Confiabilidade de Sistemas Operacionais de Propósito Geral: Um Estudo de Caso

Marcela Prince Antunes, Rivalino Matias Jr.
Faculdade de Computação
Universidade Federal de Uberlândia
Uberlândia, Brasil
mahprince@mestrado.ufu.br, rivalino@fc.ufu.br

***Resumo*— De acordo com a literatura, falhas de sistemas computacionais são causadas principalmente por defeitos de software. Revisando os trabalhos publicados na área de confiabilidade de software, observa-se que pesquisas em confiabilidade de sistemas operacionais não são frequentes. Nota-se que de nada adianta um sistema computacional com hardware e aplicações altamente confiáveis, se o SO não apresentar nível de confiabilidade equivalente. Este trabalho apresenta um estudo experimental sobre a confiabilidade de um sistema operacional de propósito geral. Foi analisada uma amostra com 3.235 registros reais de falhas de SO, obtidos de diferentes computadores e ambientes de trabalho. Com base nas principais causas de falhas identificadas, foram criados modelos estocásticos de confiabilidade que permitiram avaliar a sensibilidade da confiabilidade do sistema operacional com respeito a cada uma das causas de falhas observadas.**

Palavras-chave—sistema operacional; confiabilidade; estudo de caso

I. INTRODUÇÃO

Sistemas computacionais têm se tornado uma das mais importantes ferramentas da sociedade moderna. Esta crescente dependência faz com que as falhas destes sistemas tenham impacto significativo, podendo variar de simples inconvenientes até danos catastróficos [1]. Sabe-se com base na literatura (ex. [2], [3]) que falhas de sistemas computacionais são causadas principalmente por defeitos de software, o que tem feito pesquisas em engenharia de confiabilidade de software [4], [5] ganharem cada vez mais importância.

Ressalta-se que na maioria dos sistemas computacionais, o nível de software subdivide-se em programas de aplicação e o software de sistema operacional (SO). Revisando a literatura de confiabilidade de software experimental, observa-se que trabalhos voltados para a confiabilidade do software de SO (ex. [6], [7], [8], [9], [10], [11], [12]) são significativamente em menor número. Considerando que a execução das aplicações depende do SO, de nada adianta um sistema computacional com hardware e aplicações altamente confiáveis, sendo que o software de SO não apresenta a confiabilidade necessária. A Fig. 1 ilustra esse cenário, onde a confiabilidade do sistema computacional é diminuída pela menor confiabilidade do SO, apesar dos demais elementos apresentarem alta confiabilidade (99,999%).

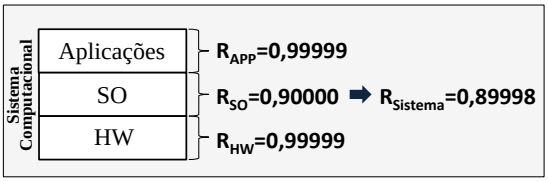

Fig. 1. Confiabilidade sistêmica.

Este trabalho apresenta um estudo experimental sobre a confiabilidade de software de sistema operacional. Especificamente, avaliou-se um sistema operacional de propósito geral, pois cada vez mais este tipo de SO tem sido usado em sistemas com requisitos de alta confiabilidade. Como exemplo, de acordo com [13], os sistemas de navegação da marinha dos Estados Unidos e os sistemas de controle e automação de usinas de energia da ABB executam sob o sistema operacional MS Windows.

Neste contexto, este trabalho estende dois estudos anteriores. Em [14], foi proposta uma abordagem de análise de confiabilidade de sistema operacional do ponto de vista do usuário. Até então, diferentes trabalhos nesta área (ex. [6], [7], [9], [12]) consideravam como falha de SO apenas ocorrências de falhas provenientes do núcleo do SO (falhas de *kernel*). O problema desta abordagem é que ela não reflete a real experiência do usuário no tocante à percepção da falha. Por exemplo, uma falha em um serviço de atualização do SO, implementado como um processo de aplicação executando no espaço do usuário, não seria contabilizada como falha do SO, apesar de impactar a operação do sistema e, por conseguinte, a experiência do usuário. Neste caso, a falha poderia corromper arquivos críticos do SO, deixando-o instável ou até mesmo interrompendo sua operação (*crash/hang*). Não importa para o usuário se a falha ocorreu em uma aplicação no espaço do usuário ou em rotinas no espaço do *kernel*, o fato é que do ponto de vista do usuário o sistema operacional parou (parcialmente ou totalmente) de funcionar. Assim, em [14] propomos uma abordagem onde falhas de SO não são apenas de *Kernel*, mas também de *Aplicações de SO* e *Serviços de SO*. Esta abordagem foi posteriormente utilizada em [15], onde foi realizado um estudo exploratório abrangente para caracterização de falhas de SO, considerando registros de falhas de 735 computadores reais. Complementando os trabalhos anteriores, neste artigo são investigadas as principais causas de falhas em cada uma das três categorias de falhas de

SO, tendo como base a amostra de falhas discutida em [15]. Também, avaliou-se quantitativamente a importância de cada uma das causas identificadas sobre a confiabilidade geral do sistema operacional investigado.

