



Fakulta aplikovaných věd

Katedra informatiky a výpočetní techniky

Semestrální práce z KIV/UPS 2011

Jednoduchá síťová hra Ping-pong (protokol UDP)

Jan Moulis
A09B0359P
jmoulis@students.zcu.cz

Zadání

Vytvoření jednoduché síťové hry (server a klient) za použití nespojovaného protokolu *UDP*. Síťová hra bude navržena podle známé klasické hry Ping-pong.

Server bude naprogramován v jazyce *C*. Výstupy serveru budou v alfanumerické podobě do konzole a do logovacího souboru. Server bude řešen pod operačním systémem *Linux*. Server musí být schopen obsluhovat více klientů (hráčů) najednou. Server bude doplněn o zpracování statistických údajů (počet odeslaných zpráv, počet přijatých zpráv, doba běhu serveru, počet připojených hráčů a počet spuštěných her).

Klient bude naprogramován v jazyce *Java*. Uživatelské rozhraní klienta může být reprezentováno konzolí nebo grafickým rozhraním.

Programátorská dokumentace

Popis serveru

Server je naprogramován v programovacím jazyce *C*. Server je tvořen z více jednodušších částí, které se dohromady starají o obsluhu klientů, her a výpisu informací.

Základní rozdělení:

- *connect_players* - Modul pro testování, zda jsou všichni klienti připojeni. V případě, že je klient nedostupný, obsahuje modul funkce pro jeho odpojení ze serveru.
- *game* - Modul reprezentující jednu běžící hru na serveru. Obsahuje vlákno starající se o běh jedné hry. Dále obsahuje funkci pro ukončení všech her v případě vypnutí serveru.
- *info_games* - Modul obsahuje funkci, která zadanému klientovi pošle informace o všech hráčích na serveru.
- *log* - Modul pro ukládání informací o běhu serveru do logovacího souboru (stejný výstup i na konzoli). Obsahuje funkce pro otevření a zavření logovacího souboru a funkce pro zápis dat do tohoto souboru.
- *main* - Hlavní modul programu, který se spouští jako první. Obsahuje funkce pro samotné spuštění serveru a následně spouští další potřebné funkce pro běh serveru.

Při spuštění serveru se v první řadě zaregistrují obslužné funkce pro ukončení programu funkcí *exit()* (funkce *exit_procedure()*) a pro ukončení systémovým signálem *SIGINT* (*CTRL+C*) (funkce *signal_callback_handler()*). Následně program otevře logovací soubor a uloží si čas, kdy byl server spuštěn (statistické údaje).

Dále jsou vstupní parametry programu (*-h* nápověda, *-i* IP adresa, *-p* port) zvalidovány, zda nejsou zadána špatná vstupní data. V případě chyby dochází k ukončení programu s chybovým výpisem.

Pokud byly vstupní parametry programu úspěšně zvalidovány, pokusí se server nabindovat na zadanou IP adresu a port. V případě chyby (nelze použít takovou IP adresu nebo je zadaný port obsazen, popřípadě číslo portu neodpovídá použitelnému rozsahu) skončí program s chybovým výpisem. V opačném případě program spustí vlákno pro testování připojení klientů (*try_connect_players*) a začne naslouchat na dané IP adrese a portu.

Poslouchání na IP adrese a portu probíhá v cyklu, kde na začátku se čeká na přijetí zprávy od klienta. Pokud tato zpráva odpovídá formátu, který používá server, tj. zpráva začíná znakem '#', otestuje se, jaký druh zprávy (požadavek) server přijal a provedou se požadované operace.

Druhy požadavků:

- *c* - odpověď klienta na dotaz serveru, zda je stále připojen
- *g* - klient požaduje zaslání informací o hrách na serveru
- *n* - nový klient se připojuje na server
- *s* - klient požaduje založení hry na serveru
- *j* - klient se chce připojit k založené hře
- *q* - klient ukončil běh svého programu, odpojení hráče ze serveru

Požadavek *c* server vyřizuje jen nastavením proměnné *connected* na hodnotu *1* u daného hráče. Hodnota *1* znamená, že je klient připojený a komunikace probíhá v pořádku.

