

Implementace diskrétního simulátoru s podporou SHO

IMS projekt — 4. zadání

4. prosince 2020

Pavel Bednář (xbedna73@stud.fit.vutbr.cz)

Matúš Višcor (xvisco00@stud.fit.vutbr.cz)

Obsah

Úvod	3
Koncepce knihovny	3
Abstraktní model demo příkladu	5
Testování knihovny a experimenty s demo příkladem	6
Závěr	7

Úvod

Cílem této práce je vytvoření knihovny pro základní diskrétní simulaci s podporou systémů hromadné obsluhy. Inspirací pro tuto knihovnu je knihovna SIMLIB [1] a přenáškové slidy předmětu IMS [2]. Diskrétní simulace je typ simulace, která probíhá v diskrétním čase. Dá se pomoci ní modelovat např. vytíženost pokladen v samoobsluze a podle toho přizpůsobit počet pokladen a pokladních. Jedny z hlavních prvků diskrétní simulace jsou procesy a události. Událost je akce, která změní stav systému. Proces je pak nějaká posloupnost událostí. Systémy hromadné obsluhy (SHO) se skládají z front, zařízení a skladu. Zařízení může mít vlastní frontu nebo sdílet frontu s jiným zařízením. Funguje zde prioritizace procesu a obsluhy. Priorita procesu ovlivňuje řazení do front. Priorita obsluhy představuje právo být obsluhován. Znamená to, že pokud je v obsluze proces s nižší prioritou obsluhy než proces, který žádá o linku, stávající proces bude přeřazen do fronty přerušených procesů a proces s vyšší prioritou obsluhy dostane linku. Rozdíl mezi zařízením a skladem je, že ve skladu figuruje pouze prioritizace procesu a v obsluze může být více procesů.

Koncepce knihovny

Jádro simulátoru tvoří diskrétní *next event* algoritmus definovaný takto:

```
while (kalendář neprázdný) {  
    vytáhni z kalendáře první prvek  
    if (čas prvního prvku > čas konce simulace)  
        konec simulace  
    nastav aktuální čas na čas prvního prvku  
    proved' popis chování prvního prvku  
}
```

Chování procesu je implementováno tak, že při první exekuci procesu se všechny metody v metodě *behavior* vloží do vektoru chování. Při dalších spuštěních procesu se již provádějí akce z vektoru chování. Toto je z důvodu, aby bylo možné implementovat přerušení procesů. Události (např. generování nových procesů), které nejsou přerušitelné se

Metoda	Popis metody
setPriority	Nastaví prioritu procesu
passivate	Pasivuje proces
activate	Aktivuje proces za aktuální čas + argument
enter	Požádání o přístup k lince (zařízení i sklad)
leave	Vystoupení z linky (zařízení i sklad)
wait	Čekání procesu
setProcessType	Nastavení jméno procesu
printStats	Tisk všech stavů linky
printText	Debugovací výpis statického textu
in	Značí začátek procesu (značka pro sbírání statistik procesu)
out	Značí konec procesu (značka pro sbírání statistik procesu)
setActivationTime	Nastavení aktivace, určené pro volání z externího objektu, např. generátoru procesů
getActivationTime	Čas aktivace, určené pro volání z externího objektu, např. generátoru procesů

TABULKA 1: PŘEHLED METOD TŘÍDY PROCESS.

Metoda	Popis metody
setActivationTime	Nastavení aktivace, určené pro volání z externího objektu, např. generátoru procesů
getActivationTime	Čas aktivace, určené pro volání z externího objektu, např. generátoru procesů