As próximas seções deste trabalho estão organizadas como segue. A Seção II apresenta a metodologia adotada, descrevendo a abordagem e material utilizados. A Seção III apresenta os resultados das diferentes análises realizadas. A Seção IV relata as conclusões do trabalho.

II. METODOLOGIA

A. Abordagem

Como citado na Seção I, neste trabalho seguiu-se o enfoque apresentado em [14], onde falhas de SO são categorizadas em três tipos: *Kernel* (SO_{KNL}), *Aplicações de SO* (SO_{APP}) e *Serviços de SO* (SO_{SVC}). O segundo e terceiro tipos de falhas ocorrem em processos do sistema operacional que realizam tarefas administrativas, executando no espaço do usuário. Suas falhas implicam na interrupção ou degradação de funcionalidades do SO. Por exemplo, uma importante aplicação de SO é o gerenciador de janelas (*window manager*), existente em sistemas operacionais que possuem interface gráfica. São exemplos de gerenciador de janelas o *Explorer.exe* (MS Windows) e o *Kwin* (Linux). Uma falha desta aplicação pode, por exemplo, impedir o usuário de executar programas, acessar arquivos ou até mesmo movimentar o ponteiro do mouse. Nestas situações, do ponto de vista do usuário, o SO falhou, independente se o *kernel* continua funcionando e a falha ocorreu em uma aplicação no espaço do usuário. Similarmente ocorre com os Serviços de SO. Deste modo, neste trabalho foi realizada uma análise de dados estratificada, considerando cada uma destas três categorias de falhas.

Com base em uma ampla amostra de dados de falhas de SO, apresentada na próxima seção, para cada categoria de falhas de SO foram identificadas suas causas principais. Ou seja, quais componentes (programas ou partes do *kernel*) mais provocaram falhas em cada categoria. Devido ao grande volume de dados envolvidos nesta análise, esse processo de mineração foi automatizado com programas criados para esse propósito. Posteriormente, realizou-se uma análise de caracterização destes dados, estratificados por categoria, a fim de se conhecer as propriedades estatísticas dos três tipos de falhas considerados. Com base nesta análise, foram estimadas as distribuições de probabilidade dos tempos entre falhas para cada componente identificado. Para se selecionar as distribuições com boa aderência aos tempos entre falhas, foi aplicado o teste de aderência Kolmogorov-Smirnov (K-S) [16], ao nível de significância (α) de 5%. Os modelos de distribuição testados foram: Beta, Exponencial, Gama, Levy, Logística, Loglogística, Lognormal, Normal, Rayleight, Valor extremo e Weibull [17], [18]. Com base nas distribuições de probabilidade selecionadas, foi criado um modelo estocástico para análise de confiabilidade do sistema operacional investigado. O método de modelagem de confiabilidade adotado foi o diagrama de blocos de confiabilidade (RBD – *Reliability Block Diagram*) [17]. O modelo RBD proposto foi elaborado de forma hierárquica [18], em dois níveis, onde o nível superior (*top level*) é formado por blocos representando as três categorias de falhas de SO (ver Seção III.C). No segundo nível, tem-se cada categoria sendo modelada por blocos representando os principais componentes do SO, da respectiva categoria, que mais causaram falhas observadas na amostra de trabalho. Para desenhar e analisar o RBD proposto foi utilizada a ferramenta SHARPE [19], [20] que suporta, dentre outras funcionalidades, o uso de RBD hierárquico.

Com base no modelo de confiabilidade construído, foram calculadas métricas de confiabilidade e realizada uma análise de sensibilidade da confiabilidade do sistema operacional com respeito aos seus componentes modelados. Esta análise teve como base o índice de importância de confiabilidade [18] destes componentes para cada categoria de falha.

