INF01209 - Fundamentos de Tolerância a Falhas

Você acessou como João Luiz Grave Gross (Sair)

Moodle do INF ► FTF 2012/2 ► Questionários ► Tolerância a falhas na aplicação ► Revisão da tentativa 1

Tolerância a falhas na aplicação

Revisão da tentativa 1

Terminar revisão

Iniciado em	sexta, 14 setembro 2012, 03:47
Completado em	segunda, 24 setembro 2012, 13:37
Tempo empregado	10 dias 9 horas
Notas	27.58/28
Nota	98.49 de um máximo de 100(98 %)

Notas:

1

No item 2.1 do artigo: *V. D. Florio e C. Blondia,* "A survey of linguistic structures for application-level fault tolerance", ACM Comput. Surv., vol. 40, no. 2, p. 6:1–6:37, maio 2008, quando argumentam sobre a necessidade de tolerância a falhas de software, os autores comentam sobre tolerância a falhas de hardware. Assinale **verdadeiro** ou **falso** segundo as opiniões expressas no texto.

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.

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.

Componentes de hardware são apenas uma das fontes de não confiabilidade em sistemas computacionais, decrescendo em importância conforme a confiabilidade desses componentes aumenta.

A prevalência de falhas de hardware está aumentando continuamente.



Verdadeiro

Correto

Notas relativas a este envio: 1/1.

Notas:

2

Ainda no item 2.1 do artigo, os autores comentam sobre a complexidade dos sistemas computacionais. Assinale **verdadeiro** ou **falso** segundo as opiniões expressas no texto.

Quanto maior o nível de abstração, maior é a complexidade dos algoritmos que entram em jogo e a consequente propensão a erro das máquinas envolvidas (sejam máquinas reais ou abstratas).



A maior parte da complexidade dos modernos sistemas computacionais reside no hardware.



Dividindo a complexidade do sistema em camadas, falhas ficam confinadas a uma única camada, tornando mais fácil gerenciar a complexidade crescente dos sistemas.



Uma forma conceitual eficente e flexível de gerenciar complexidade é considerar um sistema como uma hierarquia de máquinas abstratas sofisticadas.



Mesmo que a divisão em camadas torne a complexidade transparente, a complexidade ainda é parte do sistema global sendo desenvolvido e falhas se propagando por várias camadas podem ocorrer.

Verdadeiro **▼**

Parcialmente correta

Notas relativas a este envio: 0.8/1.

Notas:

3

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

Escolher uma resposta.

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

Correto

Notas relativas a este envio: 1/1.

Notas:

4

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 aumenta os custos de desenvolvimento mas contribui para aumento de confiabilidade porque os componentes já foram exaustivamente testados
- b. o desenvolvedor da aplicação não conhece o desenvolvedor do componente reusável e por essa razão não pode confiar na qualidade do componente
- c. 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
- 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.

Notas:

5

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.

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



soluções 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 🔻

o argumento implica que todas as tarefas de tolerância a falhas podem ser realizadas no nível da aplicação

falso 🔻

como exemplo é citado que um canal de comunicação extraordinariamente confiável livra a aplicação da obrigação de prover confiabilidade



Correto

Notas relativas a este envio: 1/1.

Notas: 1

6

No item 2.3, os autores listam problemas de projeto relacionados a tolerância a falhas no nível da aplicação. Quais foram os problemas mencionados?

Escolher uma resposta.

- a. como incorporar tolerância a falhas no nível da aplicação, como treinar a equipe de desenvolvedores para aplicar tolerância a falhas no nível da aplicação, e como gerenciar o código tolerante a falhas 🦹
- b. como treinar a equipe de desenvolvedores para aplicar tolerância a falhas no nível da aplicação, quais mecanismos de tolerância a falhas devem ser suportados, e como gerenciar o código tolerante a falhas 🦹
- 🔘 c. como incorporar tolerância 🛮 como treinar a equipe de a falhas no nível da aplicação, desenvolvedores para quais recursos de tolerância aplicar tolerância a falhas no a falhas devem ser suportados, e como gerenciar mecanismos de tolerância a o código tolerante a falhas

nível da aplicação, quais falhas devem ser suportados, e como gerenciar o código tolerante a falhas

 d. como reduzir os custos de projeto redundante, como treinar a equipe de desenvolvedores para aplicar tolerância a falhas no nível da aplicação, e como gerenciar o código tolerante a falhas 🦹

Correto

Notas relativas a este envio: 1/1.

