IOS – projekt 2 (synchronizace)

Popis Úlohy (Přívoz)

V systému máme 4 typy procesů: (0) hlavní proces, (1) přívoz, (2) osobní auto a (3) nákladní auto. Přívoz jezdí mezi dvěma přístavy a převáží auta. Každé auto (nákladní i osobní) přijede do jednoho z přístavů a čeká ve frontě na volný přívoz. Nákladní auto zabírá prostor pro 3 osobní auta. Po příjezdu přívozu jsou nejprve vyložena všechna auta z přívozu na pevninu a následně jsou střídavě (vždy jedno nákladní, jedno osobní) nakládána nákladní a osobní auta do kapacity přívozu--pokud už se na přívoz nevejde nákladní auto, ale vejde se ještě osobní, tak je naloženo osobní. Pokud žádné auto nečeká na naložení, přívoz může odjet i poloprázdný/prázdný. Přívoz jezdí do doby, dokud nejsou všechna auta převezena. Pak odjíždí do doku.

Podrobná specifikace úlohy

Spuštění:

$ ./proj2 N O K TA TP

- N: počet nákladních aut, N<10000
- O: počet osobních aut, O<10000
- K: kapacita přívozu (maximální počet naložených osobních aut), 3<=K<=100
- TA: Maximální doba jízdy auta do přístavu v mikrosekundách, 0<=TA<=10000
- TP: Maximální doba jízdy přívozu v mikrosekundách, 0<=TP<=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+N+O (hlavní proces, přívoz, N nákladních aut a O osobních aut). 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 aut na přívoz použijte semafory.

- Pracujte s procesy, ne s vlákny.
- Auta se nemusí (ale mohou) naloďovat na přívoz v pořadí, ve kterém přijela do přístavu. Vylodění může být libovolném pořadí.
- Auto se může nalodit/vylodit pouze během stání přívozu v přístavu---výpis *O/N: boarding* (*O/N: leaving*) bude ve výstupním souboru pouze mezi příslušnými výpisy *T: arrived to* a *T: leaving*.
- Nejprve se vyloďuje a pak se teprve naloďuje. Vypisy *boarding* a *leaving* nemohou být libovolně proložené.
- Přístavy jsou jednoznačně identifikovány čísly 0 a 1.

Hlavní proces

- Proces vytváří ihned po spuštění proces přívoz.
- Dále vytvoří procesy pro O osobních aut a N nákladních aut.
- Každému autu při jeho spuštění náhodně přidělí nástupní přístav.
- 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 přívoz

- Po spuštění vypíše: *A: P: started*
- *PR=0 (identifikace přístavu)*
- (*) Jede do přístavu PR---čeká pomocí volání usleep náhodný čas v intervalu <0,TP>.
- Vypíše: *A: P: arrived to PR*
- Nechá vylodit všechna auta na pevninu.
- Nechá nalodit čekající auta na přívoz do jeho kapacity.
- Vypíše: *A: P: leaving PR*
- PR=(PR+1)%2
- Pokud není přívoz prázdný, nebo mohou do přístavu ještě přijet nějaká auta, tak pokračuj bodem (*)
- Jinak jede do doků---čeká pomocí volání usleep náhodný čas v intervalu <0,TP>
- Vypíše: *A: P: finish*
- Proces končí

pozn.: Doba jízdy být v každém kole jiná.

Nákladní auto

- Každé nákladní auto je jednoznačně identifikován číslem idN, 0<idN<=N
- Po spuštění vypíše: *A: N idN: started*
- Jede do přiděleného přístavu idP---čeká v intervalu <0,TA> mikrosekund.
- Vypíše: *A: N idN: arrived to idP*
- Čeká na příjezd přívozu.
- Po příjezdu přívozu se nalodí (pokud je volná kapacita)
- Vypíše: *A: N idN: boarding*
- Vyčká na příjezd do druhého přístavu P2=(idP+1)%2.
- Vylodí se a vypíše: *A: N idN: leaving in P2*
- Proces končí.

Osobní auto

- Každé osobní auto je jednoznačně identifikován číslem idO, 0<idO<=O
- Chová se stejně jako nákladní auto, akorát namísto *A: N idN …* vypisuje *A: O idO …*

  Tedy: *A: O idO: started, A: O idO: arrived to idP, A: N idO: boarding, A: O idO: leaving in P2*

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 osobni_auto, konfigurační soubor, dynamická knihovna vozidlo, ...).
- 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 4 4 10 10 10
```

```
1: P: started
2: O 1: started
3: O 2: started
4: P: arrived to 0
5: O 1: arrived to 0
6: O 1: boarding
7: P: leaving 0
8: O 4: started
9: O 2: arrived to 1
10: N 1: started
11: P: arrived to 1
12: O 1: leaving in 1
13: O 3: started
14: O 2: boarding
15: N 2: started
16: N 3: started
17: N 4: started
18: N 3: arrived to 1
19: P: leaving 1
20: O 4: arrived to 0
21: N 4: arrived to 0
22: O 3: arrived to 1
23: P: arrived to 0
24: N 2: arrived to 1
25: N 1: arrived to 0
26: O 2: leaving in 0
27: N 1: boarding
28: N 4: boarding
29: O 4: boarding
30: P: leaving 0
31: P: arrived to 1
32: N 1: leaving in 1
33: N 4: leaving in 1
34: O 4: leaving in 1
35: N 2: boarding
36: N 3: boarding
37: O 3: boarding
38: P: leaving 1
39: P: arrived to 0
40: N 3: leaving in 0
41: O 3: leaving in 0
42: N 2: leaving in 0
43: P: leaving 0
44: P: finish
```