

Úloha č. 1 - TCP Server a Client (zadaní pro zimní semestr 2010)

Situace

Robot se pohybuje po městě. Město tvoří 2-D plocha rozdělená na pole o souřadnicích (x,y). Ve městě nejsou žádné překážky. Robot umí provést krok vpřed, otočit se doleva a zvednout značku. Poskytuje také informaci o své pozici ve městě.

Robot se může pohybovat pouze ve městě, mimo město havaruje. Město má rozměr 35 x 35 polí, souřadnice x a y mohou nabývat hodnot z intervalu <-17,17>. Pro svůj pohyb robot potřebuje elektrickou energii, kterou dodává baterie. Každý pohyb robota trochu energie spotřebuje. Aby se robot mohl dále pohybovat po městě je potřeba baterii občas nabít. Baterii lze nabíjet kdykoliv pomocí speciálního příkazu.

Na pozici (0,0) je značka. Úkol robota ke dojít je značce, zvednout ji a přečíst tajemství.

Obecné schema komunikace

Server představuje robota, kterého ovládá klient. Využívá se komunikačního protokolu TCP, server naslouchá na portu, který je zvolen v intervalu 3500 až 3800 (včetně). Protokol je textově orientovaný a lze jej vyzkoušet např. pomocí příkazu `telnet adresa_serveru 3555` (v případě portu 3555).

Po navázání spojení klientem vygeneruje server náhodně souřadnice robota a jeho orientaci. Souřadnice robota nechť jsou zvoleny v intervalu <-16,16> pro x i y. Klient si zjistí souřadnice robota. Počáteční stav baterie je 100%. Protože klient nemá možnost přímo zjistit pozici robota a jeho orientaci, musí provést např. otočení a 1 krok a porovnat nové souřadnice robota (z odpovědi s kódem 250). Poté dovede robota pomocí příkazů ke značce. Značku zvedne. Při úspěšném zvednutí server prozradí text pod značkou a ukončí spojení. Pokud na daném místě značka není, server ohlásí chybu a také ukončí spojení (klient má pouze jeden pokus na zvednutí značky).

Bezprostředně po navázání spojení server odešle text se svojí identifikací zakončený znaky `\r\n` (konec řádky) (viz seznam odpovědí, kód 220). Dále již pouze čeká na příkazy klienta a posílá odpovědi na ně.

Struktura dat

Příkazy

Příkazy posílá klient serveru.

Struktura: oslovení + mezera + jméno příkazu zakončené znaky `\r\n`

Příklad: **Magdaleno KROK\r\n**

Odpovědi

Odpověď se posílá na každý příkaz. Struktura: *kód* <mezera> řetězec odpovědi zakončený znaky `\r\n`

kde *kód* je číslo (textová reprezentace). Možné odpovědi:

odpověď	vysvětlení
220 Text pozdravu robota	Počáteční identifikace posílaná serverem. Posílá server ihned po navázání spojení.
221 USPECH Text tajemství	Úspěšné zvednutí značky. Zároveň přečte tajemství značky (řetězec znaků).
250 OK (stav baterie,x,y)	Poslední operace byla úspěšná. Hodnota <i>stav baterie</i> reprezentuje stav baterie po provedení posledního kroku (textová reprezentace). Hodnoty <i>x</i> a <i>y</i> jsou nové souřadnice robota (textová reprezentace).
500 NEZNAMY PRIKAZ	Zadaný příkaz nebyl rozpoznán. Nebyla provedena žádná akce.
530 HAVARIE	Robot vyšel mimo město a pravděpodobně havaroval.
540 BATERIE PRAZDNA	Úplné vybití baterie, spojení s robotem je přerušeno a robot je ztracen.
550 NELZE ZVEDNOUT ZNACKU	Na zadané pozici se nenachází značka.
570 PORUCHA BLOK x	Robot má poruchu v bloku <i>x</i> (číslo od 1 do 9, textová reprezentace).
571 NENI PORUCHA	Pokus o opravu bloku, který nemá poruchu.
572 ROBOT SE ROZPADL	Robot byl porouchaný a přesto měl provést krok.

Po odeslání odpovědi s kódem 221,530,540,550,571 a 572 se ihned ukončí spojení ze strany serveru. Po odeslání odpovědi s kódem 500 nebo 570 spojení zůstává navázané.

