ŽILINSKÁ UNIVERZITA V ŽILINE
Fakulta riadenia
a informatiky

Príručka pre projekt

Simulácia náhodnej pochôdzky

SEMESTRÁLNA PRÁCA

Vypracoval(a): **Patrik Šustek**
Študijná skupina: **5ZYR31**
Predmet: **Princípy operačných systémov**
Cvičiaci: **Ing. Michal Mrena, PhD.**

Obsah

1. Popis hry

Náhodná pochôdzka je jednoduchá terminálová simulácia. Cieľom je vytvoriť aplikáciu, ktorá bude simulovať pohyb opilca (radšej budeme hovoriť o chodcovi, https://en.wikipedia.org/wiki/Random_walk) v dvojrozmernom svete pozostávajúcom z políčok. Chodec so môže v každom okamžiku posunúť o jedno políčko hore, dole, vľavo alebo vpravo. Smer posunu je daný pravdepodobnosťou, ktorá predstavuje vstup do simulácie. Aplikácia umožní vypočítať pre každé políčko dvojrozmerného sveta priemerný počet krokov potrebných na dosiahnutie stredu sveta (políčko [0, 0]), ako aj pravdepodobnosť, s akou chodec dosiahne stred sveta za predpokladu, že môže vykonať najviac K posunov.

2. Štruktúra projektu

2.1. Zoznam súborov

- client.h a client.c
- server.h a server.c
- client_io.h a client_io.c
- simulation.h a simulation.c
- socket.h a socket.c (použite z cvičení predmetu POS)
- shared.h
- SMakeLists.txt

2.2. Repozitár

V repozitári boli vykonané 2 commity. Odkaz: https://github.com/Pa3ckS5/POS_SP

2.3. Popis súborov

Client

- kbhit()

Táto funkcia kontroluje, či bola stlačená klávesa na vstupe. Používa sa na detekciu užívateľského vstupu bez blokovania programu. Vracia 1, ak bola klávesa stlačená, inak vracia 0.

- check_quit(void* arg)

Toto je vlákno, ktoré nepretržite kontroluje, či používateľ stlačil klávesu. Ak je stlačená klávesa q alebo Q, vlákno sa ukončí. Používa sa na ukončenie simulácie na požiadanie používateľa.

- is_port_available(int port)

Funkcia kontroluje, či je daný port voľný a možno ho použiť pre server. Vracia true, ak je port dostupný, inak vracia false.

- start_new_simulation(SimulationConfig* config)

Táto funkcia spustí novú simuláciu na zadanom porte. Vytvára nový proces pre server a pripája sa k nemu ako klient. Odosiela konfiguráciu simulácie serveru a spravuje komunikáciu počas simulácie.

- connect_to_existing_simulation()

Funkcia umožňuje klientovi pripojiť sa k už bežiacej simulácii na zadanom porte. Prijíma správy od servera a zobrazuje stav simulácie v reálnom čase.

- display_menu()

Zobrazuje hlavné menu aplikácie s možnosťami spustenia novej simulácie, pripojenia k existujúcej simulácii, načítania simulácie zo súboru alebo ukončenia aplikácie.

- main()

Hlavná funkcia modulu, ktorá riadi chod aplikácie. Načítava používateľské vstupy a volá príslušné funkcie podľa výberu používateľa.

Server

- ThreadData

Štruktúra ThreadData slúži na prenos dát medzi vláknami. Obsahuje ukazateľ na simuláciu (Simulation*), rozsah riadkov pre spracovanie (start_row, end_row), mutex pre synchronizáciu a pole finished na sledovanie dokončených úloh.

- ClientInfo

Štruktúra ClientInfo uchováva informácie o pripojenom klientovi, vrátane soketu (client_socket), adresy klienta (client_addr) a stavu (active), ktorý indikuje, či je klient aktívny.