7 Notas: 1

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

Escolher uma

- 🔘 a. como reduzir os custos de projeto 🦹
- D. como treinar a equipe de desenvolvedores para aplicar

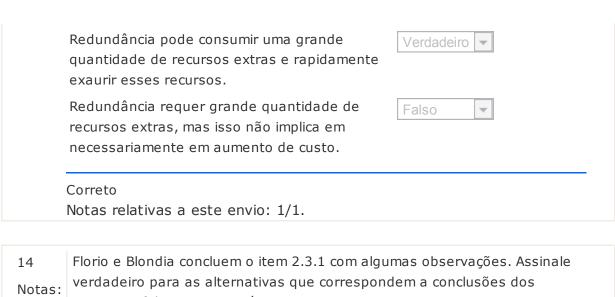
	resposta. tolerânc	ia a falhas no nível da a¡	plicação 🦹
	C. como	gerenciar o código tolera	ante a falhas 🦹
	🔵 d. quais suporta	mecanismos de tolerând dos 🚜	cia a falhas devem ser
	⊚ e. como	incorporar tolerância a f	alhas no nível da aplicação
	Correto Notas relativas a este	envio: 1/1.	
8 Notas: 1	apropriadas para a inco Soluções inadequadas a	essário o uso de técnicas rporação de tolerância a a esse problema, segund r alto grau de intrusividad	falhas no nível da aplicação. lo o artigo de Florio e
	Correto Notas relativas a este	envio: 1/1.	
9 Notas: 1	resulta na mistura do c verdadeiro para as con	ródigo funcional com o có nsequências ao desenvo ura, e falso quando a alí dos códigos:	• •
	aumentam		verdadello 🕶
	o tamanho do código re	esultante é menor	falso
	o código resultante é m	nais fácil de ser mantido	falso
	a complexidade do siste	ema diminui	falso
	as tarefas do desenvol significativamente mais		verdadeiro 🔻
	Correto Notas relativas a este	envio: 1/1.	
10 Notas:	Com relação ao proble aplicação, Florio e Blon	·	olerância a falhas no nível da
1		orporação neste nível é in vel conciliar interesses 🥻	·
			ilhas devem prevalecer em no código da aplicação 🦹
		es diferentes de desenvo gadas da parte funcional	olvedores devem ser I do código e da parte de

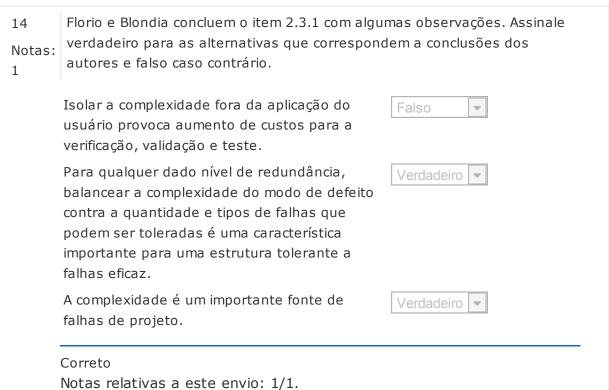
suporte a tolerância a falhas 🦹 🔘 d. uma estrutura ideal de sistema deve garantir uma adequada separação entre os interesses funcionais e os relativos a tolerância a falhas no código da aplicação 🧹 Correto Notas relativas a este envio: 1/1. 11 Segundo Florio e Blondia, uma resposta inadequada para o problema de qual recurso de tolerância a falhas deve ser suportado pelo nível da Notas: aplicação pode apresentar duas características inconvenientes: 1 🔘 a. requerer algoritmos muito sofisticados e estratégias de Escolher verificação de alto custo 🦹 uma resposta. b. requerer desenvolvedores altamente especializados e ferramentas de teste sofisticadas 🦹 🔘 c. exigir equipamento especial e linguagens de programação específicas 🦹 d. requerer alto grau de redundância e rapidamente. consumir a quantidade de redundância disponível 🗸 Correto Notas relativas a este envio: 1/1. Florio e Blondia definem 3 atributos estruturais para tolerância a falhas no 12 nível da aplicação (ALTF) chamados de SC, SA e A. Associe: Notas: 1 separação de interesses funcionais e não funcionais suporte direto a um grande conjunto de SA mecanismos de tolerância a falhas facilidade do código tolerante a falhas se adaptar ao ambiente onde se encontra no momento Correto Notas relativas a este envio: 1/1. Considerando o conceito de redundância como apresentado no item 2.3.1 13 do artigo de Florio e Blondia, é possível afirmar que: Notas:

