**Obrázok, na ktorom je text

Automaticky generovaný popis**

Meno a priezvisko: Alena Polakovičová

Mail: apolakov@students.zcu.cz

Študijné číslo: A20B0212P

23.01.2024

KIV/UPS

Kameň, papier, nožnice

Obsah

[1 Zadanie 3](#_Toc156971341)

[1.1 Úloha 3](#_Toc156971342)

[Vybraný problém 3](#_Toc156971343)

[2 Programátorská dokumentácia 4](#_Toc156971344)

[2.1 Popis riešenia 4](#_Toc156971345)

[2.1.1 Prijímanie a Spracovanie Klientov 4](#_Toc156971346)

[2.1.2 Spracovanie Hry 4](#_Toc156971347)

[2.1.3 Komunikácia s Klientmi 4](#_Toc156971348)

[2.1.4 Architektúra 4](#_Toc156971349)

[2.1.5 Riešenie Možných Problémov 4](#_Toc156971350)

[2.2 Dátové štruktúry 5](#_Toc156971351)

[2.3 Software použitý pri vývoji 5](#_Toc156971352)

[2.4 Formát správ 5](#_Toc156971353)

[Pripojenie klienta na server: 5](#_Toc156971354)

[Hlavný Herný Cyklus: 5](#_Toc156971355)

[Prijímanie Výsledku Hry od Serveru: 5](#_Toc156971356)

[Rozhodnutie o Pokračovaní v Hre: 6](#_Toc156971357)

[Spracovanie Odpojenia: 6](#_Toc156971358)

[4 Užívateľská dokumentácia 7](#_Toc156971359)

[5 Záver 7](#_Toc156971360)

[6 Použitá literatúra 8](#_Toc156971361)

# 1 Zadanie

## Úloha

Cieľom práce je zostaviť sieťovú aplikáciu, ktorá predstavuje jednoduchú hru. Táto aplikácia by mala byť navrhnutá tak, aby podporovala viacero hráčov hrajúcich cez počítač. Klúčovými aspektmi sú vytvorenie serverovej časti, ktorá bude schopná obsluhovať viacerých klientov - hráčov a zároveň zvládať viaceré hry súčasne, ako aj uchovávať stav každej hry. Klient, teda hráč, by mal mať možnosť pokračovať v hre presne tam, kde skončil, po svojom opätovnom prihlásení.

Dôležitý aspekt predstavuje robustnosť a odolnosť programu voči chybným vstupom a výpadkom serveru či klientov. Študenti by mali aplikáciu programovať v jazykoch C/C++ alebo Java, pričom je potrebné dodržiavať určené požiadavky na komunikáciu a softvérovú architektúru.

Celkovým cieľom je teda nie len vytvorenie funkčnej hry, ale aj demonštrácia schopností v oblasti sieťového programovania, paralelného spracovania, správy chýb a zaznamenávania dôležitých údajov o behu aplikácie.

## Vybraný problém

Hra "kameň, papier, nožnice" je jednoduchá hra, ktorá je známa po celom svete. Je to hra pre dvoch hráčov, pri ktorej každý hráč súčasne formuje jednu z troch tvarov pomocou ruky. Tieto tvary reprezentujú kameň, papier a nožnice. Pravidlá sú jednoduché:

Kameň porazí nožnice (kameň otupí nožnice),

Nožnice porazia papier (nožnice rozstrihnú papier),

Papier porazí kameň (papier obalí kameň).

Hra je často používaná ako zábavný spôsob rozhodovania, podobne ako hodenie mince, alebo ako samostatná zábavná hra. V kultúrach po celom svete sa môže objavovať v rôznych variáciách, s rôznymi pravidlami a názvami, ale základný koncept zostáva rovnaký.

# 2 Programátorská dokumentácia

## 2.1 Popis riešenia

Serverová aplikácia pre hru "kameň, papier, nožnice" je navrhnutá ako viacvláknový server, ktorý zvláda simultánne hry a spracováva viacero klientov súčasne. Tento prístup umožňuje efektívne využitie zdrojov a poskytuje hladký herný zážitok pre užívateľov.