- Server

Štruktúra Server spravuje všetkých pripojených klientov a ich vlákna. Obsahuje pole klientov (clients), pole vlákien (client_threads), počet pripojených klientov (client_count), mutex pre synchronizáciu a flag shutdown na ukončenie servera.

- broadcast_to_clients

Táto metóda rozosiela správu (MessageToClient) všetkým aktívnym klientom. Ak sa správa neodosiela, klient je označený ako neaktívny a jeho soket sa uzatvorí.

- fast_sim_function

Metóda fast_sim_function je vláknová funkcia, ktorá vykonáva rýchlu simuláciu pre pridelený rozsah riadkov. Používa mutex na synchronizáciu prístupu k zdieľaným dátam.

- simulate_slow

Táto metóda vykonáva pomalú simuláciu, ktorá komunikuje s klientom v reálnom čase. Klient môže posielať príkazy (q, m, e), ktoré ovplyvňujú priebeh simulácie.

- simulate_fast

Metóda simulate_fast rozdeľuje simuláciu medzi viac vlákien pre paralelné spracovanie. Kontroluje vstup od klienta a posiela stav simulácie všetkým pripojeným klientom.

- handle_client

Táto metóda spracováva pripojenie klienta. Na základe prijatej konfigurácie spustí buď rýchlu alebo pomalú simuláciu a po dokončení uvoľní zdroje.

- client_handler

Metóda client_handler je vláknová funkcia, ktorá spravuje komunikáciu s jedným klientom. Po ukončení komunikácie označí klienta ako neaktívneho a uzavrie soket.

- main

Hlavná funkcia inicializuje server, čaká na pripojenie klientov a spravuje ich vlákna. Po ukončení servera čaká na dokončenie všetkých vlákien a uvoľní zdroje.

Simulation

- Simulation

Štruktúra, ktorá uchováva všetky údaje o simulácii, vrátane konfigurácie, stavu sveta, počtu úspešných pokusov a aktuálnej pozície chodca. Obsahuje aj informácie o cieľovom políčku a prekážkach.

- create_simulation(SimulationConfig* config)

Funkcia vytvára novú simuláciu na základe poskytnutej konfigurácie. Alokuje pamäť pre svet a štatistiky, nastavuje počiatočné hodnoty a generuje prekážky, ak sú povolené.

- destroy_simulation(Simulation* sim)

Funkcia uvoľňuje pamäť alokovanú pre simuláciu, vrátane polí pre svet a štatistiky. Zaisťuje správne vyčistenie zdrojov po ukončení simulácie.

- is_goal_reached(Simulation* sim)

Funkcia kontroluje, či chodec dosiahol cieľové políčko. Vracia true, ak je chodec na cieľovom políčku, inak false.

- is_max_steps_reached(Simulation* sim)

Funkcia kontroluje, či chodec dosiahol maximálny povolený počet krokov. Vracia true, ak bol limit krokov prekročený, inak false.

- random_step(Simulation* sim)

Funkcia vykoná náhodný krok chodca na základe pravdepodobností definovaných v konfigurácii. Zohľadňuje prekážky a okraje sveta.

- random_walk(Simulation* sim, int start_x, int start_y)

Funkcia simuluje náhodnú prechádzku z daného počiatočného bodu. Vracia počet krokov potrebných na dosiahnutie cieľa alebo -1, ak cieľ nebol dosiahnutý.

- change_mode(Simulation* sim)

Funkcia prepína medzi rôznymi režimami zobrazovania simulácie (napr. vizuálny, štatistický, priemerný). Režim závisí od nastavení v konfigurácii.

- get_state_print(Simulation* sim, char* buffer)

Funkcia generuje textový reťazec reprezentujúci aktuálny stav simulácie podľa aktuálneho režimu zobrazovania. Výstup sa ukladá do poskytnutého buffera.

- get_result_print(Simulation* sim, char* buffer)

Funkcia generuje textový reťazec s výsledkami simulácie, vrátane úspešnosti a priemerného počtu krokov. Výstup sa ukladá do buffera.