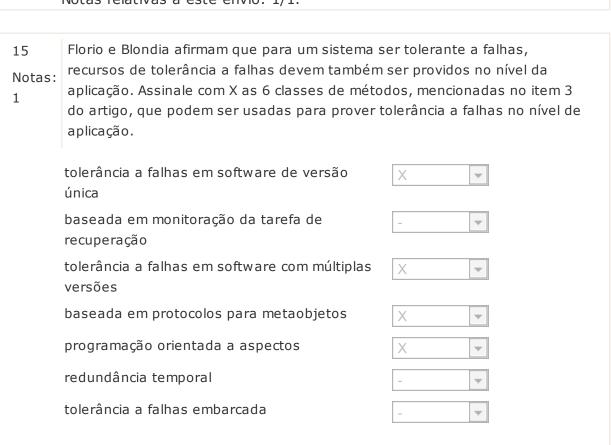
Considerando o conceito de redundância como apresentado no item 2.3.1 do artigo de Florio e Blondia, é possível afirmar que:

Tolerância a falhas é, em geral, o resultado de alguma estratégia que efetivamente explora alguma forma de redundância.

É impossível acrescentar mais confiabilidade a um canal não confiável apenas aumentando o grau de redundância de informação.







	novas lingi customizad		listribuidas X		I
	redundânc	cia espacial	X		
	Parcialmen Notas relat	te correta tivas a este envio: 0.78/	′1.		
16 Notas:	_	quisito chave para o dese acordo com Florio e Blono		sistemas to	lerantes a
1	Escolher	a. Replicação de dad	os 🦹		
	uma	b. Recursos replicado	os em hardwar	e ou em soft	ware 🎸
	resposta.	c. Recursos replicado	s em hardware	e X	
		🔘 d. Replicação tempor	al 🦹		
		e. Recursos replicado	os em software	X	
	Correto Notas relat	tivas a este envio: 1/1.			
Notas: 1	falhas con domínios		Segundo Florio	e Blondia, q	uais os seus
Notas: 1	falhas con domínios ? Escolher	n múltiplas computações. ? a. Espaço, paralelisn	Segundo Florio no e codificação	e Blondia, q	uais os seus
Notas: 1	falhas con domínios	n múltiplas computações. ? a. Espaço, paralelisn b. Tempo, concorrên	Segundo Florio no e codificação cia e paralelism	e Blondia, q	uais os seus
Notas: 1	falhas com domínios ? Escolher uma	n múltiplas computações. ? a. Espaço, paralelisn b. Tempo, concorrên c. Tempo, espaço e i	Segundo Florio no e codificação cia e paralelism nformação	e Blondia, q de dados a	uais os seus
	falhas com domínios ? Escolher uma	n múltiplas computações. ? a. Espaço, paralelisn b. Tempo, concorrên	Segundo Florio no e codificação cia e paralelism nformação	e Blondia, q de dados a	uais os seus
Notas: 1	falhas con domínios ? Escolher uma resposta.	n múltiplas computações. ? a. Espaço, paralelisn b. Tempo, concorrên c. Tempo, espaço e i	Segundo Florio no e codificação cia e paralelism nformação	e Blondia, q de dados a	uais os seus
Notas: 1 18 Notas:	falhas con domínios ? Escolher uma resposta. Correto Notas relat	n múltiplas computações. ? a. Espaço, paralelisn b. Tempo, concorrên c. Tempo, espaço e i d. Codificação de da	Segundo Florio no e codificação cia e paralelism nformação dos, tempo e co	e Blondia, q de dados no X oncorrência	uais os seus
Notas: 1	falhas con domínios ? Escolher uma resposta. Correto Notas relatores considera segundo F	n múltiplas computações. a. Espaço, paralelism b. Tempo, concorrên c. Tempo, espaço e i d. Codificação de dad tivas a este envio: 1/1.	Segundo Florio no e codificação cia e paralelism nformação dos, tempo e co	e Blondia, q de dados no X oncorrência	uais os seus
Notas: 1 18 Notas:	falhas con domínios ? Escolher uma resposta. Correto Notas relata segundo F	n múltiplas computações. a. Espaço, paralelism b. Tempo, concorrên c. Tempo, espaço e i d. Codificação de dad tivas a este envio: 1/1.	Segundo Florio no e codificação cia e paralelism nformação dos, tempo e co	e Blondia, q de dados no X oncorrência	uais os seus