V textu pozdravu robota (odpověď 220) je třeba nalézt oslovení, abychom mohli posílat robotovi příkazy. Oslovení se může ukrývat kdekoliv v textu pozdravu a lze ho nalézt jako podřetězec s formátem: „**Oslovuj mne JMÉNO.**“ Jméno je řetězec libovolných znaků kromě `\r`, `\n`, `\0` a tečky. Mezera nesmí být na začátku nebo na konci jména, uprostřed však ano. Jméno může být dlouhé až 512 bytů, minimálně však 1 byte. Na velikosti písmen ve jméně záleží. Není podporováno žádné kódování češtiny. Pokud použijete kódování češtiny, bude vše pravděpodobně chodit. V případě UTF může být problém, pokud druhý byte znaku obsahuje `\r`,`\n`,mezeru nebo tečku.

V případě serveru není povoleno rekurzivní vnořování konstrukce „Oslovuj mne Oslovuj mne Oslovuj mne Robote...“. Není též povoleno ani takové rekurzivní vnoření, kdy jedna z variant není platná, neboť přesahuje maximální povolenou délku. Příklady:

- `Oslovuj _ mne _ Honzo.` ⇒ *Honzo*
- `Oslovuj _ mne _ Oslovuj _ mne _ Honzo..` ⇒ *nepovolená konstrukce (rekurze), klient může chápat oslovení různě*
- `Oslovuj _ mne _ .Oslovuj _ mne _ Honzo.` ⇒ *Honzo* (povolená konstrukce, není rekurze, první jméno není platné, protože je prázdné, je to tedy součást textu pozdravu)
- `Oslovuj _ mne _ _ Oslovuj _ mne _ Honzo.` ⇒ *Honzo* (povolená konstrukce, není rekurze, první jméno není platné, protože začíná mezerou, je to tedy součást textu pozdravu)
- `Oslovuj _ mne _ Jene _ .Oslovuj _ mne _ Honzo.` ⇒ *Honzo* (povolená konstrukce, není rekurze, první jméno není platné, protože končí mezerou, je to tedy součást textu pozdravu)
- `Oslovuj _ mne _ Jene. Oslovuj _ mne _ Honzo.` ⇒ *nepovolená konstrukce, víc než jedno oslovení splňující syntaxi*

Robot může mít poruchu a nesplnit příkaz KROK (odpověď 570). V tomto případě je třeba robota opravit příkazem OPRAVIT a poté zavolat původní příkaz znovu. Opravit lze pouze blok v poruše, jinak robot ukončí spojení se zprávou 571. Pokud je robot porouchaný a je volán příkaz KROK, bude spojení ukončeno se zprávou 572. Na začátku není robot porouchaný. Robot nevydrží větší

množství kroků a po desátém kroku se už určitě porouchá. Může se však porouchat i dříve.

Seznam příkazů

Příkaz KROK

Syntaxe: oslovení KROK\r\n

Energetická náročnost: 10%

Význam: Provede krok robota směrem, kterým je robot orientován. Po provedení tohoto příkazu se souřadnice x nebo y zvýší nebo sníží o jedničku.

Oslovení je třeba zjistit z odpovědi 220.

Robot nesmí být porouchaný, jinak se rozpadne (zpráva 572). Pokud není porouchaný, může se porouchat (zpráva 570).

Kódy možných odpovědí: 250,530,540,570,572

Příkaz VLEVO

Syntaxe: oslovení VLEVO\r\n

Energetická náročnost: 10%

Význam: Změní orientaci robota. Robot se otočí o 90 stupňů doleva.

Oslovení je třeba zjistit z odpovědi 220.

Kódy možných odpovědí: 250,540

Příkaz ZVEDNI

Syntaxe: oslovení ZVEDNI\r\n

Energetická náročnost: 0%

Význam: Zvedne značku. Pokud není robot na souřadnici (0,0), dojde k chybovému hlášení serveru. Po zavolání tohoto příkazu server vždy ukončí spojení.

Oslovení je třeba zjistit z odpovědi 220.

Kódy možných odpovědí: 221,550

Příkaz OPRAVIT

Syntaxe: oslovení OPRAVIT n \r\n kde n je číslo od 1 do 9 a označuje číslo bloku s poruchou.

Energetická náročnost: 0%

Význam: Opraví zadaný blok robota. Opravit lze pouze blok, který byl odpovědí 570 označen jako porouchaný. Pokud je robot porouchaný a nebude opraven, nelze provést další příkaz KROK. Ostatní příkazy jsou vykonány.

Oslovení je třeba zjistit z odpovědi 220.

Kódy možných odpovědí: 250,571

Příkaz NABIT

Syntaxe: osloveni NABIT\r\n

Energetická náročnost: 0%

Význam: Nabije baterii robota na 100 procent. Nabíjení je ale náročná činnost, při níž se může robot rozbít. Pokud nabíjení proběhne v pořádku, server odešle zprávu 250 se stavem baterie nastaveným na 100% a zopakuje aktuální pozici. Pokud se robot při nabíjení poškodí, stav baterie klesne na 1% a server odešle zprávu 570 s indikací rozbitého bloku. Rozbitý blok je potřeba nejprve opravit a pak robota nabít. Pokus o nabití rozbitého robota vede k nenávratnému zničení robota (zpráva serveru 572).