- get_result_for_save(Simulation* sim, char* buffer)

Funkcia generuje textový reťazec s výsledkami simulácie, ktoré sú vhodné pre uloženie do súboru. Výstup sa ukladá do buffera.

- get_visual_print(Simulation* sim, char* buffer)

Funkcia generuje vizuálne zobrazenie sveta simulácie, vrátane pozície chodca, cieľa a prekážok. Výstup sa ukladá do buffera.

- get_success_print(Simulation* sim, char* buffer)

Funkcia generuje textový reťazec s úspešnosťou dosiahnutia cieľa pre každé políčko sveta. Výstup sa ukladá do buffera.

- get_average_print(Simulation* sim, char* buffer)

Funkcia generuje textový reťazec s priemerným počtom krokov potrebných na dosiahnutie cieľa pre každé políčko sveta. Výstup sa ukladá do buffera.

- get_statistic_print(Simulation* sim, char* buffer)

Funkcia generuje textový reťazec so štatistickými údajmi o simulácii, vrátane počtu krokov, replikácií a pravdepodobností pohybu. Výstup sa ukladá do buffera.

Client_io

- int_input_check(int min, int max)

Funkcia kontroluje a validuje celočíselný vstup od používateľa. Zabezpečuje, aby vstup bol v rámci definovaného rozsahu. Vracia platný vstup.

- double_input_check(double min, double max)

Funkcia kontroluje a validuje desatinný vstup od používateľa. Zabezpečuje, aby vstup bol v rámci definovaného rozsahu. Vracia platný vstup.

- text_input_check()

Funkcia kontroluje a validuje textový vstup od používateľa. Zabezpečuje, aby vstup nebol prázdny a nepresahoval maximálnu dĺžku. Vracia platný textový reťazec.

- save_simulation(SimulationConfig* config, const char* result)

Funkcia ukladá konfiguráciu a výsledky simulácie do textového súboru. Súbor je pomenovaný podľa názvu uvedeného v konfigurácii. Vracia true, ak sa zápis podaril, inak false.

- load_simulation(const char* filename, SimulationConfig* config)

Funkcia načíta konfiguráciu a výsledky simulácie zo súboru. Zobrazuje načítané údaje a aktualizuje konfiguráciu simulácie. Vracia true, ak sa načítanie podarilo, inak false.

- input_simulation_config(SimulationConfig* config)

Funkcia interaktívne žiada používateľa o zadanie konfigurácie simulácie, vrátane rozmerov sveta, počtu krokov, pravdepodobností pohybu a názvu súboru pre uloženie výsledkov.

- update_config(SimulationConfig* config)

Funkcia umožňuje používateľovi aktualizovať počet replikácií a názov súboru pre uloženie výsledkov simulácie. Zobrazuje aktuálne hodnoty a žiada o nové vstupy.

Modul shared

- SimulationMode

Výčtový typ, ktorý definuje rôzne režimy simulácie:

- VISUAL: Normálny režim, ktorý zobrazuje vizuálne zobrazenie sveta.
- SUCCESS: Štatistický režim, ktorý zobrazuje úspešnosť dosiahnutia cieľa pre každé políčko.
- AVERAGE: Štatistický režim, ktorý zobrazuje priemerný počet krokov potrebných na dosiahnutie cieľa.

- SimulationConfig

Štruktúra, ktorá uchováva konfiguráciu simulácie. Obsahuje:

- Rozmery sveta (world_width, world_height)
- Pravdepodobnosti pohybu v štyroch smeroch (probabilities)
- Počet replikácií (total_replications)
- Maximálny počet krokov (max_steps).
- Režim simulácie (fast_mode).
- Prítomnosť prekážok (has_obstacles).
- Názov súboru pre uloženie výsledkov (filename).

- MessageType

Výčtový typ, ktorý definuje typy správ posielaných medzi serverom a klientom:

- SIMULATION_CONFIG: Správa obsahujúca konfiguráciu simulácie.
- SIMULATION_STATE: Správa obsahujúca aktuálny stav simulácie.