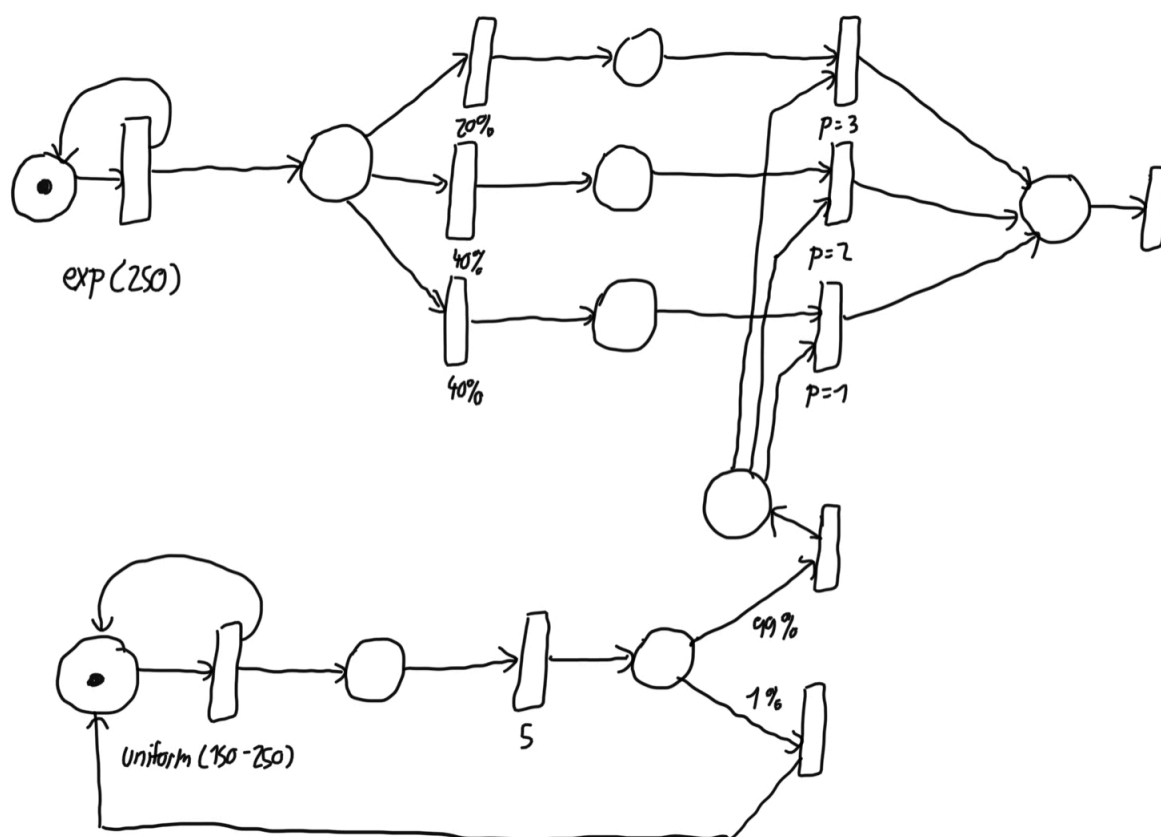
TABULKA 2: PŘEHLED METOD TŘÍDY EVENT.

naplánují a volá se jejich metoda *behavior*. Dvě hlavní třídy ze kterých uživatel vytváří pomocí dědičnosti své třídy jsou *process* a *event*. U obou je povinné implementovat virtuální metodu *behavior*, která definuje chování, jinak dojde k vyvolání výjimky třídy *simExp*. Přehled metod procesu volatelných z *behavior* je v tabulce 1, událostí (třída *event*) pak v tabulce 2. Na podporu systémů hromadné obluhy jsou zde třídy

processQueue, *facility* a *store* určené pro frontu, zařízení a sklad. Jednu frontu je možné přiřadit více zařízením. Simulace se spouští funkcí *startSim*, které se jako parametr předá čas ukončení simulace. Dále je implementována podpora generátoru rovnoměrného rozložení hustoty pravděpodobností funkcí *uniform* a exponenciálního rozložení hustoty pravděpodobností funkcí *exponencial*. Také je k dispozici třída pro statistiky procesu *stats*.

Abstraktní model demo příkladu

Demo příklad je převzat z [3]. Slovní zadání zní: V intervalech daných exponenciálním rozložením se středem 250 ms vzniká potřeba na odesílání zprávy. Existují 3 typy zpráv, které se objevují s různou pravděpodobností: s vysokou prioritou (20 %), se střední prioritou (40 %) a s malou prioritou (20 %). Odesílání zprávy linkou trvá 150-250 ms rovnoměrně. Pak následuje dotaz na správnost přenosu který trvá 5 ms. V 1 % případů dojde k chybě a celý přenos se opakuje. Abstraktní model tohoto systému vyjádřený pomocí Petriho sítí je na obrázku 1.



