

# IOS – projekt 2 (synchronizace)

Zadání je inspirováno knihou Allen B. Downey: The Little Book of Semaphores

## Popis Úlohy (Santa Claus problem)

Santa Claus spí ve své dílně na severním pólu a může být vzbuzen pouze

1. ve chvíli, kdy jsou všichni jeho sobi zpět z letní dovolené, nebo
2. někteří z jeho skřítků mají problém s výrobou hraček a potřebují pomoci.

Aby se Santa Claus prospal, skřítki čekají potichu před dílnou. Ve chvíli, kdy čekají alespoň 3, tak první 3 z fronty najednou vstoupí do dílny. Ostatní skřítki, co potřebují pomoci musí čekat před dílnou až bude volno.

Ve chvíli, kdy přijde poslední sob z dovolené je pomoc čekajícím skřítkům možné odložit. Santa dává na dveře dílny nápis „Vánoce – zavřeno“ a jde zapřahat soby do saní. Všichni skřítki, co čekají před dílnou ihned odcházejí na dovolenou. Ostatní skřítki odchází na dovolenou ve chvíli, kdy potřebují pomoc od Santy a zjistí, že je dílna zavřená.

## Podrobná specifikace úlohy

### Spuštění:

\$ ./proj2 NE NR TE TR

- NE: počet skřítků.  $0 < NE < 1000$
- NR: počet sobů.  $0 < NR < 20$
- TE: Maximální doba v milisekundách, po kterou skřítek pracuje samostatně.  $0 \leq TE \leq 1000$ .
- TR: Maximální doba v milisekundách, po které se sob vrací z dovolené domů.  $0 \leq RE \leq 1000$ .
- Všechny parametry jsou nezáporná celá čísla.

### Chybové stavy:

- Pokud některý ze vstupů nebude odpovídat očekávanému formátu nebo bude mimo povolený rozsah, program vytiskne chybové hlášení na standardní chybový výstup, uvolní všechny dosud alokované zdroje a ukončí se s kódem (exit code) 1.
- Pokud selže některá z operací se semaforem, nebo sdílenou pamětí, postupujte stejně-- program vytiskne chybové hlášení na standardní chybový výstup, uvolní všechny dosud alokované zdroje a ukončí se s kódem (exit code) 1.

## Implementační detaily:

- Každý proces vykonává své akce a současně zapisuje informace o akcích do souboru s názvem proj2.out. Součástí výstupních informací o akci je pořadové číslo A prováděné akce (viz popis výstupů). Akce se číslují od jedničky.
- Použijte sdílenou paměť pro implementaci čítače akcí a sdílených proměnných nutných pro synchronizaci.
- Použijte semaforey pro synchronizaci procesů.
- Nepoužívejte aktivní čekání (včetně cyklického časového uspání procesu) pro účely synchronizace.
- Pracujte s procesy, ne s vlákny.

## Hlavní proces

1. Hlavní proces vytváří ihned po spuštění jeden proces Santa, NE procesů skřítků a NR procesů sobů.
2. Poté čeká na ukončení všech procesů, které aplikace vytváří. Jakmile jsou tyto procesy ukončeny, ukončí se i hlavní proces s kódem (exit code) 0.

## Proces Santa

1. Po spuštění vypíše: *A: Santa: going to sleep*
2. Po probuzení skřítky jde pomáhat elfům---vypíše: *A: Santa: helping elves*
3. Poté, co pomůže skřítkům jde spát (bez ohledu na to, jestli před dílnou čekají další skřítkci) a vypíše: *A: Santa: going to sleep*
4. Po probuzení posledním sobem uzavře dílnu a vypíše: *A: Santa: closing workshop* a pak jde ihned zapřahat soby do saní.
5. Ve chvíli, kdy jsou zapřažení všichni soby vypíše: *A: Santa: Christmas started* a ihned proces končí.

## Proces Skřítek

1. Každý skřítek je unikátně identifikován číslem elfID.  $0 < \text{elfID} \leq \text{NE}$
2. Po spuštění vypíše: *A: Elf elfID: started*
3. Samostatnou práci modelujte voláním funkce usleep na náhodný čas v intervalu  $<0, \text{TE}>$ .
4. Když skončí samostatnou práci, potřebuje pomoc od Santy. Vypíše: *A: Elf elfID: need help* a zařadí se do fronty před Santovou dílnou.
5. Pokud je třetí ve frontě před dílnou, dílna je prázdná a na dílně není cedule „Vánoce – zavřeno“, tak společně s prvním a druhým z fronty vstoupí do dílny a vzbudí Santu.
6. Skřítek v dílně dostane pomoc a vypíše: *A: Elf elfID: get help* (na pořadí pomoci skřítkům v dílně nezáleží)
7. Po obdržení pomoci ihned odchází z dílny a pokud je dílna již volná, tak při odchodu z dílny může upozornit čekající skřítky, že už je volno (volitelné).
8. Pokud je na dveřích dílny nápis „Vánoce – zavřeno“ vypíše: *A: Elf elfID: taking holidays* a proces ihned skončí.

