

Erros evitados

Evita que duas tarefas modifiquem o mesmo arquivo ao mesmo tempo.
Diz que escreveu mas não escreveu no arquivo.

Condição de corrida feita com variáveis booleana simulando o lock.

```
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import javax.swing.JOptionPane;

public class main extends Thread {

    Thread threadEscreve01, threadEscreve02;
    boolean rodou = false;
    static String nomeArquivo = "so.txt";

    int ct = 0; //demonstra quantos tentou gravar no arquivo, para ser verificado se gravou a mesma quantidade no
    arquivo

    public static void main(String[] args) {

        System.out.println("O arquivo de texto " + nomeArquivo + " foi criado nessa pasta.\n\nQuantidade de acesso
        ao arquivo:");
        main ob = new main();

        ob.limparArquivo();

        while (true) {

            if (ob.rodou) { //verifica se a thread 2 já escreveu no arquivo
                ob.iniciarEscreve02();
            } else { //escreve no arquivo usando a thread 1
                ob.iniciarEscreve01();
            }
            ob.rodou = !(ob.rodou); //coloca pra a outra thread escrever

            try {
                Thread.sleep(800); //apenas para o while forçar o processamento
            } catch (Exception e) {
            }
        }
    }

    void CriarArquivo() {
        File arquivo = new File(nomeArquivo);
        try (FileWriter fw = new FileWriter(arquivo, true)) {

            fw.flush();
        } catch (IOException ex) {
            ex.printStackTrace();
        }
    }

    void limparArquivo() {
        try {

            Runtime.getRuntime().exec("cmd /c echo.>" + nomeArquivo);
        } catch (Exception ex) {
            JOptionPane.showMessageDialog(null, "Erro", "Erro no arquivo", JOptionPane.ERROR_MESSAGE);
        }
    }
}
```

```

    }
}

void escreve1() {

    escrever("escrita com a thread 1");

}

void escreve2() {

    escrever("escrita com a thread 2");

}

void iniciarEscreve01() {

    threadEscreve01 = new Thread() {
        @Override
        public void run() {
            try {
                escreve1();

            } catch (Exception e) {
                System.out.println(e);
            }
        }
    };
    threadEscreve01.start();

}

void iniciarEscreve02() {

    threadEscreve02 = new Thread() {
        @Override
        public void run() {
            try {
                escreve2();

            } catch (Exception e) {
                System.out.println(e);
            }
        }
    };
    threadEscreve02.start();

}

void escrever(String texto) {
    ct++;
    try {
        Runtime.getRuntime().exec("cmd /c echo." + ct + "escrita: " + texto + " >>" + nomeArquivo);
    } catch (Exception ex) {
        JOptionPane.showMessageDialog(null, "Erro", "Erro no arquivo", JOptionPane.ERROR_MESSAGE);
    }
    System.out.println(ct);
    if (ct>10){//fecha o programa quando executar x vezes
        System.exit(0);
    }
}

}

```