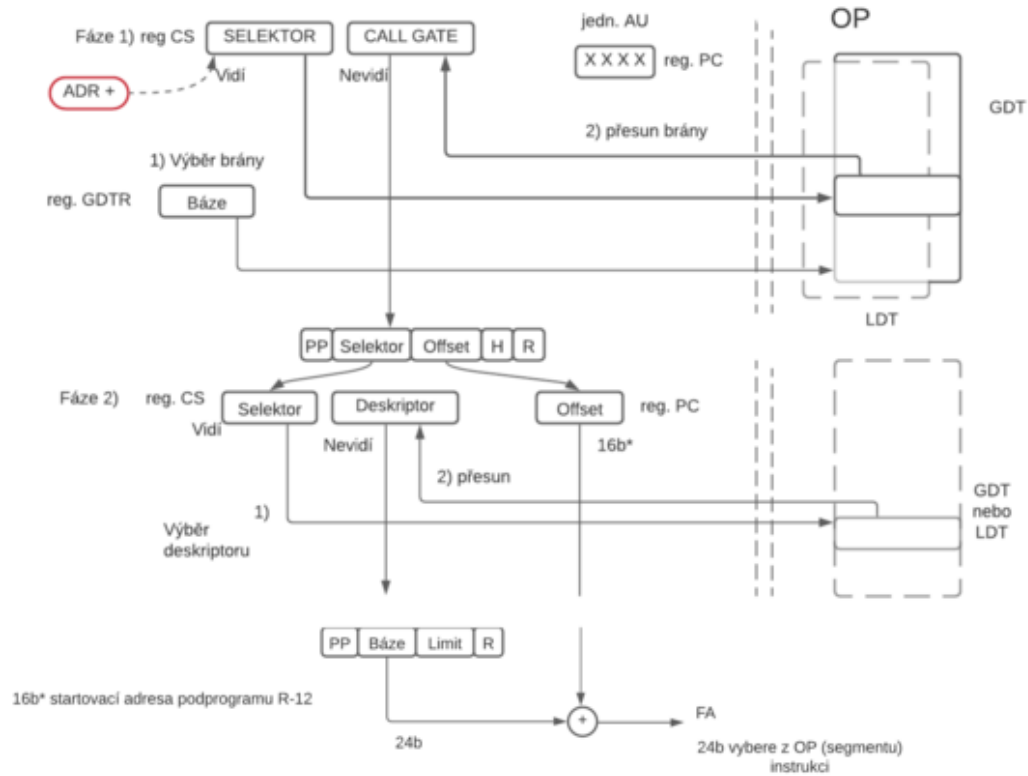




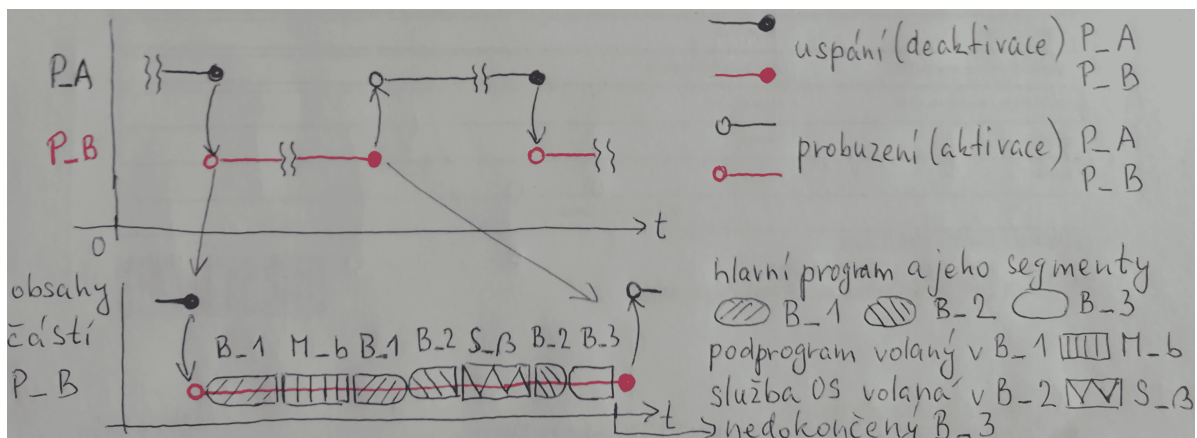
## Výpočet FA



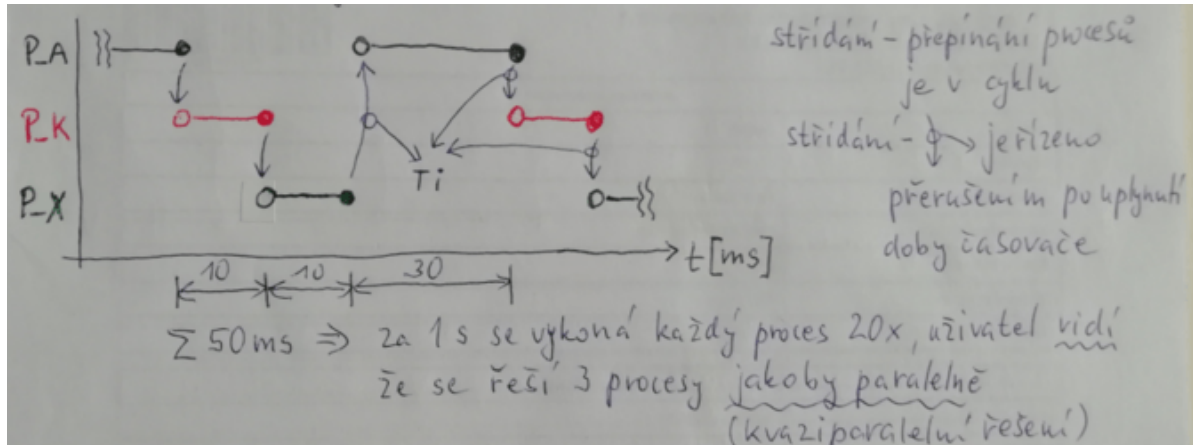
## Princip přepínání úloh a průběh přepínání

probíhá jen při Privilege Level 0. Multitasking se realizuje jako kooperativní a preemptivní. příkl. 1:

Dva procesy P\_A, P\_B ve dvou oknech mezi nimi přepínám klikáním myši



Příkl. 2: V řízení mikroproc. se střídají 3 procesy P\_A, P\_K, P\_X jejich střídání je řízeno stavem OS.



kdy dojde k přepnutí?

Způsobí ho jedna z instrukcí INTČíslo, CALLADR nebo hw signál přerušovací číslo.

Číslo nebo adresa zajistí výběr brány TASKGATE z tabulky LDT

TASKGATE

| PP | SELEKTOR | O | REZ |

## Preemptivní multitasking

V preemptivním multitaskingu o přidělování a odebírání procesoru jednotlivým úlohám plně rozhoduje operační systém. V pravidelných intervalech (typicky zhruba 100x až 1000x za sekundu) za pomoci časovače dojde k vyvolání přerušovací aktuálně běžícího programu (procesu), vyhodnotí se aktuální situace (které úlohy žádají o přidělení procesoru, jejich priority atd.) a nechá běžet buď opět úlohu, kterou přerušil, nebo jinou úlohu, která má zájem o přidělení procesoru (to která úloha bude následně běžet, určuje prioritní fronta). I v preemptivním multitaskingu však může úloha dobrovolně požádat o přepnutí kontextu a vzdát se zbytku svého kvanta (úloha takzvaně „usne“ - proces přejde do stavu sleep nebo se zablokuje provedením pomalé vstupně-výstupní operace, jako je například čtení dat z pevného disku).

