

vyuka:ios\_ulohy

## 2. Úloha (2013)

### Popis úlohy

Implementujte v jazyku C modifikovaný synchronizační problém *Santa Claus Problem* (můžete porovnat např. s jednou z verzí tohoto problému [<http://www.greenteapress.com/semaphores/1>]). Santa a skřítki jsou na severním pólu, Santa odpočívá a skřítki vyrábějí hračky. Pokud skřítek narazí na problém, jde za Santou s žádostí o pomoc. Poté, co se skřítkovi dostane pomoci od Santy, vrací se skřítek zpět do práce. Jakmile počet návštěv skřítku u Santy dosáhne předem stanoveného počtu, skřítek odjede na dovolenou. Jakmile Santa pomohl poslednímu skřítkovi (všichni jsou na dovolené), ukončí se procesy skřítků a proces Santy. Skřítki vyrábějící hračky jsou *aktivní skřítki*, skřítki na dovolené jsou *odpočívající skřítki*.

Obsluha skřítků (tj. návštěva skřítků u Santy a poskytnutí pomoci) může proběhnout dvěma různými způsoby, podle počtu *aktivních skřítků*:

1. Počet všech *aktivních skřítků* je větší než 3: Santa pomůže pouze skupině tří skřítků. První skřítek, který potřebuje Santovu pomoc, čeká u Santy na další dva skřítky. Jakmile požádá o pomoc třetí skřítek, Santa jim pomůže. Ostatní skřítki, kteří potřebují pomoc, musí se svoji žádostí počkat, až odejde skupina skřítků od Santy.
2. Počet všech *aktivních skřítků* je menší nebo roven 3: jakmile má skřítek problém, požádá Santu o pomoc a Santa mu hned pomůže. Ostatní skřítki, kteří potřebují pomoc, musí počkat, až odejde skřítek od Santy.

V průběhu řešení problému se tedy může změnit způsob obsluhy skřítků. Například se začíná s 10 *aktivními skřítky* (obsluha způsobem 1), postupně jejich počet klesá (odcházejí na dovolenou), a když zůstanou 3 *aktivní skřítki* (tj. 7 skřítků odešlo na dovolenou a jsou z nich *odpočívající skřítki*), přejde se na obsluhu způsobem 2.

### Detailní specifikace úlohy

#### Spuštění

\$ ./santa C E H S

kde

- C je počet návštěv u Santy, které musí skřítek podstoupit, než odjede na dovolenou. Pro všechny skřítky je počet návštěv stejný.  $C > 0$ .
- E je počet skřítků, kteří na začátku pracují na výrobě hraček.  $E > 0$ .
- H je maximální hodnota doby (v milisekundách) výroby hraček skřítkem, než se vyskytne problém; poté potřebuje skřítek pomoc.  $H \geq 0$ .
- S je maximální hodnota doby (v milisekundách) obsluhy skřítků Santou (délka trvání pomoci).  $S \geq 0$ .
- všechny parametry jsou celá čísla

## Implementační detaily

- Santovi a každému skřítkovi odpovídá jeden proces. Po spuštění vytváří hlavní proces E procesů pro skřítky a 1 proces pro Santu (celkem tedy bude existovat E+2 procesů včetně hlavního).
- Hlavní proces čeká na ukončení všech vytvořených procesů, a poté se sám ukončí.
- Každý proces skřítky bude interně identifikován celým číslem I, začínajícím od 1.
- Každý proces vykonává své akce a současně zapisuje informace o akcích do souboru s názvem `santa.out`.
  - Přístup k výstupnímu zařízení (zápis informací) musí být výlučný; pokud zapisuje jeden proces a další chce také zapisovat, musí počkat na uvolnění zdroje.
  - 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.
- Hlavní proces čeká na ukončení všech skřítků a Santy. Poté se ukončí s kódem (*exit code*) 0.
- Budete-li potřebovat generovat unikátní klíč, je vhodné použít funkci `ftok` [<http://pubs.opengroup.org/onlinepubs/00969699/functions/ftok.html>].
- Další funkce a systémová volání: `fork` [<http://www.kernel.org/doc/man-pages/online/pages/man2/fork.2.html>], `wait` [<http://www.kernel.org/doc/man-pages/online/pages/man2/wait.2.html>], `shmat` [<http://www.kernel.org/doc/man-pages/online/pages/man2/shmat.2.html>], `semctl` [<http://www.kernel.org/doc/man-pages/online/pages/man2/semctl.2.html>], `semget` [<http://www.kernel.org/doc/man-pages/online/pages/man2/semget.2.html>], `shmget` [<https://www.kernel.org/doc/man-pages/online/pages/man2/shmget.2.html>], `sem_open` [[http://man7.org/linux/man-pages/man3/sem\\_open.3.html](http://man7.org/linux/man-pages/man3/sem_open.3.html)], `usleep` [<http://man7.org/linux/man-pages/man3/usleep.3.html>], ...

## Chybové stavy

- Pokud některý ze vstupů nebude odpovídat očekávanému formátu, program vytiskne chybové hlášení na standardní chybový výstup, uvolní všechny alokované zdroje a ukončí se s kódem (*exit code*) 1.
- Pokud selže systémové volání, program vytiskne chybové hlášení na standardní chybový výstup, uvolní všechny alokované zdroje a ukončí se s kódem (*exit code*) 2.

## Popis procesů a jejich výstupů

Poznámka k výstupům: A je pořadové číslo prováděné akce a I je interní identifikátor skřítky.

### Santa

1. Po spuštění tiskne A: `santa: started`.
2. Čeká na požadavek.
3. Ihned po příchodu požadavku ověří stav a tiskne A: `santa: checked state: NE: WE`, kde NE je počet všech *aktivních skřítků* a WE je počet skřítků čekajících na obsluhu.
4. Pokud Santa nemůže provést obsluhu, čeká na další požadavek (zpět ke kroku 2).

