

Sistemas Distribuídos

Aula 4: Diagnóstico em Nível de Sistema

Prof. Elias P. Duarte Jr.

Universidade Federal do Paraná (UFPR)

Departamento de Informática

www.inf.ufpr.br/elias/sisdis



Sumário

- Definindo Precisamente o Diagnóstico
 - Diagnóstico em Nível de Sistema (*System-level Diagnosis*)
- O Modelo PMC
- Diagnóstico Distribuído
- Diagnóstico Adaptativo
- Diagnóstico Adaptativo e Distribuído
- O Algoritmo Adaptive-DSD (Virtual Ring: VRing)

Diagnóstico: O Problema

- Considere um sistema distribuído S , que consiste de N processos
 - $S = \{0, 1, 2, \dots, (N-1)\}$
 - Cada processo tem um identificador sequencial, os identificadores começam em 0
- Considere que cada processo pode estar em um de dois estados possíveis: **falho** ou **correto**
- O objetivo do Diagnóstico em Nível de Sistema é determinar os estados dos N processos
- Em inglês: *System-level Diagnosis*

Monitoramento Baseado em **Testes**

- Para descobrir os estados dos processos, o Diagnóstico utiliza **testes**, que são executados entre os processos
- O teste consiste da execução de uma bateria de procedimentos que permitem determinar o estado do processo testado
- Como implementar um teste?
 - Totalmente dependente da tecnologia do sistema

O Modelo PMC

- O primeiro modelo de diagnóstico
F. Preparata, G. Metze, R. Chien, “On the Connection Assignment Problem of Diagnosable Systems,” *IEEE Transactions on Computers*, Vol. 16, No. 6, 1967.
- O nome do modelo vem das iniciais dos autores
- O foco principal do modelo PMC e da pesquisa em Diagnóstico dos anos 1970:
 - O grafo de testes (*connection assignment*, no título acima)

O Grafo de Testes

- O grafo de testes $G = (V, T)$ tem como vértices os processos do sistema no conjunto V
- O conjunto de arcos (arestas direcionadas) T , consiste dos testes executados
- Assim, existe um arco (a, b) em T se a e b são vértices pertencentes a V e a testa b
- No modelo PMC o diagnóstico é feito *off-line*: os testes são definidos e executados e depois processados
- O diagnóstico é estático: processos não mudam de estado enquanto os testes são realizados

Síndrome & Unidade Central

- O conjunto de resultados de todos os testes realizados é chamada a *síndrome* do sistema
- A síndrome é processada por uma unidade central, externa ao sistema, que determina então quais processos estão falhos e quais estão corretos
 - Parece uma barbaridade, mas não é! Vc pode ser a unidade central, ou um processo “gerente”, recebe os resultados do testes e daí determina os estados

Modelo de Falhas

- No modelo PMC original: os nodos falhos têm comportamento bizantino
- Em particular: um nodo falho pode “mentir” sobre o resultado de um teste
- Os algoritmos de diagnóstico mais recentes e que vamos estudar aqui assumem falhas *crash!*
- Reforçando: vamos assumir falhas crash

Os Processos Corretos

- Se um processo correto executa um teste, então:
 - Executa o teste corretamente
 - Reporta corretamente o resultado do teste executado

Os Processos Corretos

- Se um processo correto executa um teste, então:
 - Executa o teste corretamente
 - Reporta corretamente o resultado do teste executado
- Parece simples e faz sentido, certo?
- Afinal um processo está correto executa o testes corretamente
- Esta é a principal premissa do modelo PMC

Testes 100% Corretos

- Observe que para um processo correto ser sempre capaz de determinar com 100% de precisão o estado do processo testado...
- O sistema deve ser síncrono
 - Devem haver limites de tempo conhecido para a transmissão de mensagem e execução de tarefas
 - Se os limites não existem: uma resposta de teste que atrasa pode levar à conclusão incorreta de que o processo está falho
- Assim vamos assumir o modelo síncrono