## Proces Sob

1. Každý sob je identifikován číslem rdID,  $0 < \text{rdID} \leq \text{NR}$

2. Po spuštění vypíše: *A: RD rdID: rstarted*
3. Čas na dovolené modelujte voláním *usleep* na náhodný interval  $\langle TR/2, TR \rangle$
4. Po návratu z letní dovolené vypíše: *A: RD rdID: return home*  
a následně čeká, než ho Santa zapřáhne k saním. Pokud je posledním sobem, který se vrátil z dovolené, tak vzbudí Santu.
5. Po zapřážení do saní vypíše: *A: RD rdID: get hitched*  
a následně proces končí.

## Podmínky vypracování

### Obecné informace

- Projekt implementujte v jazyce C. Komentujte zdrojové kódy, programujte přehledně. Součástí hodnocení bude i kvalita zdrojového kódu.
- Kontrolujte, zda se všechny procesy ukončují korektně a zda při ukončování správně uvolňujete všechny alokované zdroje.
- Dodržujte syntax zadaných jmen, formát souborů a formát výstupních dat. Použijte základní skript pro ověření korektnosti výstupního formátu (dostupný z webu se zadáním).
- Dotazy k zadání: Veškeré nejasnosti a dotazy řešte pouze prostřednictvím diskuzního fóra k projektu 2.
- Poznámka k testování: Můžete si nasimulovat častější přepínání procesů například vložení krátkého usnutí po uvolnění semaforů apod. Pouze pro testovací účely, do finálního řešení nevkládejte!

### Překlad

- Pro překlad používejte nástroj *make*. Součástí odevzdání bude soubor *Makefile*.
- Překlad se provede příkazem *make* v adresáři, kde je umístěn soubor *Makefile*.
- Po překladu vznikne spustitelný soubor se jménem *proj2*, který bude umístěn ve stejném adresáři jako soubor *Makefile*
- Zdrojové kódy překládejte s přepínači *-std=gnu99 -Wall -Wextra -Werror -pedantic*
- Pokud to vaše řešení vyžaduje, lze přidat další přepínače pro linker (např. kvůli semaforům či sdílené paměti, *-pthread, -lrt, . . .*).
- Vaše řešení musí být možné přeložit a spustit na serveru *merlin*.

### Odevzdání

- Součástí odevzdání budou pouze soubory se zdrojovými kódy (*\*, \*.h*) a soubor *Makefile*. Tyto soubory zabalte pomocí nástroje *zip* do archivu s názvem *proj2.zip*.
- Archiv vytvořte tak, aby po rozbalení byl soubor *Makefile* umístěn ve stejném adresáři, jako je archiv.
- Archiv *proj2.zip* odevzdejte prostřednictvím informačního systému—termín Projekt 2.
- Pokud nebude dodržena forma odevzdání nebo projekt nepůjde přeložit, bude projekt hodnocen 0 body.
- Archiv odevzdejte pomocí informačního systému v dostatečném předstihu (odevzdaný soubor můžete před vypršením termínu snadno nahradit jeho novější verzí, kdykoliv budete potřebovat).

## Příklad výstupu

Příklad výstupního souboru proj2.out pro následující příkaz:

```
$ ./proj2 5 4 100 100
-----
1: Elf 1: started
2: Elf 2: started
3: Elf 1: need help
4: Santa: going to sleep
5: RD 1: rstarted
6: Elf 2: need help
7: Elf 3: started
8: RD 4: rstarted
9: Elf 4: started
10: Elf 5: started
11: RD 3: rstarted
12: RD 3: return home
13: Elf 5: need help
14: RD 2: rstarted
15: Santa: helping elves
16: Elf 2: get help
17: Elf 5: get help
18: RD 1: return home
19: Elf 1: get help
20: Santa: going to sleep
21: Elf 4: need help
22: Elf 5: need help
23: Elf 1: need help
24: Elf 2: need help
25: Santa: helping elves
26: Elf 1: get help
27: Elf 4: get help
28: Elf 5: get help
29: Santa: going to sleep
30: RD 2: return home
31: RD 4: return home
32: Santa: closing workshop
33: RD 1: get hitched
34: Elf 2: taking holidays
35: RD 3: get hitched
36: Elf 1: need help
37: Elf 1: taking holidays
38: RD 2: get hitched
39: RD 4: get hitched
40: Santa: Christmas started
41: Elf 4: need help
42: Elf 5: need help
43: Elf 5: taking holidays
44: Elf 3: need help
45: Elf 4: taking holidays
46: Elf 3: taking holidays
```