- SIMULATION_OVER: Správa oznamujúca ukončenie simulácie.
- SIMULATION_RESULT: Správa obsahujúca výsledky simulácie.
- SIMULATION_END: Správa požiadavka na ukončenie simulácie.
- SIMULATION_MODE: Správa požiadavka na zmenu režimu simulácie.
- SERVER_CREATED: Správa oznamujúca úspešné vytvorenie servera.
- SERVER_FULL: Správa oznamujúca, že server je plný.
- SERVER_SHUTDOWN: Správa oznamujúca vypnutie servera.

- MessageToClient

Štruktúra, ktorá definuje správu odosielanú serverom klientovi. Obsahuje:

- Typ správy (type).
- Buffer pre dodatočné údaje (buffer), ktorý môže obsahovať textové informácie alebo dáta.

- BUFFER_SIZE

Konštanta definujúca veľkosť buffera pre správy a textové údaje. Hodnota je nastavená na 30 000 bajtov.

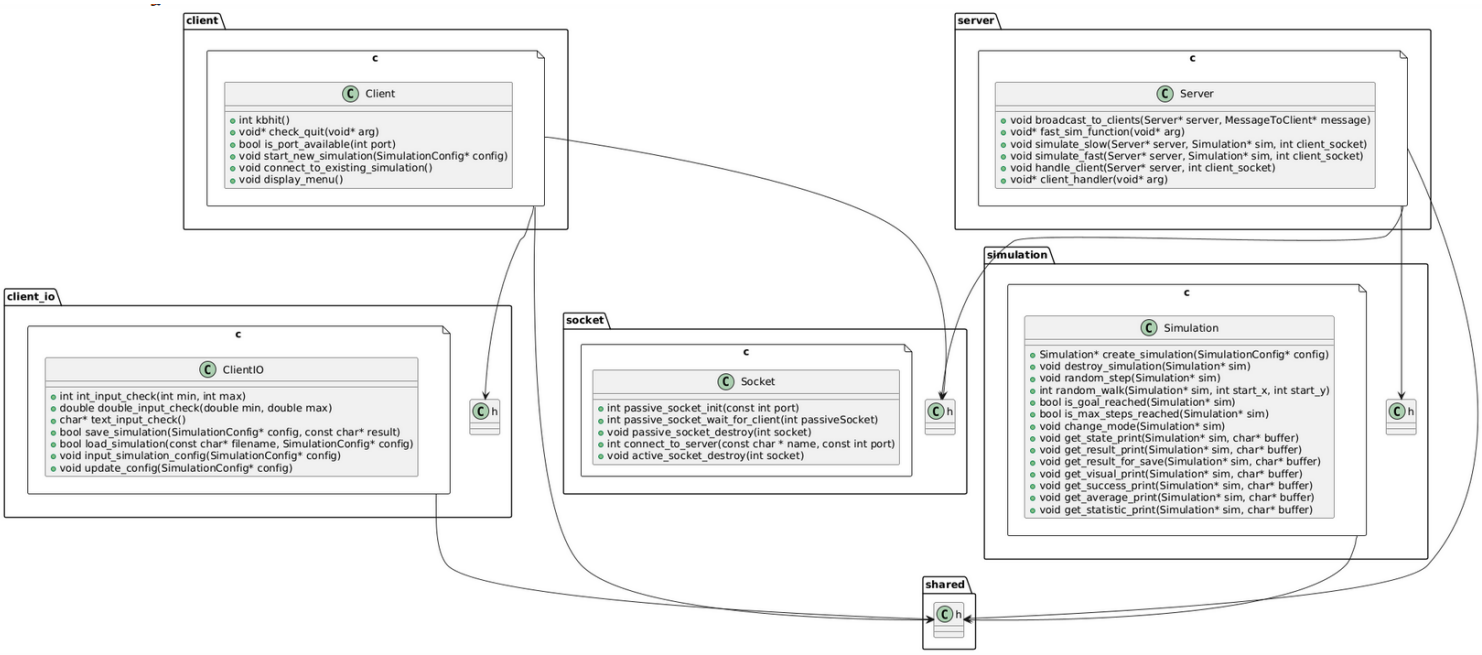
- MAX_FILENAME_LENGTH

Konštanta definujúca maximálnu dĺžku názvu súboru. Hodnota je nastavená na 31 znakov (30 znakov + ukončovací znak).