Outras Premissas do Modelo PMC

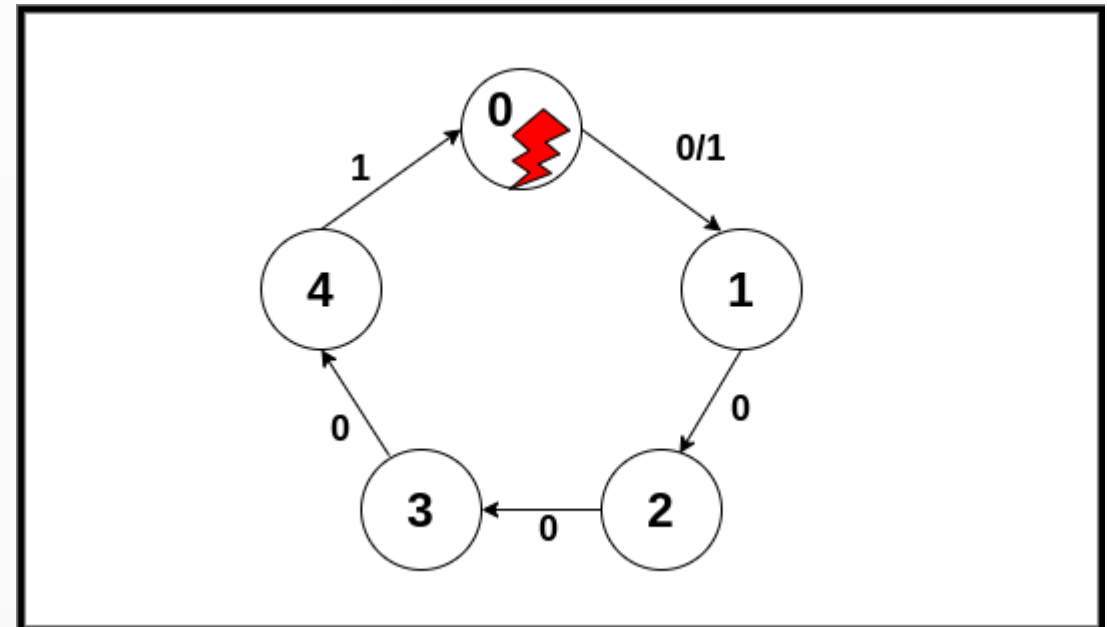
- No modelo PMC os canais de comunicação são perfeitos
- E jamais falham
- Apesar de que existem algoritmos para sistemas de topologia arbitrária (*multi-hop*), vamos assumir sistemas *fully-connected*
 - Qualquer processo é capaz de se comunicar com qualquer outro processo diretamente, sem intermediários

Um Pouco de História de Diagnóstico...

- Nos anos 1970, o foco era no grafo de testes
- Como deveria ser o grafo de testes para permitir o diagnóstico de t processos falhos?
- Conceito: *t-diagnosable systems*
- Lembre-se que naquela época o modelo de falhas assumido era bizantino

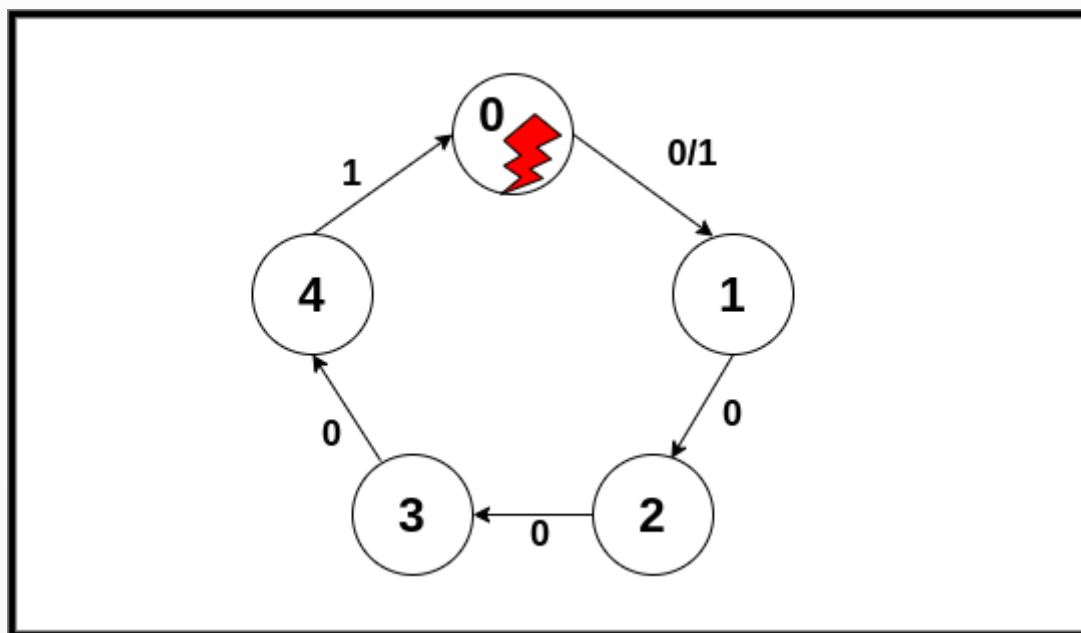
Exemplo do Sistema PCM Original

- No sistema abaixo $N=5$, e o processo 0 está falho
- Considere que apenas o processo 0 está falho, o assinalamento de testes forma o anel mostrado
- Um arco corresponde a um teste, com o resultado:
 - 1 → processo testado falho
 - 0 → processo testado correto



