IOS – projekt 2 (synchronizace)

Zadání je inspirováno knihou Allen B. Downey: The Little Book of Semaphores (The Senate Bus problem)

Popis Úlohy (Skibus)

V systému máme 3 typy procesů: (0) hlavní proces, (1) skibus a (2) lyžař. Každý lyžař jde po snídani na jednu nástupní zastávku skibusu, kde čeká na příjezd autobusu. Po příjezdu autobusu na nástupní zastávku lyžaři nastoupí. Pokud je naplněna kapacita autobusu, čekají zbývající lyžaři na další spoj. Autobus postupně obslouží všechny nástupní zastávky a doveze lyžaře na výstupní zastávku u lanovky. Pokud jsou ještě další zájemci o svezení, pokračuje dalším kolem.

Podrobná specifikace úlohy

Spuštění:

$ ./proj2 L Z K TL TB

- L: počet lyžařů, L<20000
- Z: počet nástupních zastávek, 0<Z<=10
- K: kapacita skibusu, 10<=K<=100
- TL: Maximální čas v mikrosekundách, který lyžař čeká, než přijde na zastávku, 0<=TL<=10000
- TB: Maximální doba jízdy autobusu mezi dvěma zastávkami. 0<=TB<=1000

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 semafory, sdílenou pamětí, nebo volání fork, 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.
- Celkový počet spuštěných procesů bude 2+L (hlavní proces, skibus a L lyžařů). Nespouštějte žádné další pomocné procesy.
- Použijte sdílenou paměť pro implementaci čítače akcí a sdílených proměnných nutných pro synchronizaci.
- Použijte semafory pro synchronizaci procesů.
- Nepoužívejte aktivní čekání (včetně cyklického časového uspání procesu) pro účely synchronizace. Pro čekání lyžařů na skibus a čekání skibusu na nástup/výstup lyžařů použijte semafory.
- Pracujte s procesy, ne s vlákny.

- Lyžaři nemusí (ale mohou) nastupovat do skibusu v pořadí, ve kterém přišli na zastávku. Stejně tak výstup může být libovolném pořadí.
- Lyžař může nastoupit/vystoupit pouze během stání v zastávce---výpis *L: boarding* (*L: going to ski*) bude ve výstupním souboru pouze mezi příslušnými výpisy *BUS: arrived to* a BUS: *leaving*.
- Každá zastávka je jednoznačně identifikována číslem idZ z intervalu <1,Z>.

Hlavní proces

- Proces vytváří ihned po spuštění proces skibus.
- Dále vytvoří L procesů lyžařů.
- Každému lyžaři při jeho spuštění náhodně přidělí nástupní zastávku.
- 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 Skibus

- Po spuštění vypíše: *A: BUS: started*
- *(#) idZ=1 (identifikace zastávky)*
- (*) Jede na zastávku idZ---čeká pomocí volání usleep náhodný čas v intervalu <0,TB>.
- Vypíše: *A: BUS: arrived to idZ*
- Nechá nastoupit všechny čekající lyžaře do kapacity autobusu
- Vypíše: *A: BUS: leaving idZ*
- Pokud idZ<Z, tak idZ=idZ+1 a pokračuje bodem (*)
- Jinak jede na výstupní zastávku---čeká pomocí volání usleep náhodný čas v intervalu <0,TB>.
- Vypíše: *A: BUS: arrived to final*
- Nechá vystoupit všechny lyžaře
- Vypíše: *A: BUS: leaving final*
- Pokud ještě nějací lyžaři čekají na některé ze zastávek/mohou přijít na zastávku, tak pokračuje bodem (#)
- Jinak vypíše: *A: BUS: finish*
- Proces končí

pozn.: Doba jízdy mezi zastávkami může být v každém kole jiná.

Proces Lyžař

- Každý lyžař je jednoznačně identifikován číslem idL, 0<idL<=L
- Po spuštění vypíše: *A: L idL: started*
- Dojí snídani---čeká v intervalu <0,TL> mikrosekund.
- Jde na přidělenou zastávku idZ.
- Vypíše: *A: L idL: arrived to idZ*
- Čeká na příjezd skibusu
- Po příjezdu skibusu nastoupí (pokud je volná kapacita)
- Vypíše: *A: L idL: boarding*
- Vyčká na příjezd skibusu k lanovce.
- Vypíše: *A: L idL: going to ski*
- Proces končí

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 vložením krátkého uspání 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
- Spustitelný soubor může být závislý pouze na systémových knihovnách---nesmí předpokládat existenci žádného dalšího studentem vytvořeného souboru (např. spustitelný soubor úředník, konfigurační soubor, dynamická knihovna zákazník, ...).
- 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 (*.c , *.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 8 4 10 4 5
```

```
1: L 1: started
2: L 3: started
3: L 1: arrived to 2
4: L 4: started
```

```
5: L 3: arrived to 4
6: L 2: started
7: L 5: started
8: L 2: arrived to 2
9: L 6: started
10: L 4: arrived to 2
11: L 8: started
12: L 5: arrived to 2
13: L 6: arrived to 2
14: L 7: started
15: BUS: started
16: L 8: arrived to 2
17: L 7: arrived to 2
18: BUS: arived to 1
19: BUS: leaving 1
20: BUS: arived to 2
21: L 1: boarding
22: L 2: boarding
23: L 4: boarding
24: L 5: boarding
25: L 6: boarding
26: L 8: boarding
27: L 7: boarding
28: BUS: leaving 2
29: BUS: arived to 3
30: BUS: leaving 3
31: BUS: arived to 4
32: L 3: boarding
33: BUS: leaving 4
34: BUS: arived to final
35: L 2: going to ski
36: L 7: going to ski
37: L 6: going to ski
38: L 3: going to ski
39: L 1: going to ski
40: L 5: going to ski
41: L 4: going to ski
42: L 8: going to ski
43: BUS: leaving final
44: BUS: finish
```