B. Material

Como informado na Seção I, este estudo utiliza a amostra de falhas de SO de [15]. Esta amostra é composta de 30.815 falhas de SO obtidas de 735 computadores reais. Todas as falhas são provenientes de computadores com o sistema operacional MS Windows 7 (Win7). Os detalhes sobre a estratégia de amostragem e o instrumental usado na coleta das falhas podem ser encontrados em [15]. Esta amostra está organizada em seis grupos (G1 a G6). As falhas dos primeiros três grupos (G1, G2, G3) foram coletadas de computadores localizados em laboratórios de ensino de universidades. Já os dados dos grupos G4 e G5 são provenientes de ambientes corporativos. O G6 agrupa falhas de sistemas de diferentes ambientes de trabalho, inclusive, localizados em diferentes países. A Tabela I apresenta uma visão geral dos grupos.

TABELA I. RESUMO DAS AMOSTRAS DE FALHAS DE SO

	G1	G2	G3	G4	G5	G6
Período de coleta (dias)	269	72	332	391	378	891
Total de computadores	5	63	268	275	41	83
Total de falhas de SO	284	406	6844	19725	548	3008
% SO_{KNL}	1,76	3,45	15,04	2,53	7,86	9,21
% SO_{APP}	0,35	3,94	0,89	20,55	9,87	5,65
% SO_{SVC}	97,89	92,61	84,07	76,92	82,27	85,14
Falhas por dia (média)	1,06	5,64	20,61	50,45	1,45	3,38

A fim de identificar as principais causas de falhas em cada categoria, todos os grupos foram analisados e verificou-se que os computadores com maior quantidade de falhas nas três categorias se concentram nos grupos G4 e G6. Portanto, foram selecionados os cinco computadores com maior número de falhas por categoria, de cada um dos dois grupos, para compor a amostra de trabalho que será analisada na próxima seção.

III. RESULTADOS

A. Análise das Causas de Falhas

Com base nos dados das falhas dos computadores selecionados para a amostra de trabalho (ver Seção II.B), investigou-se suas principais causas. Com relação a SO_{APP} e SO_{SVC}, o resultado apontou para quatro componentes do SO que mais causam falhas nestas categorias, os quais estão listados na Tabela II.

TABELA II. PRINCIPAIS CAUSAS DE FALHAS DE SO_{APP} e SO_{SVC}

	SO_{APP}	SO_{SVC}
Explorer.exe	X	
Rundll32.exe	X	
Atualizações do Internet Explorer		X
Atualizações do Win7		X

Na categoria SO_{APP}, o *Explorer.exe* [21] foi responsável por 84,16% das 101 falhas observadas na amostra de trabalho para esta categoria. O segundo componente do SO que mais falha nesta categoria foi o *Rundll32.exe*, responsável pelos 15,84% restantes. O *Explorer.exe* (EXP) é uma aplicação do Win7 responsável por duas funções importantes no sistema operacional: gerenciamento de janelas e arquivos. A primeira implementa a área de trabalho (*desktop*) na interface gráfica do Win7. A segunda é usada para acesso e navegação em unidades de armazenamento secundário e de rede, em nível de arquivos e diretórios. Ambas as funções são denominadas no jargão desta plataforma de *Windows shell*. Sobre o *Rundll32.exe* (RDL), este é um programa que permite carregar e executar funções exportadas de bibliotecas ligadas dinamicamente (DLL's) [22].

Na categoria de falhas SO_{SVC}, a causa predominante foi o serviço de atualização do Win7 (Windows *update*) [23], o qual foi responsável por 99,49% das 2.769 falhas observadas na amostra desta categoria. Este serviço realiza a atualização de diferentes componentes de software do Win7, contudo, foram predominantes as falhas de atualizações do Internet Explorer (AIE) e do próprio Win7 (AW7), onde a primeira representou 80,43% e a segunda 19,06% das falhas desta categoria.

Diferente das duas categorias descritas anteriormente, as falhas de *kernel* (SO_{KNL}) apresentaram uma maior distribuição entre suas principais causas. No total foram 16 causas de falhas de *kernel*. Neste estudo foram selecionadas cinco causas (ver Tabela III), as quais representam 61,53% das 169 falhas de *kernel* observadas na amostra.