OBRÁZEK 1: AM DEMONSTRAČNÍHO PŘÍKLADU

Testování knihovny a experimenty s demo příkladem

Testování probíhalo tak, že jsme generovali různé procesy na různých linkách s různými typy priorit. Pro ilustraci, na obrázku 2, je vidět výpis statistik metodou `process::printStats`. TIME značí čas události, +/- značí jestli se jedná o odchod nebo o příchod, TYPE je popis typu procesu, pokud je uživatelem zadáný a Pokladna je vizualizace linky pokladna, kde před svislou čarou je ID procesu v obsluze a za touto čarou jsou ID procesů ve frontě. V ilustračním modelu jsou 3 druhy procesů: zákazník (generuje se od času 0 každé 2 časové jednotky a stráví na zařízení pokladna 5 časových jednotek), důchodce (generuje se pouze v čase 7, stráví na zařízení pokladna 7 časových jednotek a má prioritu procesu 1) a opraváři (generují se pouze v čase 23, stráví na zařízení pokladna 9 časových jednotek a mají prioritu obsluhy 1). V našem testování funkčnosti knihovny schválně nebyl použit žádný generátor pravděpodobnostního rozdělení, abychom si mohli přesně overit jestli u procesů sedí čas příchodu, odchodu a doba na lince. Generátory jsme testovali samostatně.

```
TIME: 00000.00 (+) ID: 000 TYPE: zakaznik Pokladna: [000]
TIME: 00002.00 (+) ID: 001 TYPE: zakaznik Pokladna: [000][001]
TIME: 00004.00 (+) ID: 002 TYPE: zakaznik Pokladna: [000][001][002]
TIME: 00006.00 (-) ID: 000 TYPE: zakaznik Pokladna: [001][002]
TIME: 00008.00 (+) ID: 003 TYPE: zakaznik Pokladna: [001][002][003]
TIME: 00010.00 (+) ID: 004 TYPE: duchodce Pokladna: [001][004][002][003]
TIME: 00012.00 (+) ID: 005 TYPE: zakaznik Pokladna: [001][004][002][003][005]
TIME: 00014.00 (-) ID: 001 TYPE: zakaznik Pokladna: [004][002][003][005]
TIME: 00016.00 (+) ID: 006 TYPE: zakaznik Pokladna: [004][002][003][005][006]
TIME: 00018.00 (+) ID: 007 TYPE: zakaznik Pokladna: [004][002][003][005][006][007]
TIME: 00020.00 (+) ID: 008 TYPE: zakaznik Pokladna: [004][002][003][005][006][007][008]
TIME: 00022.00 (-) ID: 009 TYPE: zakaznik Pokladna: [004][002][003][005][006][007][008][009]
TIME: 00024.00 (+) ID: 004 TYPE: duchodce Pokladna: [002][003][005][006][007][008][009]
TIME: 00026.00 (+) ID: 010 TYPE: zakaznik Pokladna: [002][003][005][006][007][008][009][010]
TIME: 00028.00 (+) ID: 011 TYPE: zakaznik Pokladna: [002][003][005][006][007][008][009][010][011]
TIME: 00030.00 (-) ID: 002 TYPE: zakaznik Pokladna: [003][005][006][007][008][009][010][011]
TIME: 00032.00 (+) ID: 012 TYPE: zakaznik Pokladna: [003][005][006][007][008][009][010][011][012]
TIME: 00034.00 (+) ID: 013 TYPE: opravari Pokladna: [013][005][006][007][008][009][010][011][012]
TIME: 00036.00 (+) ID: 014 TYPE: zakaznik Pokladna: [013][005][006][007][008][009][010][011][012][014]
TIME: 00038.00 (+) ID: 015 TYPE: zakaznik Pokladna: [013][005][006][007][008][009][010][011][012][014][015]
TIME: 00040.00 (-) ID: 003 TYPE: zakaznik Pokladna: [013][005][006][007][008][009][010][011][012][014][015]
TIME: 00042.00 (+) ID: 016 TYPE: zakaznik Pokladna: [013][005][006][007][008][009][010][011][012][014][015][016]
TIME: 00044.00 (+) ID: 017 TYPE: zakaznik Pokladna: [013][005][006][007][008][009][010][011][012][014][015][016][017]
TIME: 00046.00 (-) ID: 013 TYPE: opravari Pokladna: [003][005][006][007][008][009][010][011][012][014][015][016][017]
TIME: 00048.00 (+) ID: 018 TYPE: zakaznik Pokladna: [003][005][006][007][008][009][010][011][012][014][015][016][017][018]
TIME: 00050.00 (+) ID: 019 TYPE: zakaznik Pokladna: [003][005][006][007][008][009][010][011][012][014][015][016][017][018][019]
TIME: 00052.00 (-) ID: 009 TYPE: zakaznik Pokladna: [005][006][007][008][009][010][011][012][014][015][016][017][018][019]
TIME: 00054.00 (+) ID: 020 TYPE: zakaznik Pokladna: [005][006][007][008][009][010][011][012][014][015][016][017][018][019][020]
TIME: 00056.00 (+) ID: 021 TYPE: zakaznik Pokladna: [005][006][007][008][009][010][011][012][014][015][016][017][018][019][020][021]
TIME: 00058.00 (+) ID: 022 TYPE: zakaznik Pokladna: [005][006][007][008][009][010][011][012][014][015][016][017][018][019][020][021][022]
TIME: 00060.00 (-) ID: 005 TYPE: zakaznik Pokladna: [006][007][008][009][010][011][012][014][015][016][017][018][019][020][021][022]
TIME: 00062.00 (+) ID: 023 TYPE: zakaznik Pokladna: [006][007][008][009][010][011][012][014][015][016][017][018][019][020][021][022][023]
