



**FAKULTA
APLIKOVANÝCH VĚD
ZÁPADOČESKÉ
UNIVERZITY
V PLZNI**

Semestrální práce z KIV/DS

SEQUENCER

David Košek
A14N0132P
kosek@students.zcu.cz

1. Zadání

- Implementace v C/C++.
- Překlad programem make.
- Žádné GUI, jen příkazový řádek, neinteraktivní - spustit, spočítat, zalogovat. OS Linux.
- Paralelizace - volitelně vlákna nebo procesy (dále v textu jen vlákna).
- Aplikace se sestává ze 2 spustitelných programů - banka a sekvencer.
- Na 1 sekvencer se může připojit více bank.
- Sekvencer obsahuje frontu, do které se přidávají přijaté operace a odebírají k rozeslání.
- Banka v náhodných intervalech vysílá operace na sekvencer.
- Generované operace ponosou informaci, zda se jedná o příjem, výdej nebo úročení a v jaké výši.
- Sekvencer bude rozesílat přijaté operace všem připojeným bankám v pořadí, ve kterém je přijal, a se sekvenčním číslem.
- Každá banka bude mít vlastní vyrovnávací paměť přijatých operací, kterou bude zpracovávat tak, aby bylo zachováno pořadí podle sekvenčního čísla operace od sekvenceru můžou dorazit přeházené.
- Všechny banky po zaúčtování operace se sekvenčním číslem N budou mít shodný stav na účtě (toto není přímo požadavek, ale důsledek správné implementace).
- Ke komunikaci mezi klientem a sekvencerem využijte socketů a protokolu UDP bez potvrzování.
- Vygenerujte datový soubor pro net_flow_vizu.
- Pro simulaci zpoždění komunikačních linek využijte socket_retarder.

2. Implementace

2.1 Sekvencer

Při startu sekvenceru se nastaví hodnoty portu a IP adresy, na které server bude naslouchat. Poté se vytvoří dvě vlákna, první vlákno je pro přijímání zpráv a druhé pro odesílání zpráv.

Přijímací vlákno vyhodnocuje přijatou zprávu a rozhoduje mezi třemi možnostmi. V případě nové banky je bance zasláno její identifikační číslo pro pozdější vyhodnocení a je přidán její port do mapy pro komunikaci. Další možnost je, že banka ukončila svůj proces, dojde k vyjmutí jejího portu z mapy. Poslední možnost je, že přišla operace, která má být vykonána nad účtem. Dojde k přidání času přijetí zprávy, nastavení sekvenčního čísla, k zalogování a poté, pokud nikdo nepracuje nad frontou, je tato zpráva vložena do fronty.

Odesílací vlákno přistupuje do fronty, pokud není fronta používána jiným vláknem a testuje, zda – li není fronta prázdná. Pokud není, vezme se operace z fronty a pošle se všem registrovaným bankám v mapě. Poté je operace z fronty vyjmuta.

Během chodu sekvenceru jsou zobrazovány jednotlivé informační zprávy. Po ukončení sekvenceru dojde k vytvoření výsledného souboru pro vygenerování grafu. Pro vygenerování výsledného souboru je nutné sekvencer ukončit pomocí kombinace CTRL+C (viz sekce spuštění).

2.2 Banka

Při startu nové banky dojde k nastavení připojení a požádání sekvenceru o přiřazení identifikačního čísla. Poté dojde k vytvoření nových dvou vláken. Jedno slouží pro příjem zpráv a druhé pro zpracování zprávy. Hlavní vlákno nekončí a generuje zprávy, které posílá na sekvencer. Banka na začátku čeká 15s. To je čas pro připojení více bank k sekvenceru.

Příjímací vlákno pracuje podobně jako příjímací vlákno na sekvenceru. Zaeviduje se čas přijetí zprávy, a pokud nikdo nepracuje nad frontou je zpráva vložena do fronty. Může také přijít speciální zpráva, která nám sděluje ukončení sekvenceru, a tudíž také dochází k ukončení banky.

Vlákno pro zpracování zpráv přistupuje k frontě, pokud nad ní někdo nepracuje a také pokud není prázdná. Poté se ve frontě vyhledá operace podle sekvenčního čísla, která má být zpracována. Sekvenční čísla generují vzestupně. Frontu v bance jsem implementoval pomocí vektoru.

3. Spuštění

3.1 Sekvencer

Sekvencer se spouští příkazem `./Sequencer <port>`, kde port je číslo portu na kterém bude sekvencer naslouchat. IP adresa je 127.0.0.1 (lokální). Sekvencer jde jednoduše ukončit pomocí kláves CTRL+C.

3.2 Banka

Banka se spouští příkazem `./Bank <ip> <port>`, kde IP je adresa sekvenceru a port je číslo portu na kterém bude sekvencer naslouchá. Banka jde jednoduše ukončit pomocí kláves CTRL+C.

3.3 Socket retarder

Při využití socket retardéru stačí spustit `runner_example.sh`, kde jsou již nadefinovány příkazy pro spuštění dvou bank. Předtím ale musíme zajistit běh sequenceru, který spustíme v jiném terminálu. Banky lze ukončit příkazem „`killall Bank`“.

3.4 Net flow vizu dia

Ve složce log je po ukončení sekvenceru vygenerován výsledný soubor pro vygenerování grafu. Pokud je soubor poškozen (neadekvátně ukončený sekvencer nebo banka) je třeba soubor upravit ručně (smazáním posledního příkazu v bankách). Pro vygenerování grafu nám slouží příkaz „./net_flow_vizu_dia.py < data.yaml > chart.dia“

4. Závěr

Aplikaci jsem programoval v C++, kde jsem se naučil používat vector, frontu a rozšířil jsem si vědomosti v tomto programovacím jazyce. Sekvencer i banka pracují dle očekávání. Největší problém práce bylo zprovoznit aplikaci se socket retardérem, což lze nahradit uspáním vlákna. Komunikace mezi bankami a sekvencerem ukazuje výsledný graf.