Požadavek *g* na serveru způsobí spuštění nového vlákna, které uživateli, který si zažádal odešle informace o všech hrách na serveru.

Požadavek *n* uloží nového klienta na server, pokud je na serveru pro klienta ještě místo. Pokud proběhne uložení klienta v pořádku, dojde k odeslání zprávy s identifikačním číslem pro klienta. Pod tímto identifikačním číslem následně klient komunikuje se serverem.

Požadavek *s* nejprve na serveru spustí kontrolu, zda není uživatelem zadaná hra již založená (než se hráč rozhodne, kterou hru chce založit a na jaké bude hrát straně, může jiný hráč jím zvolenou hru založit dříve). V takovém případě dojde k odeslání chybové zprávy uživateli. V opačném případě je vytvořeno vlákno pro danou hru a hráč, který hru založil je do této hry přidán na stranu, kterou si sám vybral. Hráči je také odeslán nový port, přes který bude probíhat komunikace v průběhu hry.

Požadavek *j* nejprve na serveru spustí kontrolu, zda hra do které se chce hráč připojit je založena a zda je v ní ještě místo (hra může být ukončena, kvůli nedostupnosti hráče, který ji založil nebo se jiný hráč připojil do hry dříve). Pokud je hra založena a je v ní místo, dojde k připojení hráče k této hře a k odeslání hráči čísla nového portu, přes který se bude komunikovat. Následně je spuštěno vlákno hry, které se začne starat o komunikaci v průběhu hry.

Požadavek *q* server vyřizuje jen nastavením proměnné *player_status* na hodnotu *-1* u daného hráče. O odpojení hráče ze serveru se již dále nestará. To probíhá v odděleném vlákne *try_connect_players*.

Po spuštění vlákna jedné hry si nejprve vlákno uloží informace o hře, která se bude hrát a poté odešle hráčům jména jejich protivníků. Dále se klientům odešle zpráva o spuštění hry a začíná samotná hra.

Hra na serveru vypadá tak, že server přijímá zprávy od hráče, který hru založil, o pozici míčku, o skóre a dále od obou hráčů pozice pálek. Tyto zprávy server jen přeposílá mezi hráči. Dále server přijímá požadavek o znovuzaslání jména protivníka a v neposlední řadě zprávu o vítězství jednoho z hráčů.

Pokud je jeden z hráčů nedostupný, dojde k pozastavení hry dokud není hráč opět dostupný. V případě ukončení hry vítězstvím jednoho hráče, dojde k odpojení hráčů z této hry a její ukončení (ukončení vlákna).

Odeslání informací o všech hrách klientovi probíhá složením informací o hře do jednotného formátu a odeslání jedné takové zprávy pro každou hru. Když jsou odeslány informace o všech hrách je odeslána zpráva klientovi, že již byly poslány všechny informace.

Testování zda jsou klienti stále připojeni a dostupní probíhá ve dvou fázích. Zaprvé se všem klientům odešle zpráva zda jsou připojeni a pokud byl klient v předchozím testování připojen, nastaví se jeho proměnná *connected* na 0.

Poté se testovací vlákno uspí na 2 sekundy. Po uplynutí této doby vlákno projde všechny klienty a podívá se, zda odpověděli a jsou připojeni. Pokud je klient připojen a odpověděl má jeho proměnná *connected* hodnotu 1. Pokud proměnná *connected* má jinou hodnotu dojde k jejímu zmenšení o 1. Když je tato hodnota menší -5, což znamená, že klient se neohlásil přibližně 12 sekund dojde k jeho odpojení od serveru. Pokud navíc hráč hrál nebo založil nějakou hru, dojde k zrušení této hry.

Při ukončení běhu serveru kombinací kláves *CTRL+C* program ukončí všechny hry a informuje hráče o vypnutí serveru. Poté vypíše statistické údaje o běhu (počet odeslaných zpráv, počet přijatých zpráv, doba běhu serveru, počet připojených hráčů a počet spuštěných her).

Popis klienta

Klient je naprogramován v programovacím jazyce *Java*. Klient je stejně jako server rozdělen na více samostatných částí.