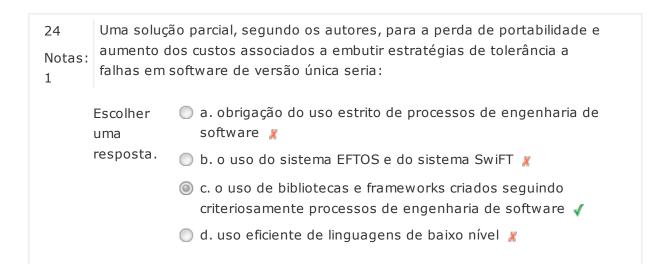
Notas:	3.1 do art		nificado aos termos na notação
	número de	e programas	p
	número de	e canais de hardware	m 🔻
	número de	e execuções	n 🔻
	Correto Notas relat	tivas a este envio: 1/1.	
20	Um sistem	na 1T/1H/1S, de acordo com Flo	orio e Blondia, é chamado de:
Notas:	Escolher	🔘 a. tolerante a falhas 🦹	
	uma resposta.	🔵 b. intolerante a falhas 🦹	
	resposta.	🔵 c. duplex 🦹	
		d. simplex √	
		🔵 e. replicado 🦹	
	Correto Notas relat	tivas a este envio: 1/1.	
21	Tolerância	a falhas em versão única de s	software implica em:
Notas:			
1	Escolher uma	a. Embutir em um sistema técnicas de detecção e re	a simplex (ou seja, sem réplicas) ccuperação de erros. 🎸
	resposta.	b. Construir programas q replicados.	ue não possam ser copiados nem
		 c. Executar software em a processos e múltiplos usu 	ambientes que inibam múltiplos Jários simultâneos. 🦹
		O d. Construir programas a	plicativos sem erros. 🦹
		técnicas de ações atômic	s programas aplicativos as as, pontos de recuperação e e tratamento de exceções. 🗶
	Correto Notas relat	tivas a este envio: 1/1.	
22	_		olerância a falhas em versão única
Notas:	ue soitwa	re apresenta como desvantag	CIIS.
_	Escolher uma	a. necessidade de optar sistema SwiFT	entre o sistema EFTOS e o
	resposta.	 b. necessidade de desen para suprir os recursos d 	volver bibliotecas e frameworks e tolerância a falhas 🧣

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

Correto

Notas relativas a este envio: 1/1.

Considerando o conceito de SV como apresentado por Florio e Blondia, 23 responda verdadeiro ou falso: Notas: 1 A adoção do SV na camada de aplicação requer | Verdadeiro concentrar no código fonte da aplicação tanto a especificação do que fazer como a estratégia de tolerância a falhas. SV baseia-se na hipótese que todos os recursos de tolerância a falhas são providos transparentemente pelo hardware, o que gera um código leve, facilmente portável, de baixo custo e baixa necessidade de manutenção. A principal vantagem de incluir em um único Falso código tanto a funcionalidade do sistema como as estratégias de tolerância a falhas é que essa abordagem aumenta a portabilidade do código pois o mesmo carrega consigo todos os recursos necessários. Como exemplos de técnicas de detecção e Verdadeiro recuperação, que podem ser embutidas em software de versão única, os autores citam ações atômicas, pontos de recuperação e rollback e tratamento de exceções. Correto Notas relativas a este envio: 1/1.