TIME: 00064.00 (-) ID: 006 TYPE: zakaznik Pokladna: [007][008][009][010][011][012][014][015][016][017][018][019][020][021][022][023][024]
TIME: 00066.00 (+) ID: 025 TYPE: zakaznik Pokladna: [007][008][009][010][011][012][014][015][016][017][018][019][020][021][022][023][024][025]
TIME: 00068.00 (+) ID: 026 TYPE: zakaznik Pokladna: [007][008][009][010][011][012][014][015][016][017][018][019][020][021][022][023][024][025][026]
TIME: 00070.00 (+) ID: 027 TYPE: zakaznik Pokladna: [007][008][009][010][011][012][014][015][016][017][018][019][020][021][022][023][024][025][026][027]
TIME: 00072.00 (-) ID: 007 TYPE: zakaznik Pokladna: [008][009][010][011][012][014][015][016][017][018][019][020][021][022][023][024][025][026][027]
TIME: 00074.00 (+) ID: 028 TYPE: zakaznik Pokladna: [008][009][010][011][012][014][015][016][017][018][019][020][021][022][023][024][025][026][027][028]
TIME: 00076.00 (+) ID: 029 TYPE: zakaznik Pokladna: [008][009][010][011][012][014][015][016][017][018][019][020][021][022][023][024][025][026][027][028][029]
TIME: 00078.00 (-) ID: 008 TYPE: zakaznik Pokladna: [009][010][011][012][014][015][016][017][018][019][020][021][022][023][024][025][026][027][028][029]
TIME: 00080.00 (+) ID: 030 TYPE: zakaznik Pokladna: [009][010][011][012][014][015][016][017][018][019][020][021][022][023][024][025][026][027][028][029][030]
TIME: 00082.00 (+) ID: 031 TYPE: zakaznik Pokladna: [009][010][011][012][014][015][016][017][018][019][020][021][022][023][024][025][026][027][028][029][030][031]
TIME: 00084.00 (+) ID: 032 TYPE: zakaznik Pokladna: [009][010][011][012][014][015][016][017][018][019][020][021][022][023][024][025][026][027][028][029][030][031][032]
TIME: 00086.00 (-) ID: 009 TYPE: zakaznik Pokladna: [010][011][012][014][015][016][017][018][019][020][021][022][023][024][025][026][027][028][029][030][031][032]
TIME: 00088.00 (+) ID: 033 TYPE: zakaznik Pokladna: [010][011][012][014][015][016][017][018][019][020][021][022][023][024][025][026][027][028][029][030][031][032][033]
TIME: 00090.00 (+) ID: 034 TYPE: zakaznik Pokladna: [010][011][012][014][015][016][017][018][019][020][021][022][023][024][025][026][027][028][029][030][031][032][033][034]
TIME: 00092.00 (-) ID: 010 TYPE: zakaznik Pokladna: [011][012][014][015][016][017][018][019][020][021][022][023][024][025][026][027][028][029][030][031][032][033][034]
TIME: 00094.00 (+) ID: 035 TYPE: zakaznik Pokladna: [011][012][014][015][016][017][018][019][020][021][022][023][024][025][026][027][028][029][030][031][032][033][034][035]
TIME: 00096.00 (+) ID: 036 TYPE: zakaznik Pokladna: [011][012][014][015][016][017][018][019][020][021][022][023][024][025][026][027][028][029][030][031][032][033][034][035][036]
TIME: 00098.00 (+) ID: 037 TYPE: zakaznik Pokladna: [011][012][014][015][016][017][018][019][020][021][022][023][024][025][026][027][028][029][030][031][032][033][034][035][036][037]
TIME: 00100.00 (-) ID: 011 TYPE: zakaznik Pokladna: [012][014][015][016][017][018][019][020][021][022][023][024][025][026][027][028][029][030][031][032][033][034][035][036][037]
TIME: 00102.00 (+) ID: 038 TYPE: zakaznik Pokladna: [012][014][015][016][017][018][019][020][021][022][023][024][025][026][027][028][029][030][031][032][033][034][035][036][037][038]
TIME: 00104.00 (+) ID: 039 TYPE: zakaznik Pokladna: [012][014][015][016][017][018][019][020][021][022][023][024][025][026][027][028][029][030][031][032][033][034][035][036][037][038][039]
TIME: 00106.00 (-) ID: 012 TYPE: zakaznik Pokladna: [014][015][016][017][018][019][020][021][022][023][024][025][026][027][028][029][030][031][032][033][034][035][036][037][038][039]
TIME: 00108.00 (+) ID: 040 TYPE: zakaznik Pokladna: [014][015][016][017][018][019][020][021][022][023][024][025][026][027][028][029][030][031][032][033][034][035][036][037][038][039][040]
TIME: 00110.00 (+) ID: 041 TYPE: zakaznik Pokladna: [014][015][016][017][018][019][020][021][022][023][024][025][026][027][028][029][030][031][032][033][034][035][036][037][038][039][040][041]
TIME: 00112.00 (+) ID: 042 TYPE: zakaznik Pokladna: [014][015][016][017][018][019][020][021][022][023][024][025][026][027][028][029][030][031][032][033][034][035][036][037][038][039][040][041][042]
```