TABELA III. PRINCIPAIS CAUSAS DE FALHAS DE *KERNEL* NA AMOSTRA

Causas de Falhas de *Kernel* (SO_{KNL})	Qtd	%
DRIVER_POWER_STATE_FAILURE	61	36,09%
NTFS_FILE_SYSTEM	12	7,10%
SYSTEM_SERVICE_EXCEPTION	12	7,10%
SYSTEM_THREAD_EXCEPTION_NOT_HANDLED	11	6,51%
PAGE_FAULT_IN_NONPAGED_AREA	8	4,73%

Esta seleção considerou não apenas o número de ocorrências, mas também se a falha foi observada em múltiplos computadores. Casos onde a ocorrência da falha foi observada em apenas um computador, ou em múltiplos computadores com predominância em apenas um sistema, não foram selecionados. Por exemplo, a causa de falha VIDEO_TDR_ERROR apareceu 23 vezes, contudo em um único computador, o que foi tratado como *outlier*. Uma descrição detalhada destas falhas foge ao escopo deste trabalho, sendo que maiores detalhes podem ser obtidos em [24].

Por questões de praticidade, daqui em diante estas cinco causas de falhas de *kernel* serão referenciadas como KF1 até KF5, onde KF1 representa DRIVER_POWER_STATE_ FAILURE, e assim sucessivamente, visando simplificar seu uso no decorrer do texto.

B. Propriedades Estatísticas

As Tabelas IV e V apresentam o resumo estatístico dos tempos entre falhas, em horas, das causas de falhas das três categorias analisadas.

TABELA IV. RESUMO DOS TEMPOS ENTRE FALHAS (SO_{APP} e SO_{SVC})

	EXP	RDL	AIE	AW7
Mínimo	0	215,86	0	0
Máximo	4012,81	4581,34	459,8	1319,84
Média	540,87	1284,45	13,27	22,12
Desvio padrão	869,27	1354,25	27,87	65,41

TABELA V. RESUMO DOS TEMPOS ENTRE FALHAS (SO_{KNL})

	KF1	KF2	KF3	KF4	KF5
Mínimo	2,29	17,65	23,02	53,86	43,99
Máximo	938,73	1124,88	695,12	1221,24	1973,26
Média	174,15	294,60	250,48	333,29	658,90
Desvio padrão	161,76	353,46	201,56	368,13	700,52

Interessante notar que o tempo médio entre falhas da categoria SO_{APP} é claramente superior ao das demais categorias, onde SO_{SVC} apresenta os menores tempos médios entre falhas. Alguns casos (EXP, AIE e AW7) apresentam tempos entre falhas com valor zero, o que indica múltiplas falhas sucessivas, uma imediatamente após a outra, daqueles componentes.

Como descrito na Seção II.A, em seguida o teste K-S foi utilizado para selecionar as distribuições de probabilidade com aderência aos tempos entre falhas de cada categoria. Com base na análise numérica do resultado do K-S e na análise gráfica do ajuste das distribuições, foram escolhidos os modelos de distribuição de probabilidade listados nas Tabelas VI e VII. Coincidentemente, para todos os casos o modelo exponencial apresentou boa aderência aos dados de falha. Portanto, esta distribuição de probabilidade foi utilizada na análise de confiabilidade apresentada nas próximas seções.

TABELA VI. DISTRIBUIÇÕES DOS TEMPOS DE FALHA (SO_{APP} e SO_{SVC})

	Distribuição Escolhida
EXP	Exponencial (λ = 0,0018)
RDL	Exponencial (λ = 0,0009)
AIE	Exponencial (λ = 0,0754)
AW7	Exponencial (λ = 0,0452)

TABELA VII. DISTRIBUIÇÕES DOS TEMPOS DE FALHA (SO_{KNL})

	Distribuição Escolhida
KF1	Exponencial (λ = 0,0058)
KF2	Exponencial (λ = 0,0036)
KF3	Exponencial (λ = 0,0044)
KF4	Exponencial (λ = 0,0036)
KF5	Exponencial (λ = 0,0016)

C. *Modelagem de Confiabilidade*

Como descrito na Seção II.A, o método de modelagem de confiabilidade adotado foi o RBD hierárquico, construído em dois níveis. Considerando as três categorias de falhas de SO analisadas, o primeiro nível do RBD foi composto de três blocos de confiabilidade, arranjados em série do ponto de vista de confiabilidade, onde cada bloco representa uma das três categorias de falhas de SO (SO_{KNL}, SO_{APP} e SO_{SVC}). Desse modo, falhas em qualquer uma destas categorias implicam na falha do SO. A Fig. 2 apresenta o primeiro nível do RBD implementado no SHARPE. As Figuras 3, 4 e 5 apresentam os RBD's de segundo nível para cada categoria de falha de SO.