Základní rozdělení:

- *Ball* - třída reprezentující míček ve hře
- *Bat* - třída reprezentující jednu pátku ve hře
- *Comunication* - třída, která se stará o komunikaci mezi klientem a serverem
- *ComunicationRecieve* - třída, která se stará o příjem dat ze serveru
- *ComunicationSend* - třída, která se stará o odesílání dat na server
- *GamePanel* - třída reprezentující herní plochu
- *InfoGamesPanel* - třída pro výpis běžících her na serveru
- *Main* - hlavní třída programu
- *ServerConnection* - třída pro připojení klienta k serveru
- *Shape* - třída reprezentující tvar míčku a pálek
- *Window* - třída reprezentující hlavní okno programu

Po spuštění programu klienta dojde k vykreslení hlavního okna programu. Pokud se chce hráč připojit na server a zvolí z menu danou položku dojde v hlavním okně ke změně prostřední komponenty na panel připojení na server. Zde po zadání jména hráče, IP adresy a portu serveru, na který se chce připojit, dojde k zvalidování těchto dat a pokud jsou zadána správně, pokusí se program spojit se serverem a zahájit komunikaci.

Aby mohl klient se serverem komunikovat, vytvoří se nová instance *Communication* a v ní se vytvoří nová komunikace pro odesílání dat a vlákno pro příjem dat. Při vytváření komunikace dochází k odeslání informací na server a čeká se na příjem zprávy od serveru, zda se připojení podařilo, či je server plný. Pokud server v daném čase neodpoví, klient zobrazí uživateli chybovou zprávu o nedostupnosti serveru a nabídne hráči možnosti, co může dál dělat.

Po úspěšném připojení na server se hráči zobrazí tabulka s informacemi o všech hrách na serveru a ovládací tlačítka pro aktualizaci informací v tabulce, založení hry a připojení do hry.

Zmáčknutím tlačítka pro aktualizaci informací o hrách dojde k odeslání požadavku na server o zaslání nových informací o hrách a aktualizaci dat v tabulce.

Vybráním hry a zmáčknutím tlačítka pro založení hry klient odešle serveru žádost o založení dané hry, ještě před tím je ale hráč dotázán na stranu, na které chce hrát. Server klientovi odpoví buď kladně zasláním nového portu, na kterém klient v průběhu hry bude komunikovat se serverem, nebo zasláním chybové zprávy, že hráčem požadovaná hra je již založena.

Vybráním hry a zmáčknutím tlačítka pro připojení ke hře klient odešle serveru žádost o připojení do dané hry. Server klientovi odpoví buď kladně zasláním nového portu, na kterém klient v průběhu hry bude komunikovat se serverem, nebo zasláním chybové zprávy, pokud hra, kterou hráč vybral, není ještě založená a nebo je již požadovaná hra plná.

Po založení/připojení do hry se zobrazí hráči hrací plocha hry *Ping-pong*. Dokud nejsou ve hře přítomni oba dva hráči je hrací plocha neaktivní. Po připojení druhého hráče do hry se hra spustí a zakládající klient začne počítat pohyb míčku v závislosti na rozměrech hrací plochy a pozici pálek. Souřadnice míčku a pozice pálek klient odesílá na server. Druhý klient (který se připojil do založené hry) odesílá jen pozici své pálek. Zakládající klient také posílá skóre hry. Hra končí dosažením 5 bodů u jednoho hráče. Tím je ukončen běh hrací plochy a dojde k přepnutí na výpis informací o hrách na serveru.

V případě vypnutí serveru nebo jeho nedostupnosti je hráč informován vyskakovacím oknem a poté dojde k přepnutí na panel připojení k serveru. Pokud naopak hráč ukončí program, je informován server o odpojení klienta.

Význam a formát přijatých/odeslaných zpráv

Mezery ve zprávách jen kvůli čitelnosti.

