

2. Úloha IOS (2019/20)

A. Popis úlohy

Implementujte v jazyce C modifikovaný synchronizační problém *Faneuil Hall Problem* (můžete se inspirovat knihou *The Little Book of Semaphores*).

Existují dva druhy vláken: přistěhovalci (*immigrants*) a jeden soudce (*judge*). Přistěhovalci musí čekat ve frontě, vstoupit do soudní budovy, zaregistrovat se a pak čekat na rozhodnutí soudce. V určitém okamžiku vstoupí soudce do budovy. Když je soudce v budově, nikdo jiný nesmí vstoupit dovnitř ani odejít z budovy. Jakmile se všichni přistěhovalci, kteří vstoupili do budovy, zaregistrují, může soudce vydat rozhodnutí (potvrdit naturalizaci). Po vydání rozhodnutí (potvrzení) si přistěhovalci vyzvednou certifikát o občanství USA. Soudce odejde z budovy v určitém okamžiku po rozhodnutí. Poté, co přistěhovalci vyzvednou certifikát, mohou odejít.

Přistěhovalci vstupují do budovy jednotlivě (pouze jeden turniket) a také se jednotlivě registrují (pouze jedno registrační místo). Soudce vydává rozhodnutí pro všechny registrované přistěhovalce naráz. Certifikáty si imigranti vyzvedávají nezávisle na sobě a přítomnosti soudce v budově.

B. Podrobná specifikace úlohy

Spuštění

```
$ ./proj2 PI IG JG IT JT
```

kde

- PI je počet procesů vygenerovaných v kategorii přistěhovalců; bude postupně vytvořeno PI immigrants.
P >= 1
- IG je maximální hodnota doby (v milisekundách), po které je generován nový proces *immigrant*.
IG >= 0 && IG <= 2000.
- JG je maximální hodnota doby (v milisekundách), po které soudce opět vstoupí do budovy.
JG >= 0 && JG <= 2000.
- IT je maximální hodnota doby (v milisekundách), která simuluje trvání vyzvedávání certifikátu přistěhovalcem.
IT >= 0 && IT <= 2000.
- JT je maximální hodnota doby (v milisekundách), která simuluje trvání vydávání rozhodnutí soudcem.
JT >= 0 && JT <= 2000.
- Všechny parametry jsou celá čísla.

Implementační detaily

- *Pracujte s procesy, ne s vlákny.*
- Hlavní proces vytváří ihned po spuštění jeden pomocný proces pro generování procesů přistěhovalců a jeden proces pro soudce. Poté čeká na ukončení všech procesů, které aplikace vytváří. Jakmile jsou tyto procesy ukončeny, ukončí se i hlavní proces s kódem (exit code) 0.
- Generování procesů
 - *immigrant*: pomocný proces generuje procesy pro *immigrants*; každý nový proces je generován po uplynutí náhodné doby z intervalu $<0, IG>$; celkem vygeneruje PI procesů. Pokud platí $IG==0$, všechny příslušné procesy se vygenerují ihned.
 - Každý takto vygenerovaný proces bude interně identifikován celým číslem I, začínajícím od 1. Číselná řada je pro každou kategorii procesů zvlášť.
 - Postupně tedy vznikne hlavní proces, proces soudce, pomocný proces a PI procesů přistěhovalců.
- 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.*

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.

Popis procesů a jejich výstupů

Poznámka k výstupům

- A je pořadové číslo prováděné akce,
- NAME je zkratka kategorie příslušného procesu, tj. JUDGE nebo IMM,
- I je interní identifikátor procesu v rámci příslušné kategorie,
- NE je aktuální počet přistěhovalců, kteří vstoupili do budovy a dosud o nich nebylo rozhodnuto
- NC je aktuální počet přistěhovalců, kteří se zaregistrovali a dosud o nich nebylo rozhodnuto
- NB je aktuální počet přistěhovalců, kteří jsou v budově
- Při vyhodnocování výstupu budou ignorovány mezery a tabelátory.

Proces osoby (immigrant)

1. Po spuštění tiskne A: NAME I: starts
2. Pokouší se dostat do budovy
 - (a) Pokud je soudce v budově, čeká, až odejde
 - (b) Pokud není soudce v budově, může vstoupit. Imigranti vstupují jeden po druhém.
Ihned po vstupu tiskne A: NAME I: enters: NE: NC: NB.
Poznámka: Hodnoty NE a NB zahrnují i tento proces, který právě vstoupil.
3. Registrace
 - (a) Každý proces *immigrant* se registruje samostatně (jeden po druhém). Pořadí registrace nemusí odpovídat pořadí vstupu.
 - (b) Ihned po registraci tiskne A: NAME I: checks: NE: NC: NB.
Poznámka: Hodnota NC zahrnuje i tento proces, který se právě registroval.
4. Proces čeká na vydání rozhodnutí soudcem.
5. Po vydání rozhodnutí si vyzvedává certifikát
 - (a) Tiskne A: NAME I: wants certificate: NE: NC: NB a uspí se na náhodnou dobu z intervalu $<0, IT>$.
 - (b) Po vzbuzení tiskne A: NAME I: got certificate: NE: NC: NB.
 - (c) *Poznámka: Hodnoty NE a NC jsou sníženy o počet přistěhovalců, jejichž případ byl právě rozhodnut soudcem.*
6. Odchod budovy
 - (a) Pokud je soudce v budově, čeká na jeho odchod.
 - (b) Pokud není soudce v budově, proces tiskne A: NAME I: leaves: NE: NC: NB a ukončí se. *Poznámka: Hodnota NB reflektuje odchod tohoto procesu z budovy.*

