



**FAKULTA  
APLIKOVANÝCH VĚD  
ZÁPADOČESKÉ  
UNIVERZITY  
V PLZNI**

**Semestrální práce z KIV/DS**

**GLOBÁLNÍ STAV**

David Košek  
A14N0132P  
kosek@students.zcu.cz

# 1. Zadání

Banka má několik poboček. Množství hotovosti banky je konstantní, ale jednotlivé pobočky mezi sebou převáží různé obnosy v pomalých nákladních autech na definovaných trasách. Úkolem je zjistit, jaký je stav na pobočkách a kolik peněz je na cestě. To vše bez narušení/zablokování provozu poboček.

- 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 z 2 spustitelných programů - banka a sekvencer.
- Proces banka bude sloužit ke sběru snímků a kontrole jejich integrity (součet)
- Komunikace bude realizována protokolem TCP/IP, sokety výstupních kanálů budou NEblokující
- Pobočky budou v náhodných intervalech na náhodných výstupech odesílat náhodné množství hotovosti.
- Vygenerujte datový soubor pro `net_flow_vizu`.
- Pro simulaci zpoždění komunikačních linek využijte `socket_retarder`.

## 2. Implementace

### 2.1 Banka

Při startu banky se nastaví hodnota portu a IP adresy, na které server bude naslouchat. Nejprve banka čeká na připojení všech poboček (nastaveno na 3), poté všem pobočkám pošle informace o ostatních pobočkách. A nadále už jen čeká na pokyn k nastartování snímkování, který pošle první pobočce, která odstartuje snímkování mezi pobočkami. A vytvoří vlákna pro získání odpovědí od všech poboček, po získání všech hodnot je sečte a zahlásí globální stav.

### 2.2 Pobočka (Branch)

Při startu nové pobočky dojde k nastavení připojení a požádání banky o přiřazení identifikačního čísla. Poté dojde k připojení k ostatním pobočkám a nastavení přijímání od ostatních poboček do správné fronty. Pobočky si mezi sebou posílají náhodně peníze. Pokud přijde na pobočku s ID 1 požadavek o snímkování globálního stavu, přepne se do tohoto stavu a pošle zprávu o tom dalším dvěma procesům. Dále nastaví potřebné flagy a nastaví si globální stav podle lokálního stavu. Další pobočky zpracují poslaný marker a

přepnou se také do globálního stavu. Všechny pobočky si uchovávají informaci o tom, kdo jim už marker zaslal a kdo ne. Pokud mi přijde zpráva z pobočky, která ještě není v globálním stavu, tak hodnotu připočtu ještě ke globálnímu stavu. Přičtení k lokálnímu stavu je automatický. Zprávy z front čtu podle jednotlivých sekvenčních čísel podle pobočky, ze které zpráva přišla. Pokud dostanu marker od všech poboček, posílám zprávu do banky s globálním stavem a přepnu se do normálního stavu. Banka poté vyhodnotí přijaté zprávy.

## 3. Spuštění

### 3.1 Banka

Banka se spouští příkazem `./Bank <port>`, kde port je číslo portu na kterém bude banka naslouchat. IP adresa je 127.0.0.1 (lokální). Pro start snímání je třeba napsat do konzole „global“ a odeslat. Bez banky nelze připojit pobočky, banka začne možnost snímat až po připojení tří poboček. Banku jde jednoduše ukončit pomocí kláves CTRL+C.

### 3.2 Branch

Branch se spouští příkazem `./Branch <port>`, kde port je číslo portu na kterém bude pobočka naslouchat. Branch 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 tří poboček. Banka se musí pustit zvlášť. Pobočky lze ukončit příkazem „`killall Branch`“.

### 3.4 Net flow vizu dia

Ve složce log jsou po ukončení banky a pobočky dostupné textové soubory se za logovanými hodnotami. Pro vygenerování grafu nám slouží skript `create_dia.sh`.

## 4. Závěr

Banka i pobočky pracují dle očekávání. Největší problém byl ve správném zpracování zpráv v globálním stavu. Komunikace mezi procesy ukazuje výsledný graf, který je přibližný k získávání globálního stavu.