Exemplo do Sistema PCM Original

- A síndrome é: $(?, 0, 0, 0, 1)$
- Com este assinalamento de testes é possível mostrar que no máximo $t=1$ processo pode estar falho (falhas bizantinas)



Exemplo do Sistema PCM Original

- É possível mostrar que no máximo $t=1$ processo pode estar falho (falhas bizantinas)
 - $(0, 0, 0, 0, 1)$ → mostra claramente que o processo está falho, se o processo 4 estivesse falho e mentindo o processo 3 identificaria, e a síndrome seria outra
 - $(1, 0, 0, 0, 1)$ → síndrome indicando duas falhas, ops! Problemas, pois só uma falha ($t=1$) é permitida, quem está falho? Só pode ser o 4 indicando que o 0 está falho, raciocínio!
- Foco da pesquisa: quais síndromes são possíveis? Quais resultados são corretos?

Um Resultado Importante

- Em 1974, Hakimi & Amin provaram que no Modelo PMC, se dois processos nunca se testam mutuamente, as seguintes condições garantem que o sistema é t -diagnosticável:

1) $N \geq 2t + 1$

2) Cada processo é testado por pelo menos outros t processos

S. L. Hakimi, A. T. Amin, "Characterization of Connection Assignment of Diagnosable Systems," *IEEE Transactions on Computers*, Vol. 23, No. 1, 1974

Diagnóstico Distribuído e Diagnóstico Adaptativo

- Em 1984, dois caminhos importantes para o diagnóstico foram propostos por grupos diferentes e publicados na mesma edição da *IEEE Transactions on Computers*:
 - S. L. Hakimi, K. Nakajima, “On Adaptive System Diagnosis,” *IEEE Transactions on Computers*, Vol. 33, No. 3, 1984
 - S. H. Hosseini, J. G. Kuhl, S. M. Reddy, “A Diagnosis Algorithm for Distributed Computing Systems with Dynamic Failure and Repair,” *IEEE Transactions on Computers*, Vol. 33, No. 3, 1984.

Diagnóstico Adaptativo

- Recapitulando, o diagnóstico **não** adaptativo é assim:
 - Define um grafo de testes
 - Os processos se testam uns aos outros, de acordo com o grafo de testes
 - O observador central recebe o conjunto de resultados de todos os testes executados – a síndrome
 - O observador central processa a síndrome e determina os estados de todos os processos do sistema

Diagnóstico Adaptativo

- Por que ficar “atado” a um grafo de testes estático?

Diagnóstico Adaptativo

- Por que ficar “atado” a um grafo de testes estático?
- No diagnóstico adaptativo, a partir do resultados dos testes executados, os processos definem os próximos testes que vão executar!
- O diagnóstico é executado em “rodadas” (*rounds*)

Diagnóstico Adaptativo

- O diagnóstico é executado em “rodadas” (*rounds*)
 - 1) Define-se um grafo de testes inicial, começa na primeira rodada
 - 2) Os nodos executam testes de acordo com o grafo de testes
 - 3) Envia os resultados para o observador central
 - 4) São definidos os novos testes que serão executados na próxima rodada
 - 5) Incrementa a rodada e volta ao item (2)

Diagnóstico Distribuído

- O diagnóstico distribuído acaba com o observador central!

Diagnóstico Distribuído

- O diagnóstico distribuído acaba com o observador central!
- Além de executarem testes...
- ... recebem resultados de testes dos demais testadores, processam os resultados (síndrome) e determinam o estado de todos os nodos

Diagnóstico Distribuído

- Em 1988 foi publicado um algoritmo distribuído para “self-diagnosis”
S. H. Hosseini, J. G. Kuhl, S. M. Reddy, “On Self-Fault Diagnosis of the Distributed Systems,” *IEEE Transactions on Computers*, Vol. 37, No. 2, 1988.
- Bastante caro em termos do número de mensagens necessárias, mas muito inovador!

