

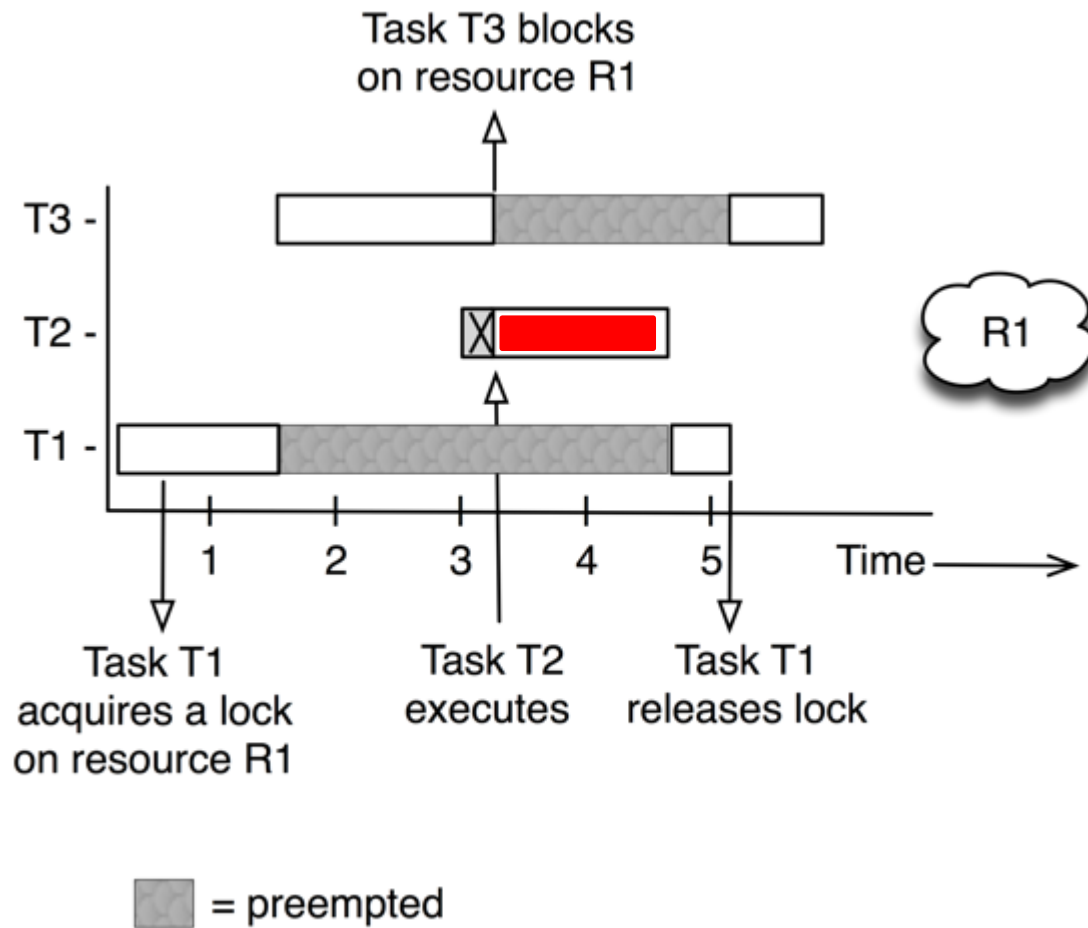
Podrobný popis RTX KEIL č. II.

Inverze priority (*Priority Inversion*)

RTX Real Time Operating system je založen na prioritním preemptivním scheduleru. Ten přiřadí taskům číslo (prioritní úroveň). scheduler vybere z fronty připravených procesů jeden z nejvyšší prioritou a ten vždy běží.