2.4. UML diagram

Na nasledujúcom obrázku je môžné vidieť vytvorený diagram UML pre program. Odkaz na použitý nástroj pre tvorbu UML diagramu :

http://www.plantuml.com/plantuml/uml/SyfFKj2rKt3CoKnELR1Io4ZDoSa70000



3. Procesy aplikácie

3.1. Serverový proces

Tento proces slúži na správu existujúcich a vytváranie nových simulácii. Kde pri vytvorení novej server inicializuje potrebné zdroje. Takisto je možné sa na hru, ktorú proces vytvára a spracúva pripojiť aj iným hráčom ak vedia jeho port.

Server takisto spracúva vstupy od hráčov, a aktualizuje stav simulácie. Pri spracovaní vstupov server kontroluje kolízie a aktualizuje pozície hadíkov.

Server po vytvorení spúšťa simuláciu pod+ poslanej konfigurácie. Danú simuláciu spustí a klientom posiela jej daný stav. Počas toho tiež spracováva vstupy ktoré posiela klient serveru. Po ukončení simulácie pošle výsledky na zobrazenie klientom a tiež výsledok simulácie pre uloženie tvorcovi simulácie. Následne sa ukončí s dealokovaním zdrojov.

Tento proces slúži na pripojenie hráča k serveru. Hráč môže potom buď vytvoriť novú simuláciu alebo sa môže pripojiť k existujúcim simuláciám. Proces spracováva vstupy od používateľov a posiela ich serveru. Slúži aj na zobrazovanie aktuálneho stavu simulácie, zadávanie vstupov, načítavanie a ukladanie do súborov.

Klient sa skúsi sa vytvoriť na serveri simuláciu, ak je voľný. Najskôr však zadá vstupy pre simuláciu. Po pripojení na server môže klient zobrazovať simuláciu, prepínať módy zobrazenia simulácie, či simuláciu zrušiť. Ak vytvoril simuláciu, nemôže z nej odísť, ale môže ju zrušiť, keďže vytvorenie simulácie klientom je ‚child' proces spustený daným klientom. Ak sa k simulácii pripojí, môže z nej odísť ale nemôže ju zrušiť. Tým chráni tvorcu simulácie, aby mu iný klient simuláciu nezrušil.

Po simulácii sa odpoja klienti a server sa zruší. Tvorcovi simulácie sa výsledky simulácie uložia do vopred zadaného súboru. Ak neexistuje súbor, vytvorí nový s príslušným názvom.

Takto uloženú simuláciu možno zo súboru načítať, zobrazia sa jej výsledky a je možné s načítanými parametrami tiež opätovne spustiť simuláciu. Klient môže pred znovuspustením zadať niektoré parametre nanovo.

4. Použitie vlákien

4.1. Vstup z klávesice klienta

Vstup od klienta je spracovávaný asynchrónne pomocou samostatného vlákna. Toto vlákno neustále kontroluje, či klient nezadal nejaký príkaz (napr. q pre ukončenie, m pre zmenu módu alebo e pre ukončenie spojenia). Tento prístup umožňuje aplikácii okamžite reagovať na vstup od užívateľa bez toho, aby blokoval hlavné vlákno, ktoré sa stará o simuláciu a komunikáciu so serverom.