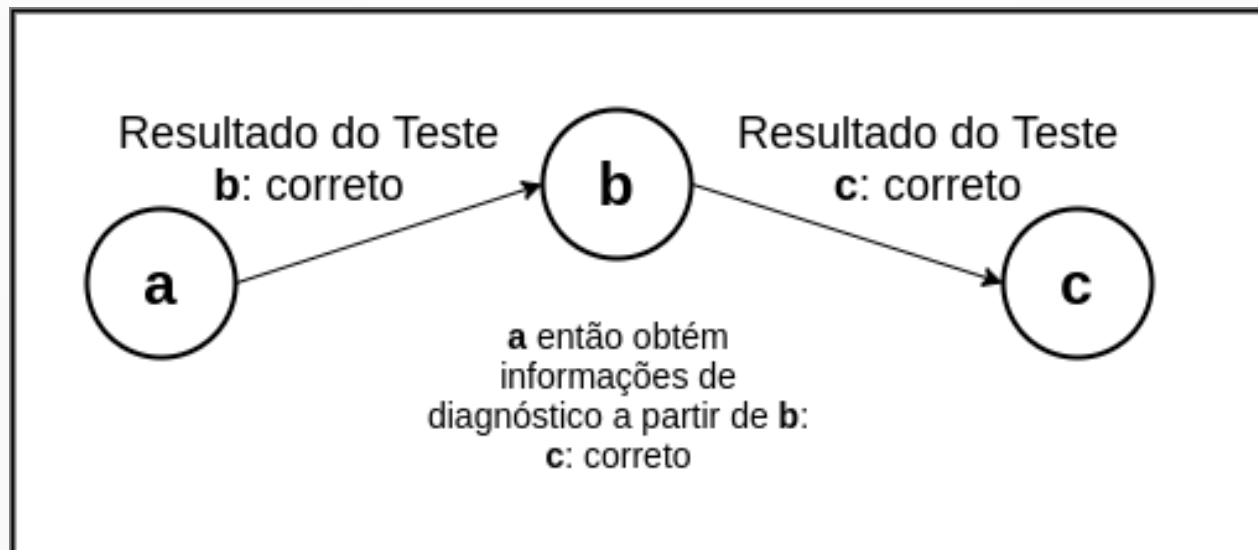
Diagnóstico Adaptativo e Distribuído

- Só em 1991/1992 um grupo de pesquisadores da Universidade de Carnegie Mellon juntou as duas coisas: diagnóstico adaptativo **e** distribuído

R. P. Bianchini Jr., R. W. Buskens, "Implementation of On-Line Distributed System-Level Diagnosis Theory," *IEEE Transactions on Computers*, Vol. 41, No. 5, 1992.

○ Algoritmo *Virtual Ring: VRing*

- Originalmente chamado Adaptive-DSD
- Baseado no modelo PMC
 - Todas as premissas que falamos até aqui, em particular teste perfeito
- Base do algoritmo VRing

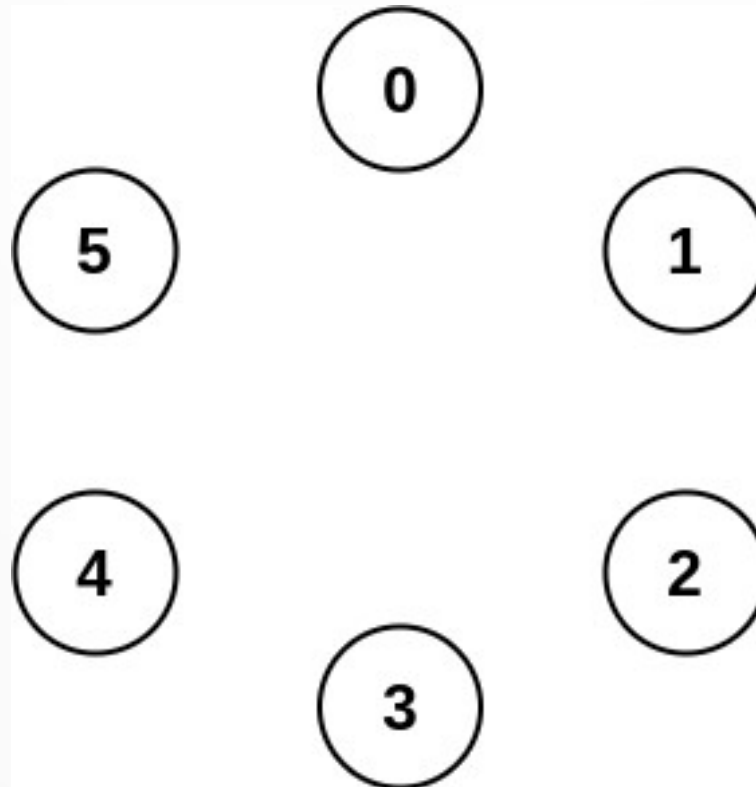


O Algoritmo *Virtual Ring: VRing*

- Executado por N processos com identificadores sequenciais, entre 0 e (N-1)
- Cada processo executa testes sequencialmente, até encontrar um processo correto, ou testar todos os processos falhos
- O processo 0 segue o processo (N-1)