5. Pokud Santa může provést obsluhu, tiskne A: `santa: can help` a poté provede obsluhu; vlastní obsluha (tj. pomoc skřítkům) je simulována uspáním procesu na náhodnou dobu v rozmezí  $<0, S>$ .
6. Pokud jsou všichni skřítki na dovolené, Santa se ukončí; před ukončením ověří stav včetně tisku A: `santa: checked state: NE: WE`, a poté tiskne A: `santa: finished`.
7. Jinak se vrací k bodu 2.

## Skřítek

1. Po spuštění tiskne A: `elf: I: started`.
2. Pracuje; práce je simulována uspáním procesu na náhodnou dobu v rozmezí  $<0, H>$ .
3. Ihned po probuzení (tj. vyskytnutí problému) tiskne A: `elf: I: needed help`.
4. Poté žádá o pomoc Santu; těsně před tím, než Santu požádá o pomoc, tiskne informaci A: `elf: I: asked for help`.
5. Skřítek čeká na dokončení obsluhy; není třeba čekat na povolení vstupu k Santovi (např. až požádají 3 skřítki), pouze musí počkat, než Santa obsluhu dokončí.
6. Po ukončení obsluhy tiskne skřítek A: `elf: I: got help`.
7. Pokud Santa začne obsluhovat skřítky, žádný další skřítek nemůže požádat o pomoc (a tedy ani tisknout informaci `asked for help`), dokud Santa neskončí s obsluhou.
8. Pokud skřítek prošel poslední obsluhou (viz parametr C), odjíždí na dovolenou, a před tím tiskne A: `elf: I: got a vacation`; jinak se vrací k bodu 2.
9. Jakmile jsou na dovolené všichni skřítki, proces skřítků se ukončí; před ukončením tiskne A: `elf: I: finished`.

## Ukázka výstupů

### Ukázka č. 1

Spuštění: `$ ./santa 2 2 10 10`

santa.out:

```
1: elf: 1: started
2: elf: 2: started
3: santa: started
4: elf: 1: needed help
5: elf: 1: asked for help
6: elf: 2: needed help
7: santa: checked state: 2: 1
8: santa: can help
9: elf: 1: got help
10: elf: 2: asked for help
11: santa: checked state: 2: 1
12: santa: can help
13: elf: 1: needed help
14: elf: 2: got help
15: elf: 1: asked for help
16: santa: checked state: 2: 1
17: santa: can help
18: elf: 2: needed help
```

```
19: elf: 1: got help
20: elf: 1: got a vacation
21: elf: 2: asked for help
22: santa: checked state: 1: 1
23: santa: can help
24: elf: 2: got help
25: elf: 2: got a vacation
26: santa: checked state: 0: 0
27: santa: finished
28: elf: 1: finished
29: elf: 2: finished
```

## Ukázka č. 2

Spuštění: \$ ./santa 2 4 10 10

santa.out:

```
1: elf: 1: started
2: elf: 2: started
3: elf: 3: started
4: santa: started
5: elf: 4: started
6: elf: 1: needed help
7: elf: 1: asked for help
8: elf: 2: needed help
9: elf: 2: asked for help
10: elf: 3: needed help
11: elf: 3: asked for help
12: santa: checked state: 4: 3
13: santa: can help
14: elf: 4: needed help
15: elf: 1: got help
16: elf: 3: got help
17: elf: 2: got help
18: elf: 4: asked for help
19: elf: 1: needed help
20: elf: 1: asked for help
21: elf: 2: needed help
22: elf: 2: asked for help
23: elf: 3: needed help
24: santa: checked state: 4: 3
25: santa: can help
26: elf: 1: got help
27: elf: 1: got a vacation
28: elf: 4: got help
29: elf: 2: got help
30: elf: 2: got a vacation
31: elf: 3: asked for help
32: santa: checked state: 2: 1
33: santa: can help
34: elf: 4: needed help
35: elf: 3: got help
36: elf: 3: got a vacation
37: elf: 4: asked for help
38: santa: checked state: 1: 1
39: santa: can help
```

```
40: elf: 4: got help
41: elf: 4: got a vacation
42: santa: checked state: 0: 0
43: santa: finished
44: elf: 4: finished
45: elf: 1: finished
46: elf: 2: finished
47: elf: 3: finished
```

## Podmínky vypracování projektu

### 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 (např. příkazem `ipcs` můžete zjistit, jaké zdroje System V jsou v systému alokovány).
- Dodržujte syntax zadaných jmen, formát souborů a formát výstupních dat! Čtěte pozorně zadání a poznámky k vypracování u jednotlivých zadání.
- Dotazy k zadání: Veškeré nejasnosti a dotazy řešte pouze prostřednictvím diskuzního fóra k projektu 2.

### 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 `santaClaus`, který bude umístěn ve stejném adresáři jako soubor `Makefile`.
- Zdrojové kódy překládejte s přepínači `-std=gnu99 -Wall -Wextra -Werror -pedantic`.

### Odevzdání

- Součástí odevzdání budou pouze soubory se zdrojovými kódy (`*.c`, `*.h`) a soubor `Makefile`. Tyto soubory zabalte pomocí nástroje `zip` do archivu s názvem `xlogin.zip`, kde `xlogin` je váš login.
- Archiv vytvořte tak, aby po rozbalení byl soubor `Makefile` umístěn ve stejném adresáři, jako je archiv.
- Archiv `xlogin.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).

## Ověřovací skripty

Základní skripty pro ověření korektnosti výstupního formátu a základní posloupnosti budou dodány později.

Last modified: 2013-03-28 17:54