4.2. Vlákna pre rýchly výpočet

Pre zrýchlenie výpočtov v rýchlom móde simulácie (fast mode) aplikácia využíva viacvláknové spracovanie. Simulácia je rozdelená na niekoľko častí, pričom každá časť je spracovávaná samostatným vláknom. Tento prístup umožňuje paralelné vykonávanie výpočtov, čo výrazne zvyšuje výkon.

4.3. Vlákna klientov

Každý klient, ktorý sa pripojí k serveru, je obsluhovaný samostatným vláknom. Tento prístup umožňuje serveru obsluhovať viac klientov súčasne bez toho, aby sa jeden klient musel čakať na dokončenie požiadaviek iného klienta.

5. Medziprocesná komunikácia (IPC)

V aplikácií sú použité schránky na medziprocesnú komunikáciu. Sú použité na komunikáciu medzi serverom a klientom. Server prijíma pripojenia klientov, zadanie

konfigurácie pre simuláciu a vstupy klientov. Klient prijíma informácie o stave hry. Server a klient komunikujú prostredníctvom a funkcií send(), select() a recv().

5.1. Obsah správ

Pri komunikácii medzi serverom sa v správach posielajú rôzne štruktúry a typy:

- SimulationConfig – odosielanie počiatočnej konfigurácie pre simuláciu
- MessageToClient – odosielanie stavu a obrazu simulácie
- char – odosielanie pokynov serveru počas simulácie

6. Problémy počas práce

Pri vypracovaní nastalo množstvo problémov. Ďalej sú vypísané tie hlavné.

Problém bol určite nedostatočné informácie a skúsenosti z cvičení. Potrebné bolo si veci doštudovať a množstvo vecí len skúšať. Problém bol tiež nájsť informácie o danej téme schránok a správne ich použiť v projekte.

Riešil som tiež problém prijímania správ od klientských schránok bez obmedzovania ostatných klientov. Pôvodné riešenie používalo funkciu recv(), ktorá spôsobovala blokovanie. Tento problém som vyriešil funkciou nastavením timeoutov, aby sa minimalizovali oneskorenia inputov od hráčov a zabezpečil plynulejší chod servera. Tiež sa potom napravila synchronizácia posielania a prijímania.

Výzvou bolo určite správne ukončovanie vlákien, napríklad na konci aplikácie. Toto som vyriešil evidovaním, ktoré vlákno patrí ku ktorému klientovi. Pre ukončenie všetkých vlákien pri vypnutí servera som použil premennú ‚shutdown', ktorá mení hodnotu z false na true. Táto premenná riadi while podmienku v implementácii, a keď sa jej hodnota zmení po ukončení simulácie, vlákna sa ukončia a spoja.

7. Používateľská príručka

7.1. Spustenie programu

Program sa spúšťa z príkazového riadka. Príkaz na spustenie aplikácie: *./client* Po spustení server začne počúvať na zadanom porte a čakať na pripojenie klientov. Počas programu sú ošetrené vstupy používateľa.

7.2. Hlavné menu

Po pripojení klienta k serveru sa zobrazí hlavné menu, ktoré ponúka nasledujúce možnosti:

- Nová simulácia: Vytvorenie a spustenie novej simulácie.
- Pripojenie k existujúcej simulácii: Pripojenie k už bežiacej simulácii.
- Načítanie simulácie: Načítanie uloženej simulácie zo súboru.
- Ukončenie: Ukončenie programu.
- hodnota 10 pre rýchle spustenie aplikácie v pomalom móde so zadaním predpripravených parametrov simulácie
- hodnota 11 pre rýchle spustenie aplikácie v rýchlom móde so zadaním predpripravených parametrov simulácie

Užívateľ si vyberie jednu z možností zadaním príslušného čísla.