Protože Tasky sdílí zdroje, události mimo dosah RTX, řízení může pozastavit task s nejvyšší prioritou a obsloužit task s nižší a zmeškat kritickou událost. **Priority inversion** je scénář když task s menší prioritou bude spuštěn i když ho **nepotřebujeme**

Inverze priority

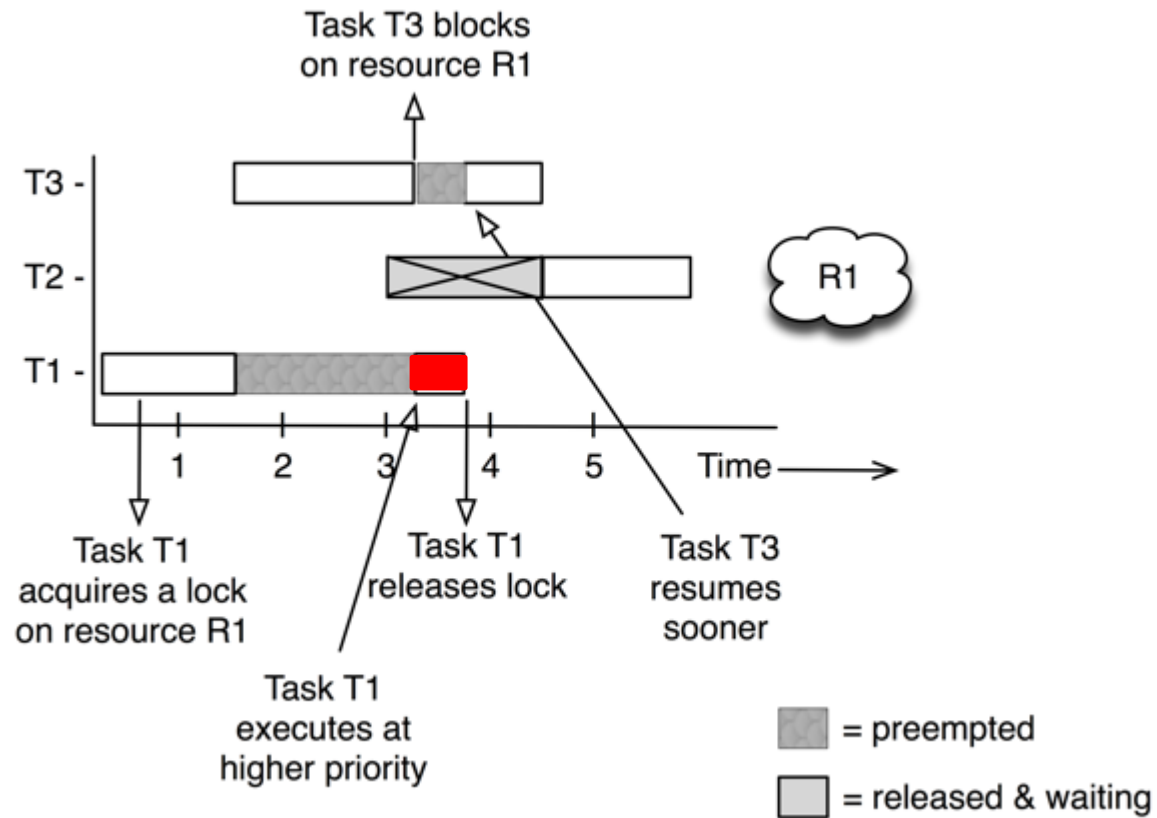


Priority inheritance

Task s nízkou prioritou dědí dočasně vyšší, když blokuje vysokoprioritní task např. čekáním na společné zdroje. Po skončení priorita se vrací do normálu.

RTX [mutex](#) objekty (Mutual Exclusive Lock objects) zaměstnávají Priority inheritance.