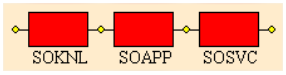

Fig. 2. Primeiro nível do RBD do SO analisado.

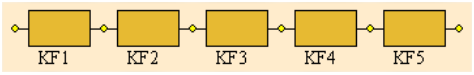

Fig. 3. Segundo nível do RBD da categoria SO_{KNL}.

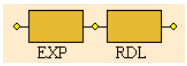

Fig. 4. Segundo nível do RBD da categoria SO_{APP}.

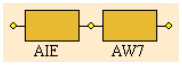

Fig. 5. Segundo nível do RBD da categoria SO_{SVC}.

Importante salientar que neste trabalho considerou-se que todas as falhas de SO, dentro da mesma categoria e entre categorias, ocorrem de forma independente.

D. *Cálculo da Confiabilidade*

Para calcular a confiabilidade do sistema, R_{SO}, primeiramente foi calculada a confiabilidade individual de cada bloco, em todas as categorias, bem como a combinação deles em série na respectiva categoria. A confiabilidade de um dado bloco, *b*, é calculada a partir da função confiabilidade, $R_b(t)$, estimada para o bloco. Neste modelo, a distribuição exponencial foi usada em todos os casos e a sua função confiabilidade é dada por (1).

$$R_b(t) = P(X > t) = \int_t^{\infty} \lambda e^{-\lambda t}\, dt = e^{-\lambda t} \quad (1)$$

onde o parâmetro λ é estimado a partir da amostra de tempos entre falhas (ver Tabelas VI e VII).

Com base na $R_b(t)$ de cada bloco, se calculou suas confiabilidades no segundo nível. Como exemplo, tem-se a Equação 2 para o cálculo da confiabilidade do componente EXP na categoria SO_{APP}. Todos os demais componentes foram calculados da mesma forma, variando apenas o valor do parâmetro λ como apresentado nas Tabelas VI e VII.

$$R_{EXP}(t) = e^{-0,0018t} \quad (2)$$

Considerando que os blocos de confiabilidade em cada categoria de falha são arranjados em série, a confiabilidade de cada categoria foi calculada pelas Equações 3, 4 e 5.

$$R_{KNL}(t) = R_{KF1}(t) \times R_{KF2}(t) \times R_{KF3}(t) \times R_{KF4}(t) \times R_{KF5}(t) \quad (3)$$

$$R_{APP}(t) = R_{EXP}(t) \times R_{RDL}(t) \quad (4)$$

$$R_{SVC}(t) = R_{AIE}(t) \times R_{AW7}(t) \quad (5)$$

Portanto, tem-se a confiabilidade do SO calculada pela Equação 6.

$$R_{SO}(t) = R_{KNL}(t) \times R_{APP}(t) \times R_{SVC}(t) \quad (6)$$

A confiabilidade do sistema, bem como dos seus componentes, foi calculada para os seguintes tempos de missão: 8, 12, 24, 48, 168 e 720 horas. Os resultados da confiabilidade por componente estão apresentados nas Tabelas VIII e IX.

TABELA VIII. CONFIABILIDADE DOS COMPONENTES DE SO_{APP} e SO_{SVC}

Tempo (h)	$R_{EXP}(t)$	$R_{RDL}(t)$	$R_{AIE}(t)$	$R_{AW7}(t)$
8	0,9857	0,9928	0,5471	0,6966
12	0,9786	0,9893	0,4046	0,5814
24	0,9577	0,9786	0,1637	0,3380
48	0,9172	0,9577	0,0268	0,1142
168	0,7390	0,8597	0,0000	0,0005
720	0,2736	0,5231	0,0000	0,0000

TABELA IX. CONFIABILIDADE EM NÍVEL DE *KERNEL*

Tempo (h)	$R_{KF1}(t)$	$R_{KF2}(t)$	$R_{KF3}(t)$	$R_{KF4}(t)$	$R_{KF5}(t)$
8	0,9547	0,9716	0,9654	0,9716	0,9873
12	0,9328	0,9577	0,9486	0,9577	0,9810
24	0,8701	0,9172	0,8998	0,9172	0,9623
48	0,7570	0,8413	0,8096	0,8413	0,9261
168	0,3774	0,5462	0,4775	0,5462	0,7643
720	0,0154	0,0749	0,0421	0,0749	0,3160

Com base nas confiabilidades estimadas para os componentes de cada categoria, a confiabilidade do SO foi calculada para os seis tempos de missão considerados. A Fig. 6 mostra a curva de confiabilidade estimada para o SO analisado, bem como a confiabilidade de cada categoria.