### 2.1.1 Prijímanie a Spracovanie Klientov

Naslúchanie a Pripájanie Nových Klientov: Server nepretržite naslúcha na špecifickom sieťovom porte a akceptuje pripojenia od nových klientov. Každý klient je identifikovaný unikátnym socket ID.

Registrácia a Rekonexia Klientov: Po pripojení klient poskytne svoje meno, ktoré server používa pre identifikáciu a správu hry. Server tiež podporuje rekonexiu klientov v prípade neočakávaného odpojenia.

### 2.1.2 Spracovanie Hry

Výber Protivníka: Po pripojení klienta, server hľadá dostupného protivníka. Hľadá sa klient, ktorý čaká v "loby" - čakacej miestnosti.

Inicializácia Hry: Po nájdení dvoch protihráčov server inicializuje hernú session, pričom obaja hráči dostanú informácie o svojom protivníkovi.

Herný Cyklus: Klienti posielajú svoje výbery (kameň, papier alebo nožnice) serveru, ktorý ich porovná a určí víťaza kola. Server následne poskytne obom hráčom výsledok kola.

### 2.1.3 Komunikácia s Klientmi

Posielanie Stavu Hry: Server komunikuje so svojimi klientmi pomocou správ, ktoré obsahujú informácie o stave hry, ako sú výsledky kôl, aktuálny výber hráča a výber protihráča.

Riadenie Hry: Server riadi tok hry, čaká na výbery od klientov a poskytuje pokyny na ďalšie kroky - napríklad možnosť hrať znova alebo opustiť hru.

### 2.1.4 Architektúra

Vlákna pre Počúvanie Klientov: Pre každého klienta je vytvorené samostatné vlákno, ktoré naslúcha ich vstupom a spracováva ich herné výbery.

Hlavné Vlákno Servera: Hlavné vlákno servera riadi proces prijímania nových klientov a priradenia protivníkov.

Synchronizácia a Bezpečnosť: Pre zabezpečenie korektnej funkcionalitiy a synchronizácie medzi vláknami sú použité mutexy a podmienkové premenné.

### 2.1.5 Riešenie Možných Problémov

V prípade chyby v sieťovom pripojení alebo neočakávaného odpojenia klienta, server dokáže identifikovať takéto prípady a umožní klientom znovu sa pripojiť a pokračovať v hre. Súčasne však kontroluje kolízie. Sú ošetrené všetky chybné vstupy a správy pre klienta aj server.

## Dátové štruktúry

Štruktúra player  
- Reprezentuje hráča, udržuje informácie ako meno, socket ID, výber v hre, informácie o protihráči a stav v hre.

Pole clients  
-Pole štruktúr player, ktoré uchováva informácie o všetkých pripojených klientoch.

## Software použitý pri vývoji

Programovací Jazyk: Server aplikácie je napísaný v programovacom jazyku C a client v programovacom jazyku python.

Vývojové prostredie: Aplikácia bola vyvíjaná v aplikácii Visual Studio, ale môže byť uprvovaná v ľubovoľnom textovom editore alebo IDE, ktoré podporuje Python a C.

Kompilátor: GCC (GNU Compiler Collection) alebo iný kompatibilný kompilátor pre C.

Operačný systém: Server beží v operačnom systéme Linux. Client môže bežať v Linuxe aj na Windowse.

## 2.4 Formát správ

•state:connecting; (server → klient)  
 •Popis: Informácia o pokuse o pripojenie klienta k serveru.  
 •Stav hráča: Pripájanie.  
 •Smer: Server hlási klientovi, že pripojenie začalo.

•name:long; (server → klient)  
 •Popis: Odmietnutie mena kvôli jeho nadmernej dĺžke.  
 •Stav hráča: Pripájanie.  
 •Smer: Server informuje klienta o potrebe skrátenia mena.

•state:lobby; (server → klient)  
 •Popis: Potvrdenie vstupu klienta do lobby pred zahájením hry.  
 •Stav hráča: V lobby, čakanie na hru.  
 •Smer: Server potvrdzuje klientovi, že je v čakacej miestnosti.

•response:ok; (server → klient)  
 •Popis: Potvrdenie správneho príkazu alebo žiadosti klienta.  
 •Stav hráča: Aktívna hra alebo po výbere akcie.  
 •Smer: Server akceptuje a potvrdzuje akciu klienta.