OBRÁZEK 2: VIZUALIZACE STAVU LINKY METODOU PRINTSTATS()

Vypsání statistik procesu se provede tak, že pro každou třídu procesu se vytvoří objekt třídy *stats* a do metody procesu *behavior* se na začátek vloží metoda *in* s argumentem, který je ukazatel na objekt třídy *stats*. Analogicky se vloží metoda *out* na konec *behavior*. Po konci simulace se statistiky vypíší metodou *stats::printStats*. Metodou *facility::printUse* získáme statistiku o využití linky. Z demo příkladu při délce simulace 11500 časových jednotek jsme zjistili následující. Průměrný čas procesu na lince byl 200, minimum 155 a maximum 409. Průměrná doba života procesu v systému byla 271, minimum 157 a maximum 706. Vytížení linky bylo 69 %.

Závěr

Naši knihovnu jsme otestovali a demonstrovali na výše zmíněných příkladech a získali výše zmíněné výsledky.

Literatura

- [1] Peringer P., Leska D., Martinek D.: SIMLIB/C++. [online], 19. říjen 2018. Dostupné z: <https://www.fit.vutbr.cz/~peringer/SIMLIB/>
- [2] Peringer P., Hrubý M.: Modelování a simulace. [online], 21. září 2020. Dostupné z: <https://www.fit.vutbr.cz/study/courses/IMS/public/prednasky/IMS.pdf>
- [3] Hrubý M.: 1. democvičení z Modelování a simulace. 21. října 2020.