2. Úloha IOS (2017/18)

A. Popis úlohy

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

Osoby (*riders*) přicházejí postupně na zastávku a čekají na autobus (*bus*). V okamžiku příjezdu autobusu nastoupí všechny čekající osoby. Kapacita autobusu je limitována. Pokud je na nástupišti více osob, než je kapacita autobusu, čekají osoby, které se již do autobusu nevešly, na další autobus. Pokud stojí na zastávce autobus a nastupují osoby přítomné na zastávce, další příchozí osoby vždy čekají na další autobus. Po nastoupení osob autobus odjíždí. Pokud při příjezdu na zastávku nikdo nečeká, autobus odjíždí prázdný.

Příklad: Kapacita autobusu je 10, čekajících osob je 8. Přijíždí autobus, osoby nastupují, přicházejí další 3 osoby. Nastoupí pouze 8 čekajících osob, autobus odjíždí a nově příchozí 3 osoby čekají na zastávce.

B. Podrobná specifikace úlohy

Spuštění

```
$ ./proj2 R C ART ABT
```

kde

- `R` je počet procesů *riders*; `A > 0`.
- `C` je kapacita autobusu; `C > 0`.
- `ART` je maximální hodnota doby (v milisekundách), po které je generován nový proces *rider*; `ART >= 0 && ART <= 1000`.
- `ABT` je maximální hodnota doby (v milisekundách), po kterou proces *bus* simuluje jízdu; `ABT >= 0 && ABT <= 1000`.
- Všechny parametry jsou celá čísla.

Implementační detaily

- *Pracujte s procesy, ne s vlákny.*
- Každé osobě odpovídá jeden proces *rider*.
- Autobus je reprezentován procesem *bus*. V systému je právě jeden autobus.
- Hlavní proces vytváří ihned po spuštění jeden proces *bus* a jeden pomocný proces pro generování procesů *rider*. 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
  - *rider*: pomocný proces generuje procesy pro osoby; každý nový proces je generován po uplynutí náhodné doby z intervalu `<0, ART>`; celkem vygeneruje `R` procesů. Pokud platí `ART==0`, všechny příslušné procesy se vygenerují ihned.
  - Každý proces *rider* bude interně identifikován celým číslem `I`, začínajícím od `1`. Číselná řada je pro každou kategorii procesů zvlášť.
  - Postupně tedy vznikne hlavní proces, jeden pomocný proces, `R` procesů osob *rider* a jeden proces *bus*.

- Každý proces *bus* i *rider* 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 semafory 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. `BUS` pro *bus* a `RID` pro *rider*,
- `I` je interní identifikátor procesu v rámci příslušné kategorie,
- `CR` je počet procesů *rider* aktuálně přítomných na zastávce,
- při vyhodnocování výstupu budou ignorovány mezery a tabelátory.

Proces *bus*

1. Po spuštění tiskne `A: NAME: start`
2. Proces přijíždí na zastávku. Po příjezdu tiskne `A: NAME: arrival`. *Po příjezdu autobusu na zastávku nemohou nově příchozí procesy rider vstoupit na zastávku a čekají na odjezd autobusu.*
3. Nastupování do autobusu a odjezd
   (a) Pokud jsou na zastávce osoby (procesy *rider*) – tiskne `A: NAME: start boarding: CR` a čeká na nastoupení osob. Jakmile je naplněna kapacita autobusu, příp. již nikdo nemůže nastoupit, tiskne `A: NAME: end boarding: CR` a odjíždí.
   (b) Pokud na zastávce nikdo není, autobus odjíždí ihned.
4. Těsně před odjezdem tiskne `A: NAME: depart`
5. Proces simuluje jízdu tak, že se uspí na náhodnou dobu z intervalu `<0, ABT>` a poté tiskne `A: NAME: end`. Pokud platí `ABT==0`, proces se neuspí.
6. Celý cyklus se opakuje od bodu 2 dokud není splněna podmínka ukončení.
7. Podmínka ukončení: všechny procesy *rider* již byly vygenerovány a žádný proces *rider* nečeká na zastávce ani na vstup na zastávku.
8. Po splnění podmínky ukončení tiskne proces `A: NAME: finish` a ukončí se.

Proces *rider*

1. Po spuštění tiskne `A: NAME I: start`
2. Pokud je na zastávce autobus, čeká na jeho odjezd a poté vstoupí na zastávku. Jinak vstoupí na zastávku ihned.
3. Ihned po vstupu na zastávku tiskne `A: NAME I: enter: CR` (*do CR se započítá i tento proces*) a čeká na autobus.
4. Při vstupování do autobusu tiskne `A: NAME I: boarding`
5. Po ukončení jízdy autobusu (= proces *bus* vytiskl zprávu `...end`) se proces ukončí, před ukončením tiskne `A: NAME I: finish`

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.

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).

Odevzdání

- Součástí odevzdání budou pouze soubory se zdrojovými kódy (`*.c`, `*.h`) a soubor `Makefile`. Tyto soubory zabalte pomocí nástoje 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 2 10
```

Výstup

```
1         : BUS          : start
2         : BUS          : arrival
3         : BUS          : depart
4         : RID 1        : start
5         : RID 1        : enter: 1
6         : RID 2        : start
7         : RID 2        : enter: 2
8         : RID 3        : start
9         : BUS          : end
10        : BUS          : arrival
11        : BUS          : start boarding: 2
12        : RID 1        : boarding
13        : RID 2        : boarding
14        : BUS          : end boarding: 0
15        : BUS          : depart
16        : RID 3        : enter: 1
17        : RID 4        : start
18        : RID 4        : enter: 2
19        : RID 5        : start
20        : BUS          : end
21        : BUS          : arrival
22        : BUS          : start boarding: 2
23        : RID 1        : finish
24        : RID 3        : boarding
25        : RID 2        : finish
26        : RID 4        : boarding
27        : BUS          : end boarding: 0
28        : BUS          : depart
29        : RID 5        : enter: 1
30        : BUS          : end
31        : BUS          : arrival
32        : BUS          : start boarding: 1
33        : RID 4        : finish
34        : RID 5        : boarding
35        : RID 3        : finish
36        : BUS          : end boarding: 0
37        : BUS          : depart
38        : BUS          : end
39        : BUS          : finish
40        : RID 5        : finish
```