UFRR

Universidade Federal de Roraima
Sistemas Operacionais – DCC403
Philip Mahama Akpanyi

Algoritmo do avestruz

Também conhecido como ***ignorar o problema***, é uma estratégia mais simples, consiste em fazer como se faz uma avestruz diante a uma situação de perigo: colocar a cabeça num buraco e fingir que o problema inexiste. É a solução mais utilizada, pois há baixa probabilidade de ocorrência de deadlock e baixo custo. O UNIX utiliza este método.

Características

- Estratégia mais econômica
- Projetista tem conhecimento de que o problema pode ocorrer, mas finge que não há problemas
- Decisão é tomada levando em consideração frequência com que isto pode ocorrer o problema que isto pode causar
- Desempenho do sistema pode ser prejudicado pela observação
- Rotina de observação pode causar lentidão
- Da ocorrência de deadlock trata-se o problema

Implementação do Algoritmo (Java)

```
final static String RECURSO1 = "Recurso 1";
final static String RECURSO2 = "Recurso 2";

public static void main(String[] args) {

    try {
        Thread t1 = new Thread1();
        Thread t2 = new Thread2();
        t1.start();
        //t1.join();
        t2.start();
        //t2.join();
    } catch (Throwable e) {
        e.printStackTrace();
    }
    JOptionPane.showMessageDialog(null, "Programa entrou em DEADLOCK");


    Thread t3 = new Thread3();
    t3.start();
```

```
}

private static class Thread1 extends Thread {

    @Override
    public void run() {
        synchronized (RECURSO1) {
            try {
                System.out.println("Thread #1: Bloqueou o recurso 1");
                Thread.sleep(100);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }

            System.out.println("Thread #1: Tentando acesso ao recurso 2");
            // Aguarda ate obter o recurso 2.
            synchronized (RECURSO2) {
                System.out.println("Thread #1: Bloqueou o recurso 2");
            }
        }
    }
}

private static class Thread2 extends Thread {

    @Override
    public void run() {
        synchronized (RECURSO2) {
            try {
                System.out.println("Thread #2: Bloqueou o recurso 2");
                Thread.sleep(50);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
            System.out.println("Theread #2: Tentando o acesso ao recurso 1");
            synchronized (RECURSO1) {
                System.out.println("Thread #2: Bloqueou o recurso 1");
            }
        }
    }
}

private static class Thread3 extends Thread {
    public void run() {
        System.out.println("Iniciou Thread 3");
        System.out.println("Thread #3: Entrando em ação");
        JOptionPane.showMessageDialog(null, "Thread #3 no comando");
        JOptionPane.showMessageDialog(null, "Algoritmo do Avestruz implementado com sucesso");
        System.out.println("Finalizou a Thread 3");
        System.exit(0);
    }
}
```

Referências

https://pt.wikipedia.org/wiki/Algoritmo_do_avestruz
http://www.guj.com.br/t/algoritmo-do-avestruz/334234/10