- $s \rightarrow k$ směr zprávy server klient
- $k \rightarrow s$ směr zprávy klient server
- Zpráva pro klienta, aby se ohlásil, zda je připojen
 $s \rightarrow k$ `#c`
- Trvalé odpojení jednoho hráče ze hry
 $s \rightarrow k$ `#ee`
- Ukončení běhu serveru
 $s \rightarrow k$ `#eq`
- Odpověď klienta, že je stále připojen
 $k \rightarrow s$ `#c id_hrace`
- Žádost klienta o zaslání informací o hrách
 $k \rightarrow s$ `#g id_hrace`
- Informace o jedné hře
 $s \rightarrow k$ `#g id_hry status_hry volné_místo jméno_levého_hráče;jméno_pravého_hráče`
- Byly poslány všechny informace o hrách
 $s \rightarrow k$ `#ga`
- Připojení nového klienta na server
 $k \rightarrow s$ `#n jméno_hráče`
- Odeslání id nově připojeného hráče
 $s \rightarrow k$ `#i id_hráče`
- Server je plný, odmítnutí hráče
 $s \rightarrow k$ `#f`
- Žádost klienta o založení nové hry
 $k \rightarrow s$ `#s id_hráče id_hry strana_hráče`
- Chybová odpověď na založení hry, hra je již založená
 $s \rightarrow k$ `#es`
- Úspěšné založení nové hry, změna portu pro hráče
 $s \rightarrow k$ `#p číslo_portu strana_hráče`
- Žádost hráče o připojení ke hře
 $k \rightarrow s$ `#j id_hráče id_hry`
- Chybová odpověď na připojení ke hře, hra není založená
 $s \rightarrow k$ `#ejs`

- Chybová odpověď na připojení ke hře, hra je plná
s -> k #ejf
- Zpráva o ukončení klientského programu
k -> s #q id_hráče
- Odeslání protihráčova jméno
s -> k #o jméno_protihráče
- Stejně jména hráčů
s -> k #en
- Spuštění/znovuspuštění hry
s -> k #s
- Jeden z hráčů nedostupný
s -> k #eo
- Pozice míčku
k -> s s -> k #b x_souřadnice y_souřadnice
- Skóre hry
k -> s s -> k #s body_levého_hráče body_pravého_hráče
- Výhra jednoho z hráčů
k -> s s -> k #w id_hráče vítězná_strana
- Pozice levé pálky
k -> s s -> k #l pozice
- Pozice pravé pálky
k -> s s -> k #r pozice
- Žádost hráče o znovuzaslání jména protihráče
k -> s #n id_hráče

Uživatelská dokumentace

Překlad serveru

Překlad programu serveru musí probíhat pod operačním systémem *Linux*. Překlad do spustitelného souboru se provádí pomocí překladače *gcc*.

Překlad se může provést dvojím způsobem. Ručním překladem programu a nebo použitím souboru *Makefile*, který provede sestavení a překlad programu sám.

Při ručním překladu je potřeba spustit terminál a přepnout se do adresáře se zdrojovými kódy. Poté se do konzole zadá příkaz:

```
gcc main.c log.c info_games.c game.c connecte_players.c -W -g -lpthread -o nazev_programu
```

výsledkem tohoto příkazu je spustitelný soubor *nazev_programu*.

Lepším řešením je použití souboru *Makefile*. Opět je nutné spustit terminál a přepnout se do adresáře, kde se nachází zdrojové soubory a soubor *Makefile*. Poté stačí zadat příkaz *make* a vytvoří se nám spustitelný soubor *ping_pong_server*. Pokud po zadání příkazu *make* obdržíme hlášku, že je spustitelný soubor aktuální a přesto chceme zkompileovat zdrojové soubory, musíme nejprve zadat příkaz *make clean* a poté příkaz *make*.

Překlad klienta

Překlad klienta probíhá pomocí buildovacího programu *Ant* (<http://ant.apache.org/>). Pro úspěšný překlad programu pomocí buildovacího skriptu *build.xml* je nutné mít nainstalován program *Ant*.

Pro překlad se stačí v konzoli přepnout do adresáře, kde je buildovací soubor *build.xml* umístěn a spustit příkaz *ant distjar*. Po vypsání informací o překladu a kompletování dostáváme hlášku *BUILD SUCCESSFUL* o úspěšném dokončení překladu. Výsledný spustitelný soubor je umístěn ve složce *jar* a jeho jméno je *ping_pong.jar*.