•ping; (klient → server)  
 •Popis: Kontrolný signál od klienta na udržanie alebo overenie spojenia.  
 •Stav hráča: Ako reakcia na serverový ping alebo preventívne.  
 •Smer: Klient odosiela serveru na udržiavanie spojenia.

•state:game ended (server → klient)  
 •Popis: Oznam o konci hry.  
 •Stav hráča: Hra skončila.  
 •Smer: Server informuje klienta, že hra bola ukončená a výsledok bol zaznamenaný.

•state:playing (server → klient)  
 •Popis: Oznam o začatí hry medzi hráčmi.  
 •Stav hráča: Začiatok hry.  
 •Smer: Server informuje klientov, že hra začína a môžu vykonávať svoje tahy.

•state:disconnected (server → klient)  
 •Popis: Upozornenie na odpojenie klienta od serveru.  
 •Stav hráča: Odpojenie.  
 •Smer: Server informuje klienta o odpojení z hry alebo servera.

•error:ConnectionError: Server is busy. Try later; (server → klient)  
 •Popis: Informácia o tom, že server je preťažený alebo nedostupný.  
 •Stav hráča: Pokus o pripojenie.  
 •Smer: Server informuje klienta, že nemôže byť obslúžený kvôli preťaženiu a mal by skúsiť pripojenie neskôr.

•info:timeout; (server → klient)  
 •Popis: Upozornenie pre klienta o vypršanom čase na odpoveď alebo akciu.  
 •Stav hráča: Čakanie na akciu.  
 •Smer: Server informuje klienta, že neodpovedal v stanovenom čase a bude odpojený.

•name:{meno}; (klient → server)  
 •Popis: Klient odosiela svoje meno na server pri pripájaní.  
 •Stav hráča: Pripájanie.  
 •Smer: Klient sa identifikuje serveru svojím menom.

•ping; (server → klient)  
 •Popis: Server odosiela signál "ping" na overenie spojenia s klientom.  
 •Stav hráča: Akýkoľvek, najmä keď je v stave čakania alebo v lobby.  
 •Smer: Udržiavanie aktívneho spojenia a overenie dostupnosti serveru.

•move: {ťah}; (klient → server)

•Ťahy: rock/paper/scissors  
 •Popis: Klient odosiela svoj výber v hre "kameň, papier, nožnice".  
 •Stav hráča: V hre, počas výberu svojho ťahu.  
 •Smer: Klient informuje server o svojom rozhodnutí v hre.

• decision:bye; (klient → server)  
 •Popis: Klient odosiela správu na ukončenie spojenia so serverom.  
 •Stav hráča: Po skončení hry alebo pri odchode z hry.  
 •Smer: Klient iniciuje odpojenie od servera.

• decision:again; (klient → server)  
 •Popis: Klient žiada o ďalšiu hru po skončení predchádzajúcej.  
 •Stav hráča: Po zobrazení výsledkov hry, keď chce pokračovať v ďalšej hre.  
 •Smer: Klient žiada server o restart hry.

•pong; (klient → server)  
 •Popis: Klient potvrdzuje spojenie.  
 •Stav hráča: Akýkoľvek, najmä keď je v stave čakania alebo v lobby.  
 •Smer: Udržiavanie aktívneho spojenia a overenie dostupnosti.

# 4 Užívateľská dokumentácia

Pre správny bez serveru je potrebný Linux. ;

Krok1 – Zostavenie (nepovinný)

Aby sme zabezpečili bezchybný chod programu, môžeme znova zostaviť program. Tento krok by nemal byť nutný, ale na niektorých zariadeniach môže byť vyžadovaný.

make -f makefile

Krok2- Spustenie serveru

Server môže byť spustený ako

./server číslo\_portu

Krok3- Spustenie klienta

Klient môže byť spúštaný na operačnom systému linux aj windows.

Pre Linux je to:

python3 client.py ip\_adresa číslo\_portu

Pre windows je to:

python client.py ip\_adresa číslo\_portu

# 5Záver

# 6 Použitá literatúra

[1] Unknown: Rock-paper-scissors, The Game Gal, 2016