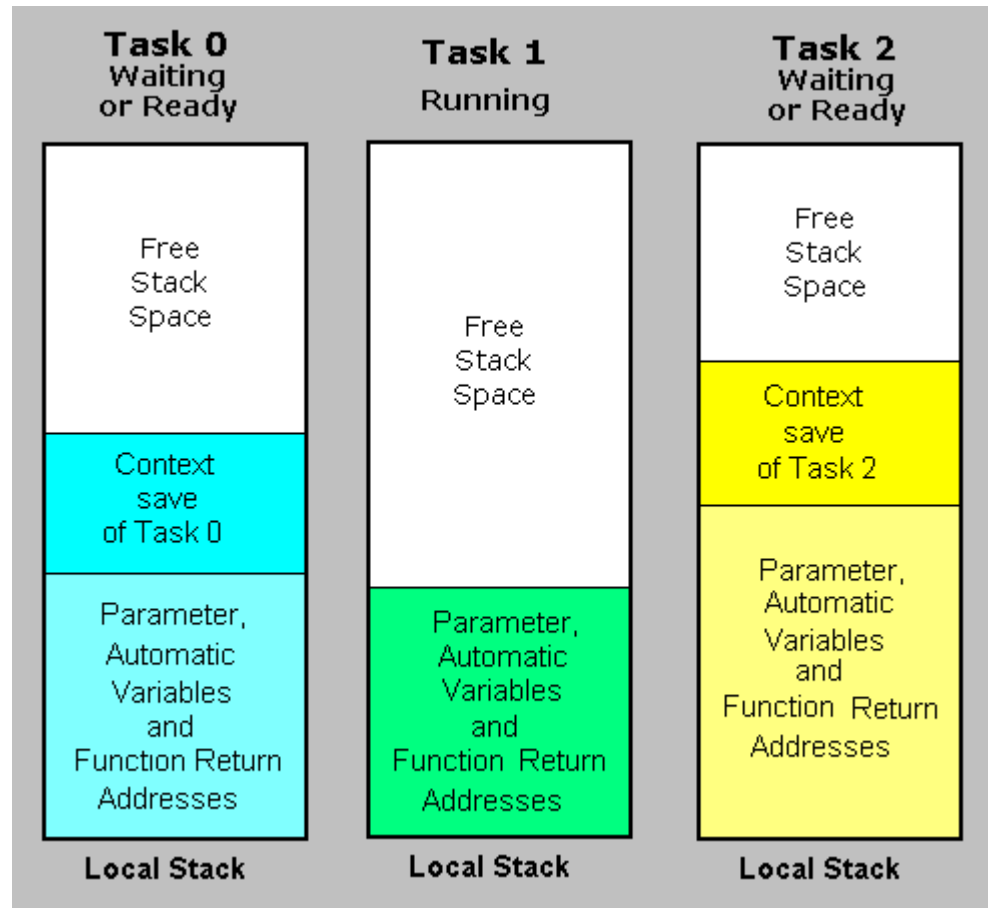
Inverze priority - řešení



Řízení zásobníku (*Stack Management*)

Přepnutí kontextu –
Task 1 ve stavu **RUN**

- Paměť pro zásobníky konfiguruje soubor ARM **startup** .
- Všechny tasky běží v **user mode**.
- Scheduler přepíná mezi user/system mode zásobníky.



Startovací soubor Keil v.4 procesoru ARM STM32F4 pro RTX

```
***** (C) COPYRIGHT 2011 STMicroelectronics *****
,* File Name      : startup_stm32f4xx.s
,* Author       : MCD Application Team
,* Description    : STM32F4xx devices vector table for MDK-ARM toolchain.
,*               This module performs:
,*               - Set the initial SP
,*               - Set the initial PC == Reset_Handler
,*               - Set the vector table entries with the exceptions ISR address
,*               - Configure the system clock and the external SRAM mounted on
,*               STM324xG-EVAL board to be used as data memory (optional,
,*               to be enabled by user)
,*               - Branches to __main in the C library (which eventually
,*               calls main()).
```

Stack_Size **EQU** **0x00000400**

```
        AREA  STACK, NOINIT, READWRITE, ALIGN=3
Stack_Mem  SPACE  Stack_Size
__initial_sp
```

Heap_Size **EQU** **0x00000200**

```
        AREA  HEAP, NOINIT, READWRITE, ALIGN=3
__heap_base
Heap_Mem   SPACE  Heap_Size
__heap_limit
```

```
.
.
.
.
.
```