Kód možných odpovědí: 250, 570, 572

Jiné příkazy robot nezná. Na chybně zadané příkazy server reaguje odpovědí s kódem 500.

Příklad komunikace

tučně = server (robot)

normálně = klient

...klient naváže spojení...

220 Ahoj, tady je robot. Oslovuj mne Ty pekelný stroji.

Ty pekelný stroji VLEVO

250 OK (90,-3,4)

Ty pekelný stroji KROK

250 OK (80,-3,5)

Ty pekelný stroji VLEVO

250 OK (70,-3,5)

Ty pekelný stroji VLEVO

250 OK (60,-3,5)

Ty pekelný stroji KROK

250 OK (50,-3,4)

Ty pekelný stroji KROK

570 PORUCHA BLOK 3

Ty pekelný stroji OPRAVIT 3

250 OK (50,-3,4)

Ty pekelný stroji KROK

250 OK (40,-3,3)

Ty pekelný stroji KROK

250 OK (30,-3,2)

Ty pekelný stroji KROK

250 OK (20,-3,1)

Ty pekelný stroji KROK

250 OK (10,-3,0)

Ty pekelný stroji NABIT

570 PORUCHA BLOK 5

Ty pekelný stroji OPRAVIT 5

250 OK (1,-3,0)

Ty pekelný stroji NABIT

250 OK (100,-3,0)

Ty pekelný stroji VLEVO

250 OK (90,-3,0)

Ty pekelný stroji KROK

250 OK (80,-2,0)

Ty pekelný stroji KROK

250 OK (70,-1,0)

Ty pekelný stroji KROK

250 OK (60,0,0)

Ty pekelný stroji ZVEDNI

221 USPECH Kobyla ma maly bok.

...server uzavřel spojení...

...klient uzavřel spojení...

Požadavky

- programuje se jak klientská, tak serverová část,
- klient umí vyzvednout tajemství zcela automaticky bez zásahu člověka (během celého spojení),
- klientská i serverová část musí 100% fungovat s protistranou na serveru

dsnlab1.felk.cvut.cz (viz níže),

- server musí běžet na portu v intervalu <3500,3800> - zvolte si sami jeho hodnotu, pozor na kolize s ostatními studenty,
- klient má možnost navázat se stejným serverem více spojení paralelně i postupně za sebou (bez restartu serveru),
- server i klient musí být stabilní a odolní proti neočekávaným vstupům,
- programy by měly vypisovat alespoň základní informace o svém stavu (např. posílané a přijímané příkazy či odpovědi),
- IP adresa nebo DNS jméno serveru se zadává jako parametr v klientu do konzole či do GUI (tzn. není zadrátovaná ve zdrojového kódu),
- klient viditelně vypíše v případě úspěšného zvednutí značky její tajemství,
- všechny parametry se zadávají na příkazovou řádku za spouštěný příkaz,
- server musí vypisovat informace o aktuálně obslouženém klientovi (aby bylo možno říct, komu byla odpověď odeslána),
- zdrojový text musí být komentovaný, v hlavičce všech zdrojových textů musí být uveden autor,
- povolené programovací jazyky jsou následující: Java, C, C++.
- při implementaci pište klientskou i serverovou část do 2 zdrojových souborů (jeden pro klienta, jeden pro server). Tyto soubory pojmenujte klient.java (.c,.cpp) a server.java (.c,.cpp). Tyto soubory musí být přeložitelné řádkovým překladačem (javac/gcc/g++). Odevzdávají se pouze tyto dva zdrojové kódy, neimplementujte tedy nic do dalších souborů mimo tyto dva.

Implementační poznámky

- **nepoužívejte pro komunikaci metodu `PrintStream.println(příkaz)`. Zjednodušeně udělá ekvivalent následujícího: `print(příkaz + System.getProperty(„line.separator“))`, kde `line.separator` je na různých platformách různý.**
- Používejte metody, které znáte a víte přesně, co dělají. `BufferedReader.readLine()` se pro server opravdu nehodí, viz Javadoc - `BufferedReader.readLine()` [[http://java.sun.com/javase/6/docs/api/java/io/BufferedReader.html#readLine\(\)](http://java.sun.com/javase/6/docs/api/java/io/BufferedReader.html#readLine())]

Originalita

- Vaše vypracovaná úloha musí být 100% originál,
- je povoleno využít pouze veřejně dostupné zdroje výukové a dokumentační povahy - manuálové stránky, vzorová použití vybrané funkce na Internetu, příklad v knize apod.),
- není povoleno využít jakkoli velkou část zdrojového textu od jiného autora,
- v případě nejasnosti se na možnost využití zdroje informujte u cvičícího,
- veškeré použité zdroje je třeba zmínit v hlavičce zdrojového textu,
- v případě porušení libovolného z těchto pravidel bude hříšník odeslán k disciplinární komisi (neudělení zápočtu je samozřejmostí).

