|  |  |  |
| --- | --- | --- |
|  | UNIVERSIDADE FEDERAL DO CEARÁ  CENTRO DE TECNOLOGIA  DEPARTAMENTO DE ENGENHARIA DE TELEINFORMÁTICA  CAMPUS DO PICI, CAIXA POSTAL 6007, CEP 60.755-640  FORTALEZA – CEARÁ – BRASIL  FONE (+55) 85 3366-9467 – FAX (+55) 85 3366-9468 |  |

**PROPOSTA DO PROJETO DE FINAL DE CURSO**

**I. Identificação**

|  |
| --- |
| Título: Ferramenta de Diagnóstico de Falhas em Memória para Sistemas Operacionais Linux |
| Aluno: Francisco Plínio Oliveira Silveira |
| Prof. Orientador (Departamento): Alexandre Augusto da Penha Coelho (DETI) |

**II. Objetivo\***

|  |
| --- |
| Principal   1. O objetivo principal é desenvolver um framework com um conjunto de funcionalidades e algoritmos capazes de detectar variados tipos de falhas em módulos de memória RAM em sistemas computacionais baseados em Linux;   Específicos   1. Selecionar os melhores algoritmos para detecção de falhas em memória disponíveis na literatura; 2. Implementar os algoritmos de forma eficiente e flexível; 3. Fazer um software capaz de testar sistemas variados que utilizem Linux; 4. Medir a eficiência dos algoritmos para cada tipo de falha; 5. Disponibilizar o software na forma de um pacote de fácil instalação. |

**III. Descrição\***

|  |
| --- |
| As células de memória RAM estão entre os circuitos VLSI mais densos fabricados atualmente. Devido às linhas dos transistores estarem muito próximas umas das outras, estes circuitos apresentam uma média muito alta de defeitos físicos por unidade de área, comparados com outros tipos de circuitos. Este fato tem motivado pesquisadores a desenvolverem eficientes algoritmos de testes, para memória RAM, que provêem uma boa cobertura de falhas.  Apesar da quantidade de algoritmos disponíveis atualmente, poucas aplicações os utilizam, de fato, para diagnóstico de sistemas finais. Seu uso mais comum é na detecção de falhas de manufatura, sendo executado logo após a fabricação do chip de memória. Muitas vezes, porém, um usuário (doméstico ou empresa) deseja testar uma memória já em uso há alguns anos, ou mesmo diagnosticar periodicamente um componente operando em condições de estresse. Neste caso, as opções são escassas e utilizam algoritmos simples e de baixa cobertura de falhas.  A ferramenta mais utilizada e, talvez, a mais eficiente disponível atualmente é o Memtest86, software open source que executa diretamente no momento do boot. Sua abordagem evita todo o controle que qualquer sistema operacional exerce, permitindo, assim, testar a maior quantidade possível de memória física. Uma desvantagem desta abordagem é a necessidade de interromper toda a atividade do sistema e reiniciá-lo, algo muitas vezes incômodo para máquinas de uso pessoal ou mesmo inviável em sistemas embarcados e servidores. Outra desvantagem está relacionada às características de inicialização do hardware, pois muitas vezes estes não possuem suporte a boot loader através de mídia externa como USB, CD-ROM, Ethernet, dentre outras, inviabilizando o carregamento do Memtest86.  Já uma ferramenta de teste rodando sobre Linux possui as vantagens de ser executado nos mais diversos sistemas computacionais, incluindo desde pequenas plataformas de sistemas embarcados até servidores de grande porte. Além de não ser necessária uma paralisação total da máquina, possibilitando um diagnóstico do componente de memória com o hardware em pleno funcionamento. Esta flexibilidade é conquistada em detrimento da quantidade de memória testada, pois uma aplicação rodando sobre um sistema operacional jamais poderá ter acesso a toda a memória física disponível.  Tendo em vista a utilidade deste tipo de diagnóstico executando sobre sistemas Linux e a falta de soluções tão completas como as já encontradas em forma de ferramentas de inicialização, a proposta é desenvolver um framework com os algoritmos de testes de memórias mais consagrados e agrupá-los em uma aplicação capaz de detectar uma boa variedade de falhas em memórias RAM de sistemas Linux. |

**IV. Metodologia\***

|  |
| --- |
| A metodologia a ser seguida neste trabalho obedecerá as seguintes atividades:   1. Revisão bibliográfica sobre tipos de falhas em memória e algoritmos de detecção; 2. Concepção, projeto e implementação do software de detecção de falhas sobre plataforma Linux; 3. Testes em máquina virtual e validação dos algoritmos; 4. Testes em plataformas variadas (desktops, notebooks, servidores e sistemas embarcados); 5. Análise dos resultados obtidos e comparação com a bibliografia existente; |

**V. Recursos Necessários**

|  |
| --- |
| 1. Computador com compilador C para Linux para o desenvolvimento; 2. Livros e material de consulta na área, proveniente das bibliotecas da UFC e do acervo de publicações cientifica da CAPES e IEEE. 3. Computadores com plataforma Linux para teste, disponíveis no LESC; 4. Máquina virtual Faumachine (capaz de simular erros em memória) disponível gratuitamente na web; 5. Placas de memórias com defeito para teste, disponíveis no LESC. |

\*Não há limite para o número de linhas.

### Fortaleza, 26 de agosto de 2011.

### \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Aluno

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Prof. Orientador

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Visto do Chefe do Departamento

Aprovado pela Coordenação do Curso em / / .

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Coordenador do Curso de Graduação