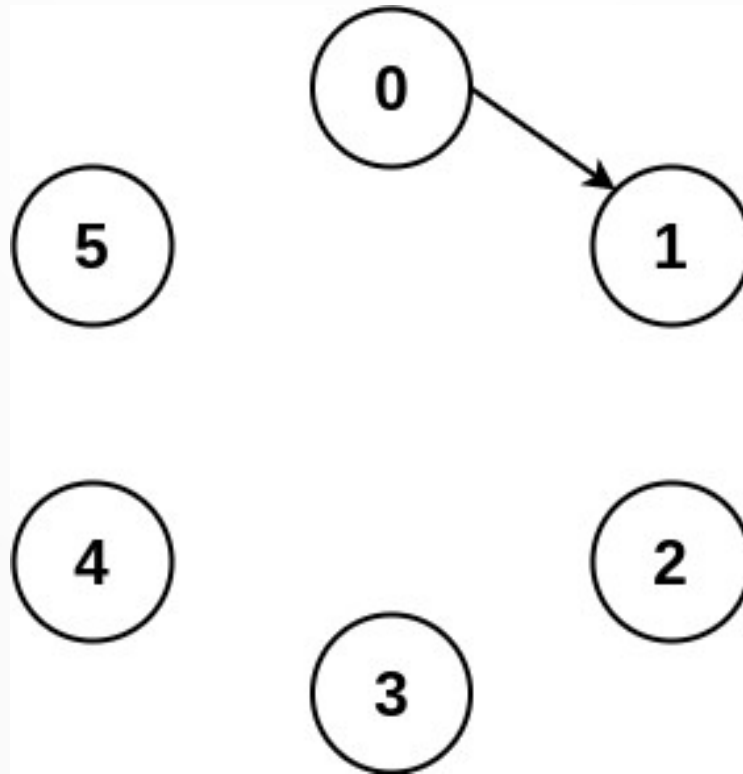
○ Algoritmo *Virtual Ring: VRing*

- Cada processo testa sequencialmente



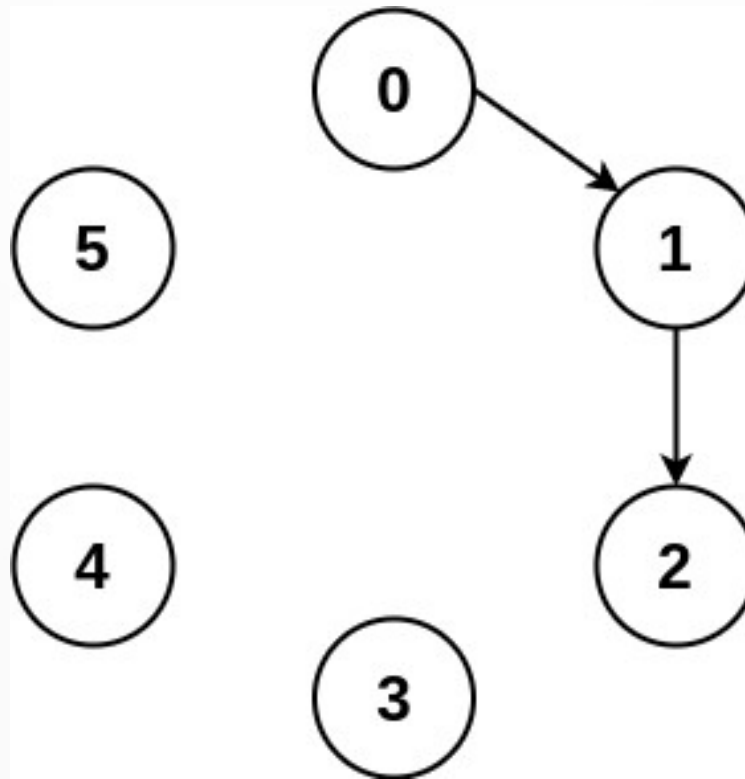
○ Algoritmo *Virtual Ring: VRing*

- Cada processo testa sequencialmente



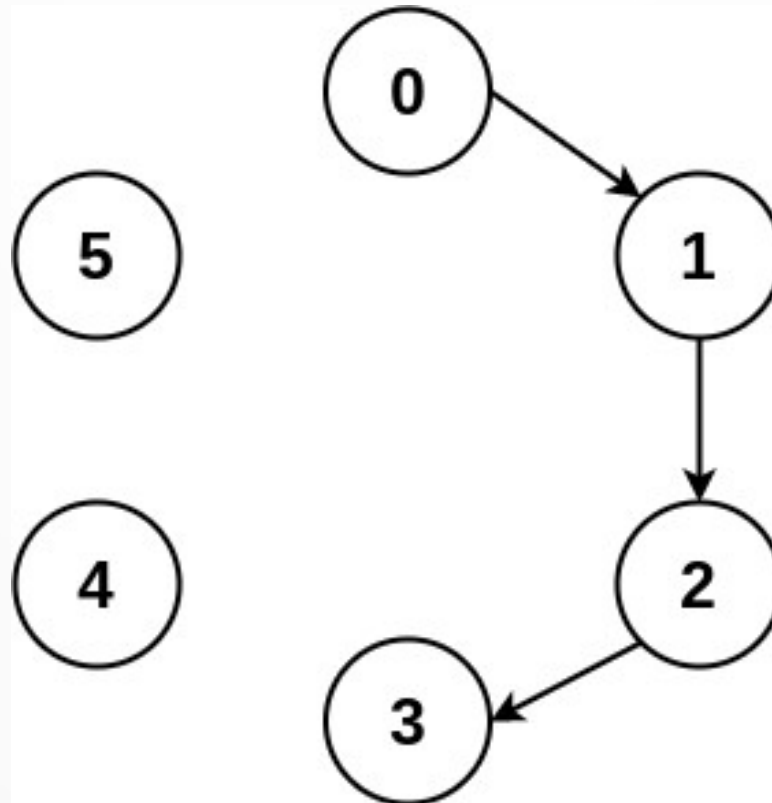
○ Algoritmo *Virtual Ring: VRing*

- Cada processo testa sequencialmente



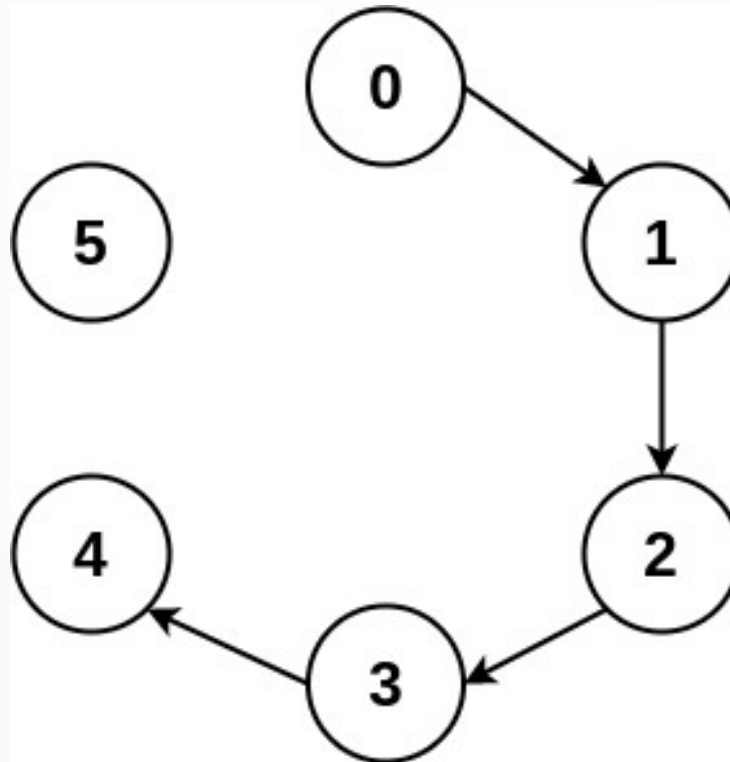
○ Algoritmo *Virtual Ring: VRing*