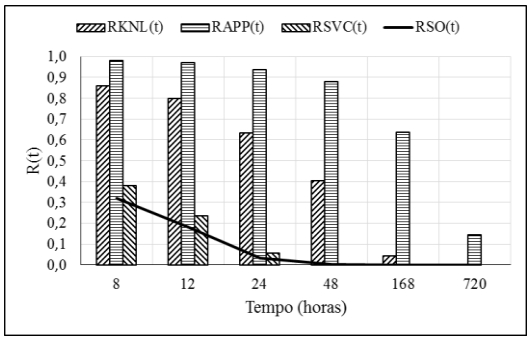

Fig. 6. Curva de confiabilidade do SO analisado.

A Fig. 7 compara a confiabilidade do SO considerando a abordagem tradicional (apenas falhas de *kernel*) com a adotada neste estudo que considera três categorias de falhas de SO.

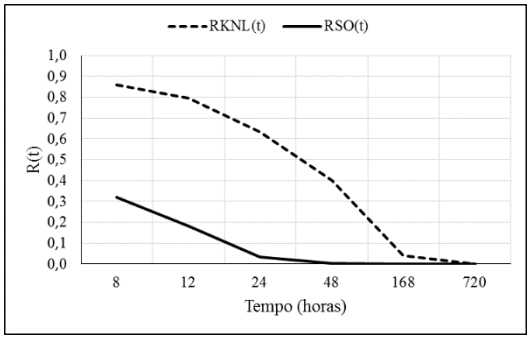

Fig. 7. Comparação de curvas de confiabilidade.

Fica claro que na abordagem tradicional (curva tracejada), a qual considera apenas falhas de *kernel*, a confiabilidade do SO se apresenta superior. Contudo, este resultado não é representativo da experiência do usuário, pois do seu ponto de vista a confiabilidade do SO é afetada por qualquer tipo de falha que cause a degradação ou interrupção de suas funções ou serviços; e não apenas daquelas que estão circunscritas ao núcleo do SO. Deste modo, a segunda curva (sólida) melhor representa esta percepção de confiabilidade por parte dos usuários. Isso pode ser observado na prática, para qualquer SO, comparando as informações de confiabilidade divulgadas pelo fabricante (quase sempre baseadas na abordagem tradicional) com a percepção real dos usuários daquele sistema.

E. Índices de Importância de Confiabilidade

Outra análise que foi realizada refere-se ao índice de importância relativa de cada componente, I_R, com respeito à confiabilidade do sistema. Este índice permite identificar os componentes que têm maior impacto na redução da confiabilidade do sistema, bem como mensurar seus efeitos sobre a confiabilidade total.

A importância de confiabilidade de um componente *i* em um sistema de *n* componentes é dada pela Equação 7 [25].

$$I_{R_i}(t) = \frac{\partial R(t)}{\partial R_i(t)} \tag{7}$$

onde $R(t)$ é a confiabilidade do sistema, $R_i(t)$ é a confiabilidade do componente *i*, e *t* é o tempo de missão sendo considerado. No caso do modelo RBD proposto neste trabalho, o cálculo do I_{R_i} é realizado inicialmente tomando as derivadas parciais da Equação 6 em relação a confiabilidade de cada categoria analisada. As Equações 8, 9 e 10 foram usadas para este cálculo, cujos resultados estão listados na Tabela X para cada tempo de missão considerado.

$$I_{R_{KNL}}(t) = \frac{\partial R_{SO}(t)}{\partial R_{KNL}(t)} = R_{APP}(t) \times R_{SVC}(t) \tag{8}$$

$$I_{R_{APP}}(t) = \frac{\partial R_{SO}(t)}{\partial R_{APP}(t)} = R_{KNL}(t) \times R_{SVC}(t) \tag{9}$$

$$I_{R_{SVC}}(t) = \frac{\partial R_{SO}(t)}{\partial R_{SVC}(t)} = R_{KNL}(t) \times R_{APP}(t) \tag{10}$$

TABELA X. ÍNDICE DE IMPORTÂNCIA DE CONFIABILIDADE P/ CATEGORIA

Tempo (h)	$I_{R_{KNL}}$	$I_{R_{APP}}$	$I_{R_{SVC}}$
8	0,3729	0,3273	0,8406
12	0,2277	0,1873	0,7707
24	0,0519	0,0351	0,5940
48	0,0027	0,0012	0,3529
168	0,0000	0,0000	0,0261
720	0,0000	0,0000	0,0000

Note que SO_{SVC} é a categoria que possui maior índice de importância sobre a confiabilidade sistêmica, ou seja, seus componentes causam maior impacto na redução da confiabilidade do SO. A fim de avaliar a importância de cada componente desta categoria, o mesmo procedimento anterior foi realizado para calcular seus índices de importância de confiabilidade. Os resultados estão listados na Tabela XI.