	Correto Notas relativas a este envio: 1/1.	
25	Considerando o sistema SwiFT, assinale as se	entenças verdeiras e falsas.
Notas: 1	SwiFT inclui uma série de componentes reutilizáveis de software adequados para incluir técnicas de tolerância a falhas em software de versão única	Verdadeiro ▼
	SwiFT é o nome de um computador tolerante a falhas que foi popular na década de 90	Falso 🔻
	SwiFT introduziu apenas um pequeno overhea 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.	
26 Notas: 1), SA (adequação sintática) e <i>A</i> ersão única, SV, como os
Notas:	Notas relativas a este envio: 1/1. Florio e Blondia definem no item 2.3 os concei interesses funcionais e de tolerância a falhas (adaptabilidade). Em relação a software de ve), SA (adequação sintática) e A ersão única, SV, como os
Notas:	Notas relativas a este envio: 1/1. Florio e Blondia definem no item 2.3 os concei interesses funcionais e de tolerância a falhas (adaptabilidade). Em relação a software de ve autores avaliam esses atributos? Indique suf), SA (adequação sintática) e A ersão única, SV, como os ciente ou insuficiente.
Notas:	Notas relativas a este envio: 1/1. Florio e Blondia definem no item 2.3 os concei interesses funcionais e de tolerância a falhas (adaptabilidade). Em relação a software de ve autores avaliam esses atributos? Indique suf separação de interesses), SA (adequação sintática) e A ersão única, SV, como os iciente ou insuficiente.
Notas:	Notas relativas a este envio: 1/1. Florio e Blondia definem no item 2.3 os concei interesses funcionais e de tolerância a falhas (adaptabilidade). Em relação a software de ve autores avaliam esses atributos? Indique suf separação de interesses adequação sintática), SA (adequação sintática) e A ersão única, SV, como os ciente ou insuficiente. insuficiente insuficiente
Notas:	Notas relativas a este envio: 1/1. Florio e Blondia definem no item 2.3 os concei interesses funcionais e de tolerância a falhas (adaptabilidade). Em relação a software de ve autores avaliam esses atributos? Indique suf separação de interesses adequação sintática adaptabilidade Correto Notas relativas a este envio: 1/1. Na conclusão do item 3.1.1, Florio e Blondia, a desvantagens da abordagem de versão única	npós reforçarem uma série de a, alertam para uma
Notas: 1 27 Notas:	Notas relativas a este envio: 1/1. Florio e Blondia definem no item 2.3 os concei interesses funcionais e de tolerância a falhas (adaptabilidade). Em relação a software de ve autores avaliam esses atributos? Indique suf separação de interesses adequação sintática adaptabilidade Correto Notas relativas a este envio: 1/1. Na conclusão do item 3.1.1, Florio e Blondia, a desvantagens da abordagem de versão única	npós reforçarem uma série de la alertam para uma Qual é ela?

- ◎ c. não apresenta qualquer restrição quanto a classe de aplicação que pode se beneficiar da abordagem
- d. não apresenta impacto nos custos de desenvolvimento e manutenção

Correto

Notas relativas a este envio: 1/1.

Notas:

28

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

- a. só podem ser usadas em programas escritos em C ou
 C++
- resposta.
- b. introduzem grande overhead no desempenho para apenas um pequeno aumento nos atributos de dependabilidade
- c. não apresentam qualquer impacto no custo ou no desempenho da aplicação
- 🔘 d. prejudicam a dependabilidade do sistema 🧣
- e. permitem ao projetista reusar peças de software sofisticadas e amplamente testadas

Correto

Notas relativas a este envio: 1/1.

Terminar revisão

Você acessou como João Luiz Grave Gross (Sair)

FTF 2012/2