- Cada processo testa sequencialmente



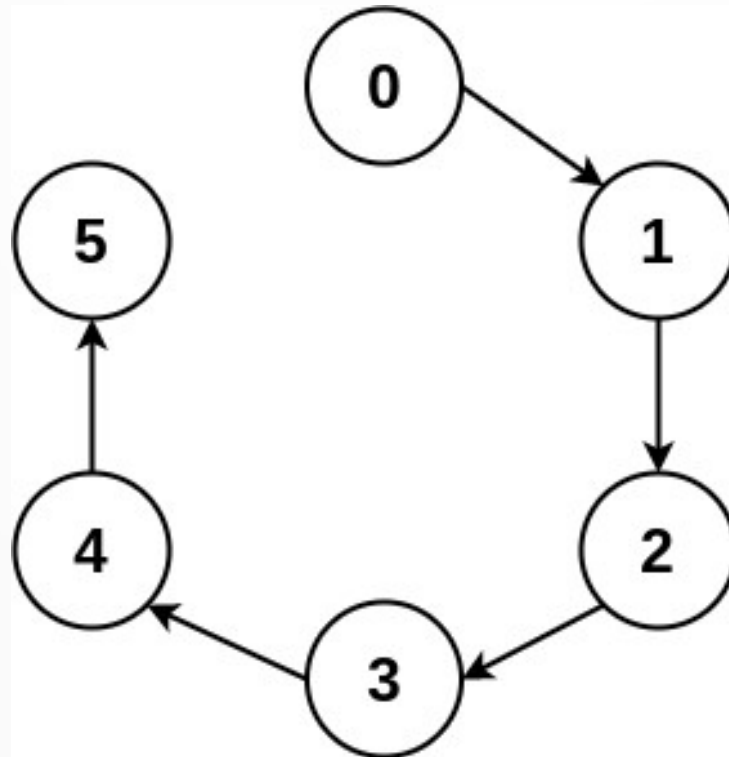
○ Algoritmo *Virtual Ring: VRing*

- Cada processo testa sequencialmente



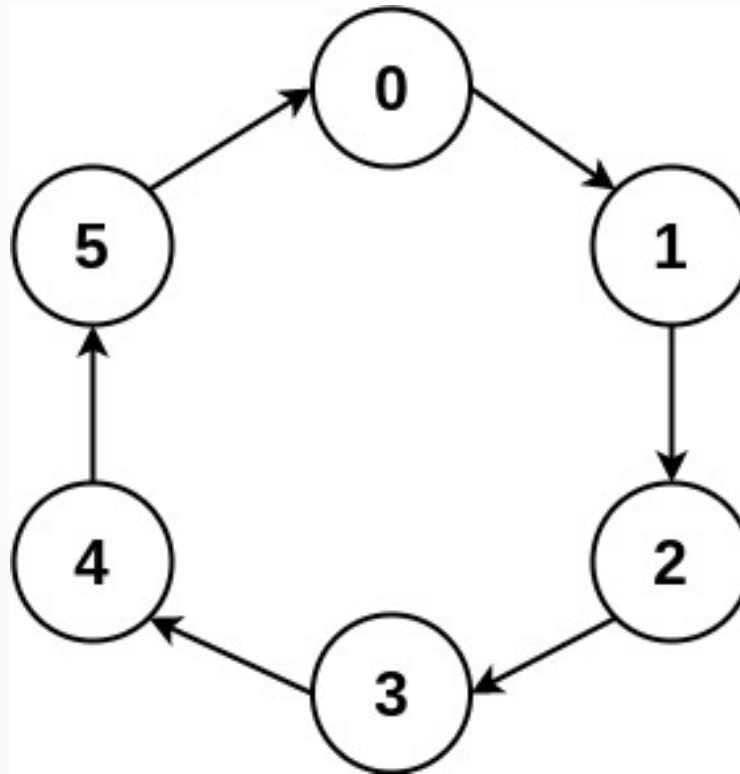
○ Algoritmo *Virtual Ring: VRing*

- Cada processo testa sequencialmente



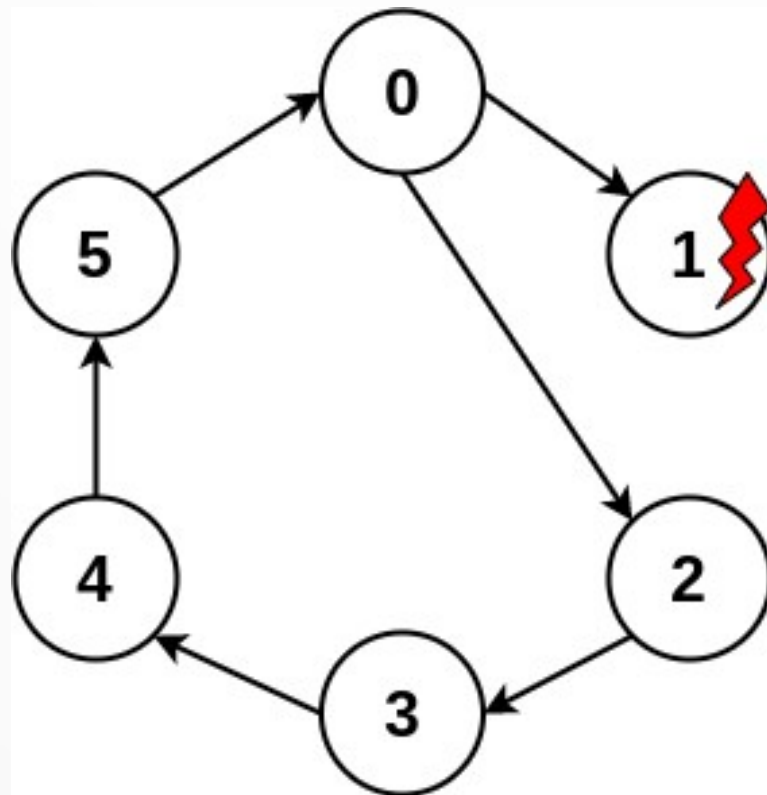
○ Algoritmo *Virtual Ring: VRing*

- Cada processo testa sequencialmente



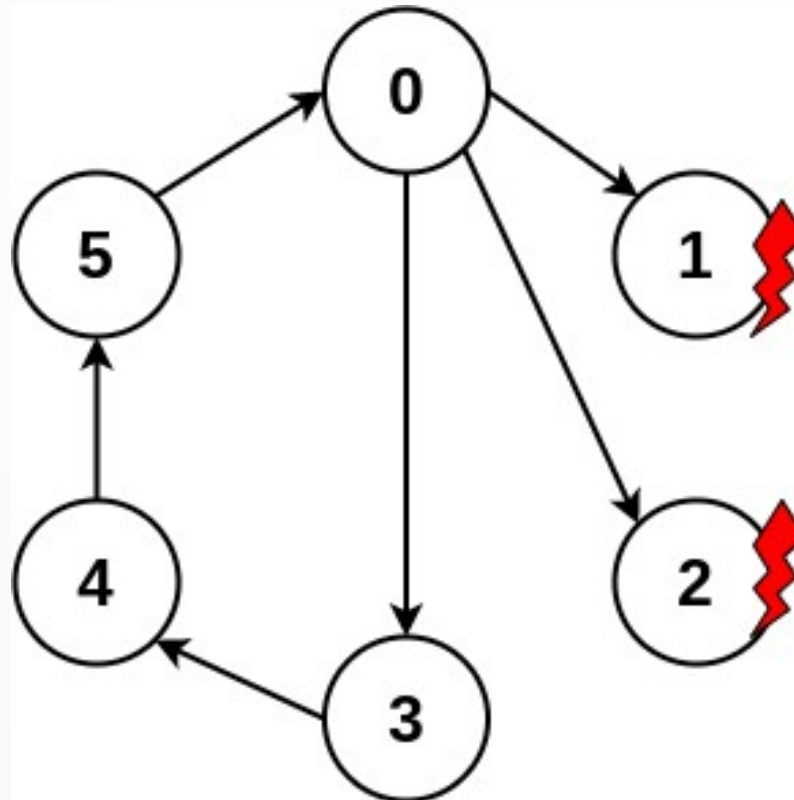