## Kooperativní multitasking

Kooperativní multitasking vyžaduje aktivní spoluúčast běžících úloh. Každá úloha je povinná dostatečně často systémovým voláním předat řízení zpět operačnímu systému, který díky tomu může spustit jinou úlohu, která se po chvíli opět dobrovolně vzdá procesoru atd. Výhodou řešení je jednodušší implementace operačního systému. Podstatnou nevýhodou je skutečnost, že chybně naprogramovaná úloha, která nevrátí řízení zpět operačnímu systému, způsobí úplné zastavení systému i ostatních úloh.

## Segment TSS

Ze stávajícího procesu P\_A bude řízení mikroproc. přepnuto na proces P\_B.

- proces P\_A se uspí, deaktivuje, přitom se jeho aktuální stav uloží do systémového segmentu TSSa(registry mikroproc. do TSSa zkopírují)
- proces P\_B se probudí, aktivuje, jeho poslední stav se vybere ze systémového segmentu TSSb (z položek TSSb se obsah nakopíruje do registrů mikroproc.).
- proces P\_B pak může rozeběhnout jako kdyby nikdy nebyl deaktivovaný.

Každý proces při multitaskingu má svůj segment TSS, jeho báze v OP je uložena v reg. TR (v jedn. AU)

velikost TSS - 16b HW systém = 50Byte - 32b HW systém = 110 Byte

- současné HW systémy mikroproc. mají "několik" stovek Byte

Obsah TSS (16bHW): - dvojice CS:PC/IP, DS:SI(DI), ES:BP(BX), SS:SP!

- datové registry AX, BS, CX,DX,freg,MSW 4xzásobníky

PL=0-3

- bazové registry LDTR, GDTR, TR
- ukazatel na minulý proces, atd

Všechny činnosti spojené s uspaním procesu P\_A (uložení do TSSa)

s probuzením procesu P\_B (výběr z TSSb do registrů

mikroproc.)

jsou řízené hw mikroprocesu (jednotka AU a další mají automatickou činnost)

nepoužívá se žádný SW, hw řízení je 40 až 50x rychlejší přepnutí než SW.