

# Sistemas Operacionais

## Gestão de tarefas - tópicos

Prof. Carlos Maziero

DInf UFPR, Curitiba PR

Julho de 2020

# Conteúdo

**1** Inversão de prioridades

**2** Herança de prioridades

# Inversão de prioridades

## Problema

Uma tarefa de **alta prioridade** impedida de executar por causa de uma tarefa de **baixa prioridade**.

Pode ocorrer se:

- A tarefa de alta prioridade precisa de um recurso.
- Esse recurso está com a tarefa de baixa prioridade.
- A CPU está ocupada com outras tarefas.
- A tarefa de baixa prioridade não consegue executar.

# Inversão de prioridades

Considere um sistema com:

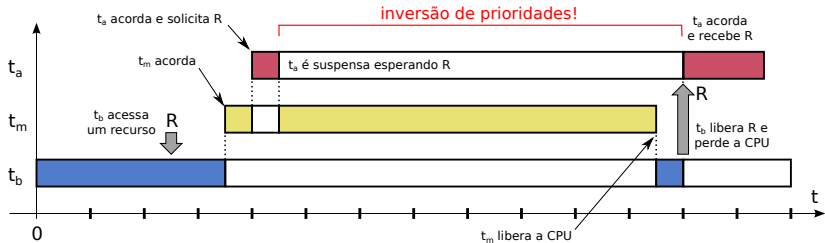
- $t_a$  uma tarefa de prioridade **alta**
- $t_b$  uma tarefa de prioridade **baixa**
- $t_m$  uma tarefa de prioridade **média**
- $R$  um recurso acessado em **exclusão mútua**

Exclusão mútua:

- Somente uma tarefa usa o recurso de cada vez
- As outras tarefas ficam esperando, suspensas

# Inversão de prioridades

Um roteiro de inversão de prioridades:



**Inversão:**  $t_a$  não consegue executar por causa de  $t_b$

# Inversão de prioridades

Passos:

- 1  $t_b$  recebe o processador
- 2  $t_b$  acessa o recurso  $R$  e começa a usá-lo
- 3  $t_b$  perde o processador para alguma  $t_m$
- 4  $t_b$  volta à fila de prontas (mantendo o acesso a  $R$ )
- 5  $t_a$  recebe o processador e solicita acesso a  $R$
- 6  $t_a$  é **suspensa**, pois  $R$  está com  $t_b$
- 7  $t_b$  não libera  $R$ , pois não consegue executar

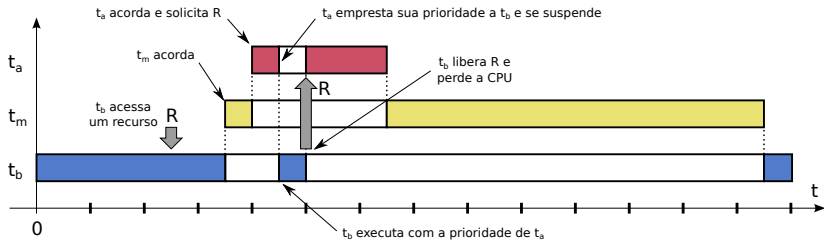
# Inversão de prioridades

Solução: **Protocolos de herança de prioridade**

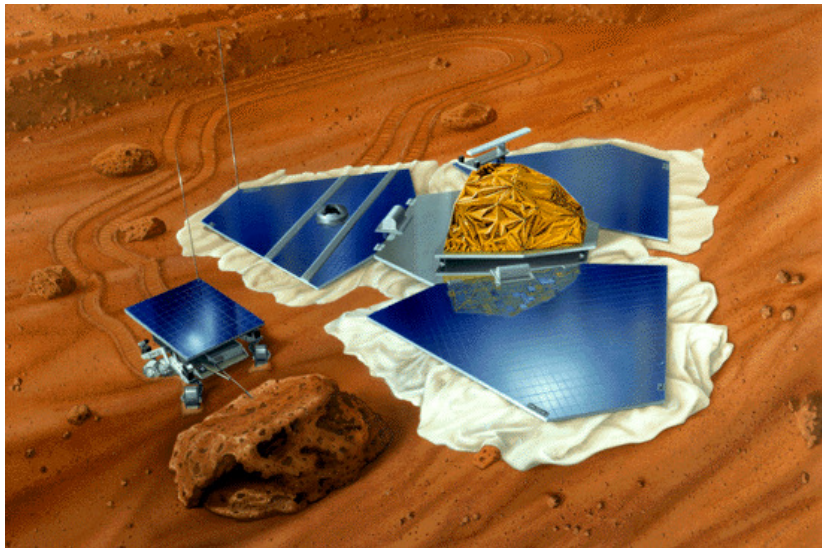
Um exemplo de protocolo:

- 1  $t_a$  pede acesso ao recurso  $R$  (que está com  $t_b$ )
- 2  $t_a$  “empresta” sua prioridade a  $t_b$
- 3  $t_b$  passa à frente das demais tarefas na fila
- 4  $t_b$  consegue executar e liberar  $R$
- 5  $t_b$  retorna à sua prioridade anterior
- 6  $t_a$  volta a executar e obtém acesso a  $R$

# Herança de prioridades



## Exemplo: Sonda Mars PathFinder (1997)



# Sonda Mars Pathfinder

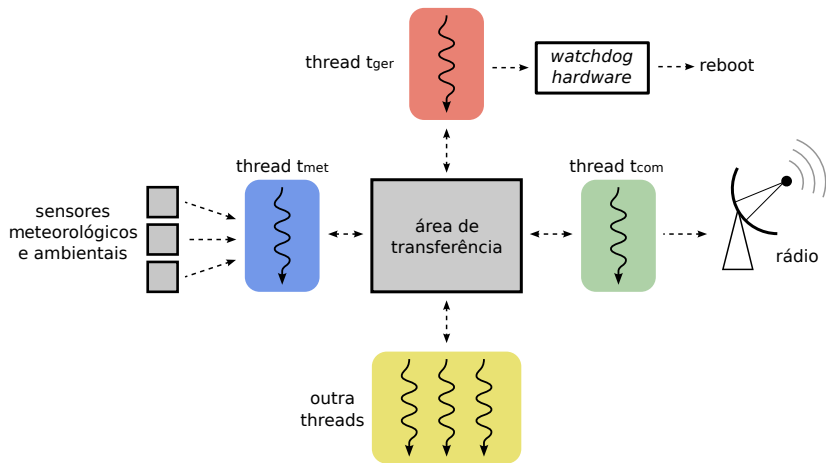
## Sistema principal:

- IBM Risc 6000, SO VX-Works e linguagem C
- +100 *threads* no código
- Escalonamento RR com prioridades fixas
- Área de transferência de dados com exclusão mútua
- Um *watchdog* reinicia o sistema caso ele trave

## Problema:

- *Reboots* frequentes e inesperados
- Indisponibilidade do sistema por várias horas

# Sonda Mars PathFinder



# Sonda Mars PathFinder

*Threads* envolvidas na inversão:

| tarefa | função                            | prioridade | duração |
|--------|-----------------------------------|------------|---------|
| $t_g$  | gerência da área de transferência | alta       | curta   |
| $t_m$  | coleta de dados meteorológicos    | baixa      | curta   |
| $t_c$  | comunicação com a Terra           | média      | longa   |

Solução:

- Ativar herança de prioridade nos semáforos
- Corrigiram um sistema a 400.000.000 Km da Terra