Testování

Release candidate :) verze serveru se konečně přesunula na server dsnlab1.fel.cvut.cz.

Referenční server opět běží na portu 3555. Snad by konečně měl fungovat. Pokud se opět nepůjde, dejte mi vědět. Otestovat jej stále můžete s pomocí příkazu `telnet dsnlab1.felk.cvut.cz 3555`.

Testovací klient implementující nový protokol je dostupný na webové adrese <http://dsnlab1.felk.cvut.cz/karel.php> [<http://dsnlab1.felk.cvut.cz/karel.php>]. Výsledkem běhu klienta je webová stránka a na jejím konci je tabulka s výsledky jednotlivých testů. Klient musí nejprve dokončit celý běh, proto mějte trpělivost při čekání na výsledky. Testování referenčního serveru, který běží na stejném stroji trvá 10-15 sec. Testování vzdáleného serveru bude trvat přiměřeně déle. Pro představu, jak má vypadat výstup do pole **host** zadejte `localhost` a do pole **port** zadejte `3555`.

Termín a způsob odevzdání

Úlohu je třeba odevzdat do 7. (lichá skupina) nebo 8. (sudá skupina) týdne semestru. Za každý týden prodlení je penalizace -0,5 bodu (jako přelom týdne se počítá konec cvičení). Úlohu není možné odevzdat po zápočtovém týdnu.

Pokud úlohu odevzdáte do 5./7. týdne včetně, hrozí vám body navíc za aktivitu. Takto lze odevzdat pouze 100% funkční úlohu.

Pozor: Každé cvičení máte možnost využít JEDEN pokus na odevzdání této úlohy. Důkladně proto svoji úlohu předem otestujte. Úlohu můžete odevzdat pouze na tom cvičení, na kterém jste zapsáni.

Odevzdat lze pouze takovou úlohu, která je nahraná na server service.felk.cvut.cz. Upload se provede takto:

1. zabalte zdrojový kód klienta i serveru do jednoho souboru (.zip) - tento archiv bude tedy obsahovat právě 2 soubory
2. archiv pojmenujte takto: **uloha1-_váš_login_.zip**, např. `uloha1-xsmitka.zip`
3. proveďte upload na server service.felk.cvut.cz do složky `courses/Y36PSI/prj/_váš_login_` (návod [<http://service.felk.cvut.cz/pubguide/>])

Postup odevzdání:

1. otestujte svoje řešení proti `dsnlab1` (klienta i server) i svého klienta a servera proti sobě
2. proveďte upload úlohy na server `service`
3. připravte si zdrojové texty k nahlédnutí
4. připravte si internetový prohlížeč na adresu <http://dsnlab1.felk.cvut.cz/cgi-bin/robottcp> [<http://dsnlab1.felk.cvut.cz/cgi-bin/robottcp>] a vyplňte adresu a port
5. připravte si druhou záložku/okno internetového prohlížeče na adresu http://service.felk.cvut.cz/courses/X36PKO/prj/_vas_login_ [http://service.felk.cvut.cz/courses/X36PKO/prj/_vas_login_]
6. spusťte terminál, nastavte adresář s Vaší úlohou jako aktuální a zadejte příkaz `ls -l`
7. zavolejte cvičícího

Bodování

Za úlohu můžete dostat max. 6 bodů. Penalizace:

- server nezpracovává požadavky klienta paralelně: -2
- chybně implementovaný protokol na straně klienta: -3

- chybně implementovaný protokol na straně serveru: -3
- nestabilita: -3
- malá odolnost vůči nestandardním vstupům: -3
- klient nevyzvedává tajemství plně automaticky: -3
- chyby v programu, které se neprojevily přímo: dle povahy -1 až -5
- nečistý zdrojový kód: -1
- nedodržení pokynů pro odevzdání (nejdříve upload na service, správné pojmenování archivu apod.): -2

Literatura

Robot Karel online [<http://karel.oldium.net>]

vyuka/y36psi/cviceni/uloha1-karel-zadani.txt · Poslední úprava: 2010/10/12 23:40 autor: cepekmir

Kromě míst, kde je explicitně uvedeno jinak, je obsah této wiki licencován pod následující licencí: Public Domain [<http://creativecommons.org/licenses/publicdomain/>]