

IOS – projekt 2 (synchronizace)

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

Popis Úlohy (Pošta)

V systému máme 3 typy procesů: (0) hlavní proces, (1) poštovní úředník a (2) zákazník. Každý zákazník jde na poštu vyřídit jeden ze tří typů požadavků: listovní služby, balíky, peněžní služby. Každý požadavek je jednoznačně identifikován číslem (*dopisy:1, balíky:2, peněžní služby:3*). Po příchodu se zařadí do fronty dle činnosti, kterou jde vyřídit. Každý úředník obsluhuje všechny fronty (vybírá pokaždé náhodně jednu z front). Pokud aktuálně nečeká žádný zákazník, tak si úředník bere krátkou přestávku. Po uzavření pošty úředníci dokončí obsluhu všech zákazníků ve frontě a po vyprázdnění všech front odhází domů. Případní zákazníci, kteří přijdou po uzavření pošty, odcházejí domů (zítra je také den).

Podrobná specifikace úlohy

Spuštění:

\$./proj2 NZ NU TZ TU F

- NZ: počet zákazníků
- NU: počet úředníků
- TZ: Maximální čas v milisekundách, po který zákazník po svém vytvoření čeká, než vejde na poštu (eventuálně odchází s nepořízenou). $0 \leq TZ \leq 10000$
- TU: Maximální délka přestávky úředníka v milisekundách. $0 \leq TU \leq 100$
- F: Maximální čas v milisekundách, po kterém je uzavřena pošta pro nově příchozí. $0 \leq F \leq 10000$

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 čísly od jedničky.
- Použijte sdílenou paměť pro implementaci čítače akcí a sdílených proměnných nutných pro synchronizaci.
- Použijte semaforem pro synchronizaci procesů.
- Nepoužívejte aktivní čekání (včetně cyklického časového uspání procesu) pro účely synchronizace.

- Pracujte s procesy, ne s vlákny.

Hlavní proces

- Hlavní proces vytváří ihned po spuštění NZ procesů zákazníků a NU procesů úředníků.
- Čeká pomocí volání `usleep` náhodný čas v intervalu $<F/2, F>$
- Vypíše: *A: closing*
- 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 Zákazník

- Každý zákazník je jednoznačně identifikován číslem `idZ`, $0 < idZ \leq NZ$
- Po spuštění vypíše: *A: Z idZ: started*
- Následně čeká pomocí volání `usleep` náhodný čas v intervalu $<0, TZ>$
- Pokud je pošta uzavřena
 - Vypíše: *A: Z idZ: going home*
 - Proces končí
- Pokud je pošta otevřená, náhodně vybere činnost `X`---číslo z intervalu $<1, 3>$
 - Vypíše: *A: Z idZ: entering office for a service X*
 - Zařadí se do fronty `X` a čeká na zavolání úředníkem.
 - Vypíše: *Z idZ: called by office worker*
 - Následně čeká pomocí volání `usleep` náhodný čas v intervalu $<0, 10>$ (synchronizace s úředníkem na dokončení žádosti není vyžadována).
 - Vypíše: *A: Z idZ: going home*
 - Proces končí

Proces Úředník

- Každý úředník je jednoznačně identifikován číslem `idU`, $0 < idU \leq NU$
- Po spuštění vypíše: *A: U idU: started*
[začátek cyklu]
- Úředník jde obsloužit zákazníka z fronty `X` (vybere náhodně libovolnou neprázdnou).
 - Vypíše: *A: U idU: serving a service of type X*
 - Následně čeká pomocí volání `usleep` náhodný čas v intervalu $<0, 10>$
 - Vypíše: *A: U idU: service finished*
 - Pokračuje na [začátek cyklu]
- Pokud v žádné frontě nečeká zákazník a pošta je otevřená vypíše
 - Vypíše: *A: U idU: taking break*
 - Následně čeká pomocí volání `usleep` náhodný čas v intervalu $<0, TU>$
 - Vypíše: *A: U idU: break finished*
 - Pokračuje na [začátek cyklu]
- Pokud v žádné frontě nečeká zákazník a pošta je zavřená
 - Vypíše: *A: U idU: going home*
 - 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 vložím 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 3 2 100 100 100
```

```
1: U 1: started
2: Z 3: started
3: Z 1: started
4: Z 1: entering office for a service 2
5: U 2: started
6: Z 2: started
```

7: Z 3: entering office for a service 1
8: Z 1: called by office worker
9: U 1: serving a service of type 2
10: U 1: service finished
11: Z 1: going home
12: Z 3: called by office worker
13: U 2: serving a service of type 1
14: U 1: taking break
15: closing
16: U 1: break finished
17: U 1: going home
18: Z 2: going home
19: U 2: service finished
20: U 2: going home
21: Z 3: going home