Spuštění serveru

Spuštění serveru musí probíhat pod operačním systémem *Linux* v příkazové řádce. Opět je nutné se stejně jako u překladu v terminál přepnout do složky, ve které je nyní umístěn přeložený spustitelný soubor *ping_pong_server*. Server je nutné spustit s parametrem *-i* pro nastavení *IP adresy* a parametrem *-p* pro nastavení *portu*. Pro spuštění programu se zadá příkaz:

```
./ping_pong_server -i IP_adresa -p port
```

v případě chybného zadání parametrů program vypíše chybovou hlášku o neúspěšném spuštění. Pro nápovědu programu zadejte příkaz:

```
./ping_pong_server -h
```

Informace o běhu programu jsou ukládány do logovacího souboru *log_file.log*.

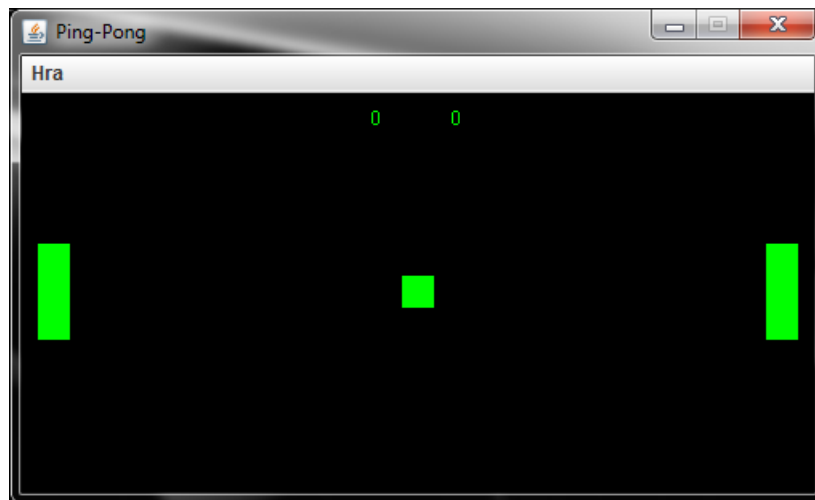
Spuštění klienta

Spuštění programu klienta může proběhnout dvěma způsoby. Buď dvojklikem na soubor *ping_pong.jar*, který je umístěn ve složce *jar*, nebo spuštěním příkazu *java -jar ping_pong.jar* v konzoli. Pokud chce uživatel program spustit z konzole, musí se uživatel přepnout do složky *jar* a poté zadat příkaz.

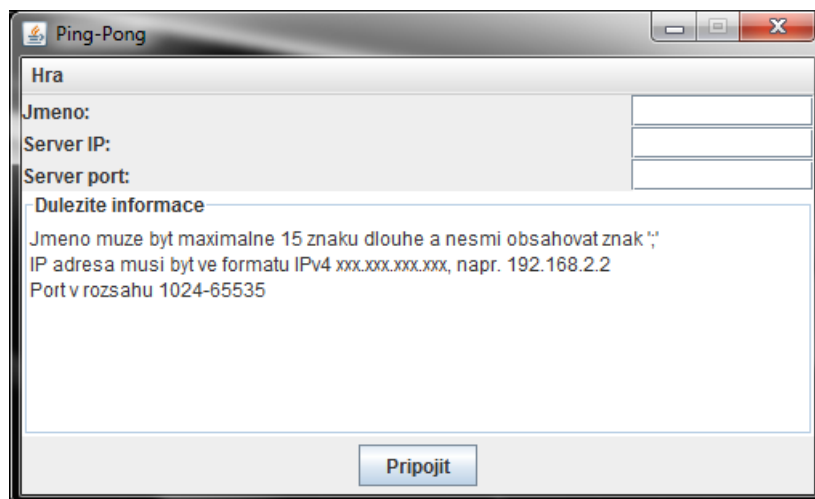
Pokud se po dvojkliku spustí jiný program než *Ping-pong* (např. komprimační program) znamená to, že přípona *.jar* není asociována s běhovým prostředím Javy ale s komprimačním programem a ten ji má otevírat. Řešením tohoto problému může být spuštění programu přes konzoli, nebo kliknutím pravého tlačítka myši na soubor *ping_pong.jar* a vybraní otevření tohoto programu v běhovém prostředí Java.