7.3. 3. Nová simulácia

Ak užívateľ zvolí možnosť Nová simulácia, bude vyzvaný k zadaniu konfiguračných parametrov simulácie. Tieto parametre zahŕňajú:

- Mód simulácie: Normálny mód alebo mód úspešnosti.
- Typ sveta: Bez prekážok alebo s prekážkami.
- Šírka a výška mriežky: Rozmery sveta (minimálne 3x3, maximálne 50x50).
- Maximálny počet krokov: Maximálny počet krokov, ktoré môže simulácia vykonať (1 až 1000).
- Počet replikácií: Počet opakovaní pre každé políčko (1 až 1000).
- Pravdepodobnosti pohybu: Pravdepodobnosti pre pohyb smerom hore, dole, vľavo a vpravo (súčet musí byť 1.00).
- Názov súboru: Názov súboru, do ktorého sa uloží výsledok simulácie.

Po zadaní všetkých parametrov sa simulácia spustí. Počas simulácie môže užívateľ meniť mód zobrazenia (vizualizácia, úspešnosť, priemer) alebo ukončiť simuláciu. Ako meniť je napísané v ďalšej časti.

7.4. Klávesové skratky počas simulácie

Počas simulácie môže užívateľ využívať nasledujúce klávesové skratky:

- q: Ukončenie simulácie.
- m: Zmena módu zobrazenia (vizualizácia, úspešnosť, priemer).
- e: Ukončenie spojenia so serverom.

7.5. Pripojenie k existujúcej simulácii

Ak užívateľ zvolí možnosť Pripojenie k existujúcej simulácii, bude pripojený k už bežiacej simulácii na serveri. Užívateľ môže sledovať priebeh simulácie a meniť mód zobrazenia. Táto možnosť je užitočná pre viac užívateľov, ktorí chcú sledovať rovnakú simuláciu súčasne.

7.6. Načítanie simulácie

Možnosť Načítanie simulácie umožňuje užívateľovi načítať uloženú simuláciu zo súboru. Užívateľ musí zadať názov súboru, z ktorého sa má simulácia načítať. Po načítaní sa zobrazia konfiguračné parametre a výsledky simulácie. Následne je možné simuláciu znovu spustiť so zadaním nového počtu replikácii a novým názvom súboru pre uloženie výsledkov.

7.7. Ukladanie a načítavanie simulácií

Simulácie je možné ukladať do textových súborov. Uložené simulácie obsahujú konfiguračné parametre a výsledky simulácie. Načítanie simulácie umožňuje zobraziť tieto výsledky bez potreby opätovného spustenia simulácie.

7.8. Ukončenie servera

Server môže byť ukončený buď priamo z kódu (nastavením premennej shutdown na true), alebo ukončením všetkých klientov. Po ukončení servera sa uvoľnia všetky alokované zdroje a uzavriú sieťové spojenia.

7.9. Ukončenie aplikácie

Klient ukončí aplikáciu zadaním možnosti „Ukončiť aplikáciu“ v hlavnom menu.

8. Záver

Tento projekt implementuje simuláciu náhodných prechádzok (random walk) s možnosťou spustenia v rôznych režimoch, vrátane vizuálneho a štatistického zobrazenia. Aplikácia pozostáva z klienta a servera, kde klient poskytuje užívateľské rozhranie na konfiguráciu a spustenie simulácie, zatiaľ čo server vykonáva samotnú simuláciu a posiela výsledky klientovi. Projekt využíva multithreading na paralelné spracovanie a soketovú komunikáciu pre prenos dát medzi klientom a serverom. Modulárna štruktúra kódu, oddelenie logiky simulácie, vstupno-výstupných operácií a sieťovej komunikácie zaisťuje prehľadnosť a ľahkú rozšíriteľnosť projektu. Tento projekt je písaný v programovacom jazyku C.

9. Zdroje, inšpirácia, čerpanie informácií, použité dokumentácie

Cvičenia a prednášky z predmetu POS

https://stackoverflow.com/questions/448944/c-non-blocking-keyboard-input

https://pubs.opengroup.org/onlinepubs/007908799/xsh/pthread.h.html

https://stackoverflow.com/

https://man7.org/linux/man-pages/