas sem necessidade de plataformas caras, os vendedores dessas plataformas podem ter prejuízos financeiros	Verdadeiro

Correto
Notas relativas a este envio: 1/1.

16
Notas: 1

E finalmente, o que os autores têm a dizer, no item IV que trata da discussão, sobre trabalhos acadêmicos?

o impacto no setor comercial não precisa ser usado para avaliar trabalhos teóricos que lançam luz sobre questões de fundamental importância e que influenciam futuros trabalhos teóricos	Verdadeiro
o truque para ter impacto no mundo real é atacar um problema difícil através de uma abordagem teórica que permita toda uma nova linha de produtos, como foi o caso da solução prática para o problema bizantino	Verdadeiro
problemas reais são essencialmente determinísticos e a comunidade teórica pode ter sucesso comercial desenvolvendo soluções para esses sistemas	Falso

Correto
Notas relativas a este envio: 1/1.

17
Notas: 1

No item V, após uma série de observações, os autores fazem algumas recomendações para o pessoal envolvido em desenvolver soluções para aumentar a confiabilidade em sistemas distribuídos. Em relação a **primeira** recomendação, os autores sugerem que:

desenvolvedores precisam saber avaliar suas soluções em cenários realísticos	Verdadeiro
desenvolvedores precisam criar demonstrações das suas soluções para sistemas distribuídos usando plataformas reais, se possível	Verdadeiro
trabalhos de pesquisa não precisam ser demonstrados com rigor teórico, desde que as demonstrações práticas em plataformas reais sejam bem sucedidas	Falso

Correto
Notas relativas a este envio: 1/1.

18

No item V, após uma série de observações, os autores fazem algumas recomendações para o pessoal envolvido em

Notas: 1

desenvolver soluções para aumentar a confiabilidade em sistemas distribuídos. Em relação a **segunda** recomendação, os autores sugerem que:

trabalhos teóricos não precisam se preocupar em provar seu valor em termos econômicos	Falso
a perspectiva econômica prejudica o julgamento do valor de um trabalho acadêmico	Verdadeiro
o valor dos resultados alcançados deve ser mostrado em termos tais que os potenciais compradores possam entender a partir de sua perceptiva econômica	Verdadeiro

Parcialmente correta
Notas relativas a este envio: 0.67/1.

19

Notas: 1

No item V, os autores fazem algumas recomendações para o pessoal envolvido em desenvolver soluções para aumentar a confiabilidade em sistemas distribuídos. Em relação a **terceira** recomendação, que foca a continuidade do produto a longo prazo, os autores sugerem que:

se o time de desenvolvimento não pode garantir suporte de longo prazo, o time deveria tentar disseminar sua solução através de fornecedores, ao menos durante um período de transição	Verdadeiro
se o time de desenvolvimento não puder garantir suporte de longo prazo a uma dada solução, melhor nem começar o seu desenvolvimento	Falso
qualquer time de desenvolvimento deve garantir suporte de longo prazo ao produto criando uma empresa fornecedora de soluções	Falso

Correto
Notas relativas a este envio: 1/1.

20

Notas: 1

A última recomendação diz respeito a propriedade intelectual. Problemas em relação a propriedade intelectual podem prejudicar o relacionamento entre a indústria e a academia com prejuízo para ambas as partes envolvidas. Birman e demais autores sugerem, neste aspecto, que:

Escolher uma resposta.

- a. pesquisadores só deveriam produzir software com o modelo de licença adotado pelo Unix BSD ✗
- b. pesquisadores só deveriam produzir software com o modelo de licença de software livre ✗
- c. pesquisadores entrem em contato com o escritório de licenciamento de suas universidades para esclarecer e padronizar o tratamento de patentes ✓
- d. pesquisadores evitem contato com empresas que operam com patentes ✗

Correto
Notas relativas a este envio: 1/1.

Fechar esta janela Fechar esta janela