Ovládání klienta

Po spuštění programu klienta se uživateli zobrazí následující hlavní okno programu.



Uživatel si může v menu vybrat akci, kterou chce provést. Po vybrání připojení na server se uživateli změní okno programu a bude dotázán na zadání svého jména, IP adresy a čísla portu serveru.



Po úspěšném připojení se uživateli zobrazí výpis informací o všech hrách na serveru.

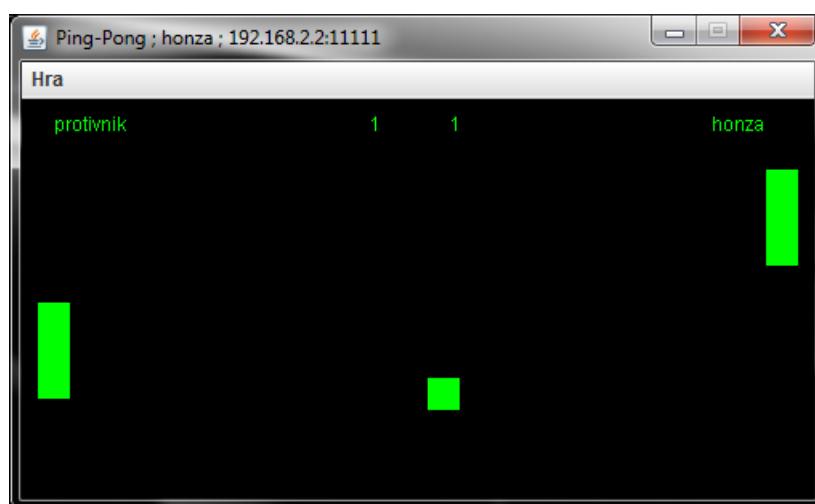


The screenshot shows a window titled "Ping-Pong ; honza ; 192.168.2.2:11111". It contains a table with the following data:

ID hry	ID status	Status	Levy hrac	Pravy hrac
0	0	hra nezalozena	volne misto	volne misto
1	0	hra nezalozena	volne misto	volne misto
2	0	hra nezalozena	volne misto	volne misto
3	0	hra nezalozena	volne misto	volne misto
4	0	hra nezalozena	volne misto	volne misto
5	0	hra nezalozena	volne misto	volne misto
6	0	hra nezalozena	volne misto	volne misto
7	0	hra nezalozena	volne misto	volne misto
8	0	hra nezalozena	volne misto	volne misto

Below the table are three buttons: "Aktualizovat", "Zalozeni hry", and "Pripojeni ke hre".

Po výběru hry a založení/připojení do hry se uživateli zobrazí hrací plocha s míčkem a dvěma páčkami. Hráč ovládá svojí páčku pohyby myši uvnitř hrací plochy (okna).



Závěr

Program splňuje zadání v celém rozsahu. Jako komunikační protokol je zde použit nespojovaný protokol UDP. Server je naprogramován v programovacím jazyce *C*, klient je naprogramován v programovacím jazyce *Java*. Běh serveru je řešen pod operačním systémem *Linux* (testováno v distribucích *Debian* a *Ubuntu*). Klient byl vyvíjen pod operačním systémem *Windows 7*, ale jeho funkčnost je i pod systémem *Linux* (testováno v distribucích *Debian* a *Ubuntu*).

Server je schopen obsluhovat více klientů najednou (použití vláken). Stav serveru (připojení nového hráče, odpojení hráče, založení a ukončení hry, chybové výpisy a statistické údaje) je vypisován do konzole a navíc do logovacího souboru, pro snadné zjištění probíhajících operací v programu.

Klientský program je rozšířen o grafické uživatelské rozhraní, bez kterého by se hra Ping-pong neobešla.

Možným rozšířením programu by mohla být úprava serveru ze stávajícího stavu, kdy počet hráčů resp. počet her je omezen na velikost pole pro hráče resp. hry. Dalším rozšířením by mohlo být vylepšení počítání pohybu míčku v závislosti na rychlosti a směru páčky při odrazu míčku.