TABELA XI. ÍNDICE DE IMPORTÂNCIA DE CONFIABILIDADE P/ COMPONENTE DA CATEGORIA SO_{SVC}

Tempo (h)	$I_{R_{AIE}}$	$I_{R_{AW7}}$
8	0,5856	0,4599
12	0,4481	0,3119
24	0,2008	0,0973
48	0,0403	0,0095
168	0,0000	0,0000
720	0,0000	0,0000

Em sistemas com um grande número de componentes, o cálculo de importância de confiabilidade permite identificar os componentes prioritários para a melhoria da confiabilidade, já que muitas vezes não é possível ou viável implementar melhorias em todos. No estudo em questão, considerando que as atualizações do IE demonstram maior impacto sobre a confiabilidade do sistema analisado, uma alternativa para aumentar a confiabilidade do SO seria a remoção deste componente. Note que neste caso não adianta apenas usar outro software, como ocorre comumente, pois as falhas observadas são do serviço de atualização quando executado para este componente de software, e não do componente propriamente. Ou seja, enquanto o software alvo das atualizações continuar

instalado, sendo ele utilizado ou não, suas atualizações continuarão sendo processadas pelo serviço de atualização, o que aumenta a probabilidade de falhas deste serviço de acordo com o observado na amostra. A Tabela XII compara a confiabilidade do SO com e sem esse tipo de falha (AIE). A Fig. 8 apresenta as respectivas curvas de confiabilidade.

TABELA XII. CONFIABILIDADE DO SO COM E SEM AIE

Tempo (h)	$R_{SO}(t)$ com AIE	$R_{SO}(t)$ sem AIE
8	0,3203	0,5856
12	0,1813	0,4481
24	0,0329	0,2008
48	0,0011	0,0403
168	0,0000	0,0000
720	0,0000	0,0000

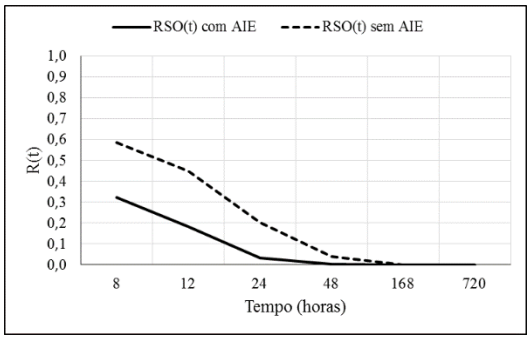

Fig. 8. Comparação de curvas de confiabilidade do SO com e sem AIE.

IV. CONCLUSÕES

Neste artigo foi apresentado um estudo experimental sobre a confiabilidade de um sistema operacional de propósito geral. A metodologia adotada considerou falhas de *Serviços de SO* e de *Aplicações de SO*, além das falhas de *kernel*, para o cálculo da confiabilidade do sistema. As evidências da amostra analisada sugerem que falhas nas duas primeiras categorias têm impacto importante na redução da confiabilidade do sistema, com ênfase para as falhas de serviços. A análise das principais causas de falhas de SO observadas mostra que a atualização de software é o serviço de SO com maior índice de importância de confiabilidade, ou seja, o componente cuja melhoria deve ser priorizada. Neste caso, uma sugestão é reduzir o escopo das atualizações removendo componentes de software que não são necessários, o que ajudaria a reduzir o risco de falhas neste serviço. Para tanto, faz-se importante identificar quais tipos de atualizações causam mais impacto na operação deste serviço, a fim de se implementar tal medida. Neste trabalho, foi dado um exemplo de como isso pode ser feito por meio do cálculo de importância de confiabilidade por categoria de falhas de SO. Um trabalho mais abrangente está sendo desenvolvido para identificar padrões de falhas no MS Windows 7, assim como estender a coleta e análise de dados para as falhas do Linux.

REFERÊNCIAS

[1] N. G. Leveson and C. S. Turner, "An investigation of the Therac-25 accidents," in IEEE Computer, Vol.26:7, pp.18-41, 1993.

[2] M. Sullivan and R. Chillarege, "Software defects and their impact on system availability - a study of field failures in operating systems," in Proc. of the 21th International Symposium on Fault-Tolerant Computing, pp.2-9, 1991.

