

**Hra: Connect 4**

SEMESTRÁLNA PRÁCA

Vypracoval(a): **Michael Fabo, Miriam Gmitterová**

Študijná skupina: **5ZYI37**

Predmet: **Princípy operačných systémov**

Cvičiaci: **Ing. Peter Sedláček, PhD.**

Obsah

[1. Štruktúra projektu 3](#_Toc155632983)

[1.1. UML server: 4](#_Toc155632984)

[1.2. UML klient: 5](#_Toc155632985)

[2. Použitie socketov a vlákien 6](#_Toc155632986)

[2.1. Sockety 6](#_Toc155632987)

[2.2. Vlákna 6](#_Toc155632988)

[3. Synchronizačný problém 6](#_Toc155632989)

[4. Ďalšie problémy 7](#_Toc155632990)

# Štruktúra projektu

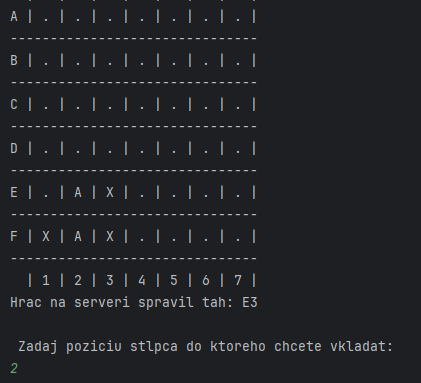
Náš projekt predstavuje sieťovú hru pre dvoch hráčov s názvom Connect 4. Jej podstatou je spojiť 4 znaky na hracej ploche v jednom z 8 smerov. Hráči sa striedajú na ťahoch a ten, ktorému sa to podarí ako prvému vyhráva. Jeden hráč hrá na Serveri a druhý sa pripája ako Klient.

Projekt je rozdelený na 2 projekty, Server a Klient. Server je písaný v programovacom jazyku C a Klient v jazyku C++.

Serverová časť predstavuje hostiteľa hry, má na starosti zapnutie hry, komunikuje s klientom, čítanie vstupov hráča, vykonávanie výherných validácií a vykreslenie hracej plochy.

Klientska časť zabezpečuje pripojenie na server, číta vstupy hráča, vypisuje stav hry.

**Ukážka hry v konzole:**



## UML server:

Obrázok, na ktorom je text, snímka obrazovky, diagram, rovnobežný

Automaticky generovaný popis

## UML klient:

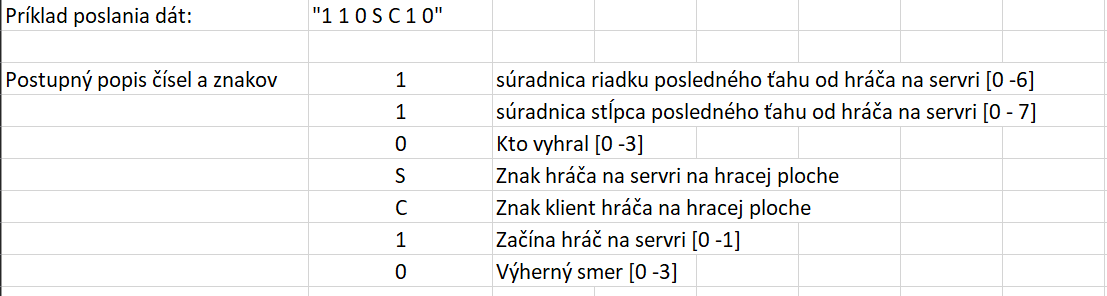
Obrázok, na ktorom je text, snímka obrazovky, rad, diagram

Automaticky generovaný popis

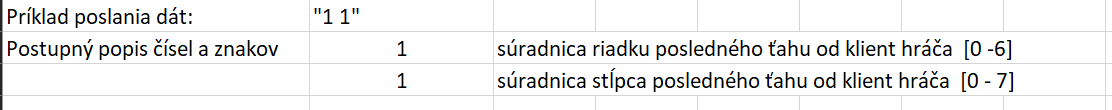
# Použitie socketov a vlákien

## Sockety

**Server:** Socket sme použili pre pripojenie klienta a komunikáciu s klientom pre prijímanie a odosielanie dát.



**Klient:** Socket sme použili na klientovi pri komunikácií so serverom pre prijímanie a odosielanie dát.



## Vlákna

**Server:** Na serveri sme použili 3 vlákna. Jedno pre komunikáciu s klientom, druhé pre komunikáciu s hráčom na serveri a tretie pre prijímanie a odosielanie dát klientovi.

**Klient:** V klientovi sme použili 2 vlákna. Jedno na komunikáciu s hráčom na klientskej časti a druhé pre čítanie dát zo servra.

# Synchronizačný problém

Náš kód pracuje s viacerými vláknami, ktoré môžu súčasne pristupovať ku zdieľaným dátam. Práve pri týchto operáciách na zdieľaných dátach môžu vznikať synchronizačné problémy.

Príklady v kóde, kde môže dôjsť k problémom:

1. data->hra.hraSkoncila: Viacero vlákien môže naraz skončiť hru (data->hra.hraSkoncila = true;).

Zároveň, iné vlákno môže kontrolovať túto podmienku a pokračovať v čítaní alebo zápise do štruktúry hra.

1. data->hra.jeServerHracNaRade: Podobný problém môže nastať pri práci s touto premennou, ktorá signalizuje, ktorý hráč je na rade. Viaceré vlákna môžu súčasne meniť alebo čítať túto hodnotu.
2. Metóda vykresliHraciuPlochu používa na vykreslenie dáta zo štruktúry, používa sa pri vykresľovaní ťahov hráčov v dvoch vláknach.
3. Pri odosielaní dát klientovi vo vlákne consume.
4. Pri spracovaní dát klienta vo vlákne consume.

Tieto problémy sme vyriešili mutexami, podmienkovými premennými a signalizáciami.

Jeden mutex sme použili pre štruktúru thread data a druhý pre štruktúru hra. Pre štruktúru hra sme takisto využili 2 podmienková premenné.

V klientskej časti sa používajú mutexy (data->mutex) a podmienkové premenné (data->jeClientHracNaRade) na riadenie prístupu k zdieľaným údajom medzi dvoma vláknami (produce a consume) pre predídenie synchronizačného problému.

# Ďalšie problémy

V rámci projektu sme sa stretli napríklad s problémom s čítaním dát zo servra a potrebovali sme dorobiť metódu pre získanie dát.

Ďalším problémom bolo riešenie výherných validácií, kde napríklad validácie po diagonálach boli zložitejšie.

Ďalej sme riešili ošetrenie kódu pred zlými vstupmi od používateľa alebo od klienta.