VRing: Testando um Nodo Falho, O Testador Segue até Testar um Correto

- Testes são executados sequencialmente até encontrar um nodo correto: anel que não se rompe



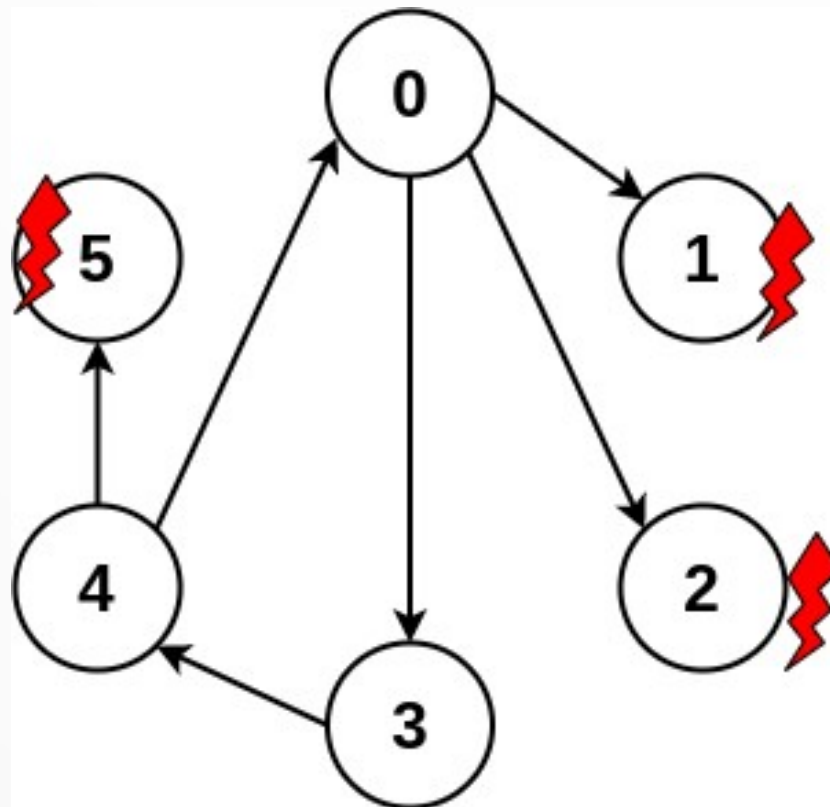
VRing: Testando um Nodo Falho, O Testador Segue até Testar um Correto

- Testes são executados sequencialmente até encontrar um nodo correto: anel que não se rompe



VRing: Testando um Nodo Falho, O Testador Segue até Testar um Correto

- Testes são executados sequencialmente até encontrar um nodo correto: anel que não se rompe



Conclusão

- Diagnóstico em Nível de Sistema
- Modelo PMC
- *Diagnosability*
- Diagnóstico Adaptativo
- Diagnóstico Distribuído
- Diagnóstico Adaptativo e Distribuído: VRing
- Próxima aula: VRing!

Obrigado!
Página da Disciplina
Sistemas Distribuídos:
www.inf.ufpr.br/elias/sisdis