[3] Z. Li, L. Tan, X. Wang, S. Lu, Y. Zhou, C. Zhai, "Have things changed now? an empirical study of bug characteristics in modern open source software," in Proc. of the 1st Workshop on Architectural and System Support for Improving Software Dependability, pp.25-33, 2006.

[4] M. R. Lyu, "Software reliability engineering: a roadmap," in Proc. of the Future of Software Engineering, pp.153-170, 2007.

[5] J. Xavier, A. Macedo, R. Matias and L. Araújo, "A survey on research in software reliability engineering in the last decade," in Proc. of the 29th ACM Symposium on Applied Computing, pp.1190-1191, 2014.

[6] M. Kalyanakrishnam, Z. Kalbarczyk and R. Iyer, "Failure data analysis of a LAN of Windows NT based computers," in Proc. of the 18th IEEE Symposium on Reliable Distributed Systems, pp.178-187, 1999.

[7] J. Xu, Z. Kalbarczyk and R. Iyer, "Networked Windows NT system field failure data analysis," in Proc. of Pacific Rim International Symposium on Dependable Computing, pp.178-185, 1999.

[8] A. Chou, J. Yang, B. Chelf, S. Hallem and D. Engler, "An empirical study of operating systems errors," in Proc. of the 18th ACM Symposium on Operating Systems Principles, pp.73-88, 2001.

[9] M. M. Swift, B. N. Bershad and H. M. Levy, "Improving the reliability of commodity operating systems," in Proc. of the 19th ACM Symposium on Operating Systems Principles, pp.207-222, 2003.

[10] A. Ganapathi and D. Patterson, "Crash data collection: a Windows case study," in Proc. of the International Conference on Dependable Systems and Networks, pp.280-285, 2005.

[11] A. Ganapathi, V. Ganapathi and D. Patterson, "Windows XP kernel crash analysis," in Proc. of the 20th Conference on Large Installation System Administration, pp.149-159, 2006.

[12] B. Murphy, "Reliability estimates for the Windows operating system," Microsoft Research Cambridge, http://www.dcl.hpi.uni-potsdam.de/meetings/mshpsummit// slides/brendan.murphy.pdf, 2008.

[13] P. L. Li, M. Ni, S. Xue, J. P. Mullally, M. Garzia and M. Khambatti, "Reliability assessment of Mass-Market software: insights from Windows Vista," in Proc. of the 19th International Symposium on Software Reliability Engineering, pp.265-270, 2008.

[14] R. Matias, G. Oliveira and L. Araújo, "Operating system reliability from the quality of experience viewpoint: an exploratory study," in Proc. of the 28th ACM Symposium on Applied Computing, pp.1644-1649, 2013.

[15] R. Matias, M. Antunes, L. Araújo, C. Sousa and L. Henrique, "An empirical exploratory study on operating system reliability," in Proc. of the 29th ACM Symposium on Applied Computing, pp.1523-1528, 2014.

[16] C. Cullen and H. Frey, Probabilistic Techniques in Exposure Assessment: A Handbook for Dealing with Variability and Uncertainty in Models and Inputs, Springer, New York, 1999.

[17] K. S. Trivedi, Probability and Statistics with Reliability, Queuing, and Computer Science Applications, John Wiley and Sons, New York, 2001.

[18] R. Billinton and R. N. Allan, Reliability Evaluation of Engineering Systems: Concepts and Techniques, Springer, New York, 1983.

[19] K. S. Trivedi and R. Sahner, "SHARPE at the age of twenty two," in ACM SIGMETRICS Performance Evaluation Review, Vol.36:4, pp.52-57, 2009.

[20] SHARPE Portal, http://sharpe.pratt.duke.edu

[21] Steven Sinofsky, "Improvements in Windows Explorer", http://blogs.msdn.com/b/b8/archive/2011/08/29/improvements-in-windows-explorer.aspx, 2011.

[22] Microsoft, "Dynamic-Link Libraries", http://msdn.microsoft.com/en-us/library/ms682589.aspx

[23] Microsoft, "Windows Update", http://windows.microsoft.com/en-us/windows/windows-update

[24] Microsoft, "Blue Screens Code Reference", http://msdn.microsoft.com/en-us/library/windows/hardware/hh994433(v=vs.85).aspx

[25] L. M. Leemis, Reliability - Probabilistic Models and Statistical Methods, Prentice Hall, New Jersey, 1995.