

Watchdog



Tecnologia é uma palavra para descrever uma coisa que ainda não funciona.

Douglas Adams

Problemas com funcionamento do sw

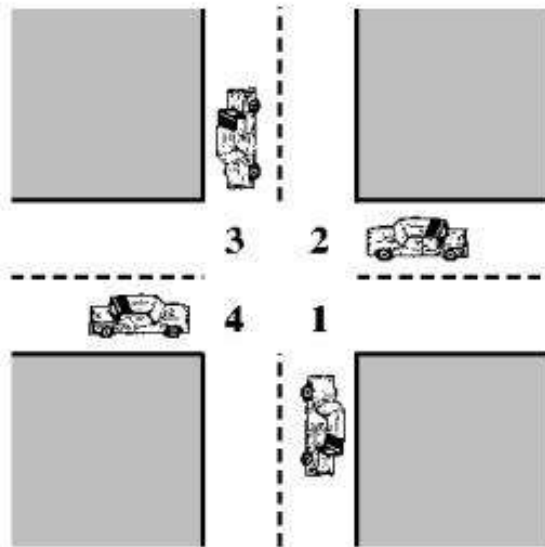
- Situações incômodas/problemáticas num programa que está rodando.
 - Loop infinito.
 - Dead lock.
 - Hack.
 - Bugs/erros nos algoritmos.



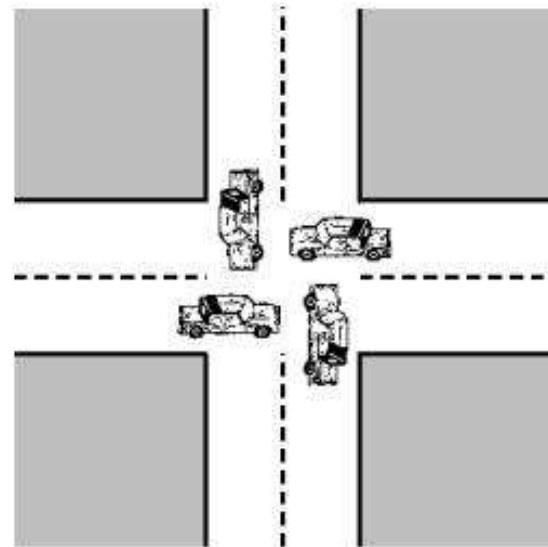
Loop infinito



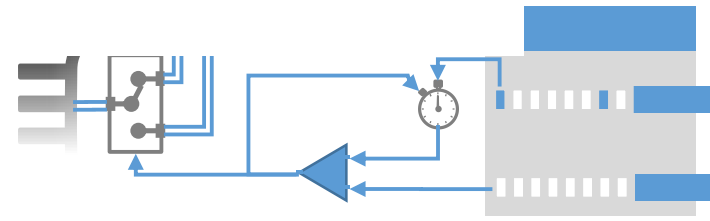
Deadlock



(a) Deadlock possible



(b) Deadlock





DeadLock

- Brigadeiro Faria Lima x Presidente Juscelino (SP)
 - Foto de Silvio Tanaka

000 HACKING TUTORIAL

THE BEST OF 2010

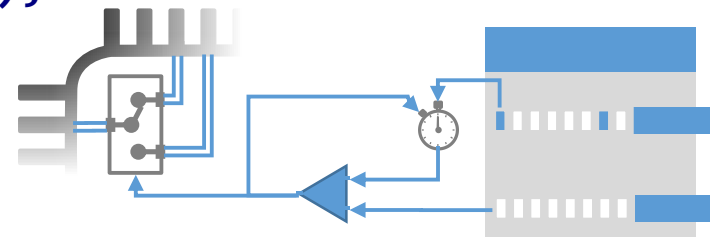
Hack



Exemplo de "Loop infinito"

- Se o botão não for pressionado o while não termina
- Se o while não terminar a função kpDebounce() não é executada
- Se a função kpDebounce() não é executada o sistema não percebe que a tecla mudou
- Se o sistema não perceber que a tecla não mudou, o while não termina

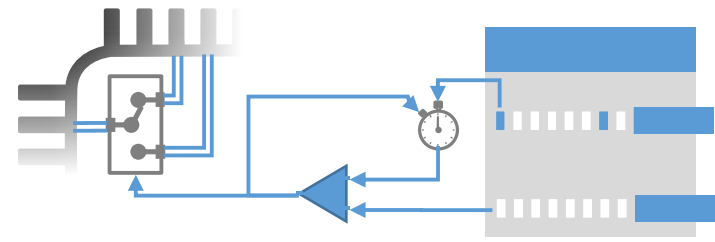
```
void main(void){  
    kpInit();  
    lcdInit();  
    for(;;){  
        kpDebounce();  
        while (kpRead() == 0);  
        lcdPrint(KpRead());  
    }  
}
```



Solução

- Ao invés de ficar preso no while aguardando o pressionamento, podemos apenas perguntar se foi pressionada a cada loop.

```
void main(void){  
    kpInit();  
    lcdInit();  
    for(;;){  
        kpDebounce();  
        if (kpRead() != 0){  
            lcdNumber(kpRead());  
        }  
    }  
}
```



Os Loops infinitos "escondidos"

- O exemplo apresentado é bem pequeno, e a solução relativamente simples
- Sistemas mais complexos podem possuir interdependência de funções que ficam escondidas em sub-rotinas
- Nestes casos garantir que o sistema funciona sem nenhuma condição de deadlock ou loop infinito pode ser difícil
 - Isto é agravado se o sistema precisar de lidar com entradas dos usuários
- Solução: Watchdog



Watchdog

- Visa aumentar a segurança do projeto.
 - Funciona como um temporizador que precisa ter seu contador constantemente reiniciado.
 - Caso não seja reiniciado no tempo exigido, o watchdog reinicia o microcontrolador.
 - Se o código ficar preso em alguma parte do programa, o watchdog é capaz de reiniciar o sistema para verificar o que aconteceu.
-

Watchdog

```
void main(void){
    kpInit();
    lcdInit();
    watchdogInit();
    for(;;){
        kpDebounce();
        if (kpRead() != 0){
            lcdNumber(kpRead());
        }
        watchdogFeed();
    }
}
```

Watchdog

- Quando o sistema é reiniciado, alguns microcontroladores permitem que o programa saiba o porque ele foi reiniciado
 - A causa mais comum é um evento de power-on: a placa acabou de ser ligada
 - Outras causas são: watchdog, alguns tipos de falha, nível de tensão abaixo do permitido, etc.
- No caso do reset por watchdog podemos saber que alguma condição de tempo não foi atendida.
 - Isso pode ser utilizado para indicar um problema com o hardware, enviando uma mensagem para o usuário evitando que um hardware defeituoso continue sendo utilizado.

Watchdog

```
void main(void){
    systemInit();
    //primeiro teste antes da continuidade do programa
    if(watchdogProblem()){
        lcdString("Problema 01");
        for(;;); //não pode continuar execução
    }
    pwmInit();
    serialInit();
    lcdInit();
    watchdogInit();

    for(;;){
        serialProcess();
        pwmProcess();
        watchdogFeed();
    }
}
```

//executar a cada 1ms