Proces osoby (judge)

1. Po spuštění netiskne proces nic.
2. Jakmile je připraven vstoupit do budovy, tj. uplyne náhodná doba z intervalu $\langle 0, JG \rangle$, tiskne A: NAME: enters: NE: NC: NB.
3. Vydání rozhodnutí
 - (a) Pokud nejsou všichni přistěhovalci, kteří vstoupili do budovy a dosud o nich nebylo rozhodnuto, registrovaní (NE != NC)
 - i. tiskne A: NAME: waits for imm: NE: NC: NB
 - ii. a čeká, až se všichni registrují
 - (b) Pokud jsou všichni registrovaní (NE == NC)
 - i. tiskne A: NAME: starts confirmation: NE: NC: NB
 - ii. uspí se na náhodnou dobu z intervalu $\langle 0, JT \rangle$
 - iii. poté tiskne A: NAME: ends confirmation: NE: NC: NB. *Poznámka k tomuto výstupu: Hodnoty NE a NC jsou sníženy o počet přistěhovalců, jejichž případ byl právě rozhodnut (soudce rozhoduje všechny naráz, takže $NE=NC=0$).*
4. Odchod z budovy
 - (a) Uspí se na náhodnou dobu z intervalu $\langle 0, JT \rangle$.
 - (b) Poté odchází z budovy a tiskne A: NAME: leaves: NE: NC: NB.
5. Pokud nebyly soudcem zpracovány všechny procesy přistěhovalců (PI), pokračuje bodem 2. V opačném případě se ukončí.
6. Pokud byly soudcem zpracovány všechny procesy přistěhovalců (PI), proces soudce se ukončí a tiskne A: NAME: finishes.

C. 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). Informace o skriptu jsou uvedeny v komentáři skriptu.
- Dotazy k zadání: Veškeré nejasnosti a dotazy řešte pouze prostřednictvím diskuzního fóra k projektu 2.
- Poznámka k výstupům: Informace z bodu 3-a)-i) procesu soudce se nemusí vždy objevit ve výstupu (je silně závislé na přepínání procesů).
- Poznámka k testování: Můžete si nasimulovat častější přepínání procesů např. vložení krátkého uspaní po uvolnění semaforů apod. *Pouze pro testovací účely, do finálního řešení nekládejte!*

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, ...).

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).

D. Ukázka výstupů

Ukázka 1

Spuštění `$./proj2 5 2 7 1 1`

Výstup

1	: IMM 1	: starts			
2	: IMM 1	: enters	: 1	: 0	: 1
3	: IMM 1	: checks	: 1	: 1	: 1
4	: IMM 2	: starts			
5	: IMM 2	: enters	: 2	: 1	: 2
6	: IMM 2	: checks	: 2	: 2	: 2
7	: IMM 3	: starts			
8	: IMM 3	: enters	: 3	: 2	: 3
9	: IMM 3	: checks	: 3	: 3	: 3
10	: IMM 4	: starts			
11	: IMM 4	: enters	: 4	: 3	: 4
12	: JUDGE	: wants to enter			
13	: JUDGE	: enters	: 4	: 3	: 4
14	: JUDGE	: waits for imm	: 4	: 3	: 4
15	: IMM 4	: checks	: 4	: 4	: 4
16	: JUDGE	: starts confirmation	: 4	: 4	: 4
17	: JUDGE	: ends confirmation	: 0	: 0	: 4
18	: IMM 5	: starts			
19	: IMM 1	: wants certificate	: 0	: 0	: 4
20	: IMM 2	: wants certificate	: 0	: 0	: 4
21	: IMM 1	: got certificate	: 0	: 0	: 4
22	: IMM 3	: wants certificate	: 0	: 0	: 4
23	: IMM 2	: got certificate	: 0	: 0	: 4
24	: IMM 4	: wants certificate	: 0	: 0	: 4
25	: IMM 3	: got certificate	: 0	: 0	: 4
26	: IMM 4	: got certificate	: 0	: 0	: 4
27	: JUDGE	: leaves	: 0	: 0	: 4
28	: IMM 5	: enters	: 1	: 0	: 5
29	: IMM 1	: leaves	: 1	: 0	: 4
30	: IMM 2	: leaves	: 1	: 0	: 3
31	: IMM 3	: leaves	: 1	: 0	: 2
32	: IMM 5	: checks	: 1	: 1	: 2
33	: JUDGE	: wants to enter			
34	: IMM 4	: leaves	: 1	: 1	: 1
35	: JUDGE	: enters	: 1	: 1	: 1
36	: JUDGE	: starts confirmation	: 1	: 1	: 1
37	: JUDGE	: ends confirmation	: 0	: 0	: 1
38	: IMM 5	: wants certificate	: 0	: 0	: 1
39	: IMM 5	: got certificate	: 0	: 0	: 1
40	: JUDGE	: leaves	: 0	: 0	: 1
41	: JUDGE	: finishes			
42	: IMM 5	: leaves	: 0	: 0	: 0

Další ukázky jsou v příložených souborech.