VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

# Síťové aplikace a správa sítí
# HTTP nástěnka

Radek Duchoň (xducho07) 17. listopadu 2019

# 1 Úvod

- Tato dokumentace vznikla jako manuál k projektu HTTP nástěnka do předmětu ISA – Síťové aplikace a správa sítí. Projekt je reprezentován dvěma samostatnými aplikacemi – isaserver a isaclient.
- Isaserver slouží jako server s nástěnkami, které může klient spravovat pomocí HTTP API skrz aplikaci isaclient. API umožňuje prohlížet, přidávat, upravovat a mazat příspěvky na nástěnkách a nástěnky samotné. Nástěnka obsahuje seřazený seznam textových příspěvků.
- Program je implementován v jazyce C++ s využitým standardem C++11.

# 2 Návrh aplikace

## 2.1 Server

- Naslouchá na přiděleném portu a čeká na příchozí komunikaci.
- Při příchozí komunikaci přijme data (omezené množství), zpracuje je a odpoví.
- Podle zaslaných požadavků spravuje nástěnky.
- Po zpracování požadavku začne znovu čekat na příchozí komunikaci.
- S volitelným argumentem bude vypisovat příchozí i odchozí zprávy.
- Při chybě se ukončí a vypíše chybovou zprávu nápovědu.

## 2.2 Klient

- Zpracuje vstupní argumenty a vytvoří na jejich základě HTTP požadavek.
- Jedním spuštěním zašle právě jeden požadavek. (Při správném spuštění.)
- Po zaslání požadavku přijme odpověď a vypíše ji.
- Při chybě se ukončí a vypíše chybovou zprávu a nápovědu.

# 3 Popis implementace

## 3.1 Server

Makro handle_error(msg):
- Vypíše chybové hlášení, nápovědu a vrátí 1.

Makro vperr(msg):
- Vypíše předdefinovanou chybu pomocí perror(), nápovědu a zavolá return.

Funkce int main(int arg, char *argv[]):
- Zavolá funkci arghelp() pro případný výpis nápovědy a ukončení programu.
- Zkontroluje, zda jsou zadány všechny potřebné argumenty a jejich validitu.
- Zavolá funkci server, odkud by už neměl proběhnout návrat.

Funkce int arghelp(int argc, char *argv[]):
- Snaží se najít argument pro nápovědu. („-h“ nebo „—help“)
- V případě úspěchu vypíše nápovědu pomocí funkce help().

Funkce void help(void):
- Vypíše nápovědu

Funkce void server(int port):
- Vytvoří a inicializuje potřebné struktury pro síťovou komunikaci.
- K nastavení naslouchání používá funkce socket(), bind() a listen().
- V cyklu následně zpracovává postupně příchozí spojení, k tomu je využívána funkce accept(), která vrací descriptor, pro který se pomocí funkce setstockopt() a struktury timeval nastaví časové omezení pro příjem dat – v implementovaném provedení na půl sekundy.
- Pomocí funkcí receive() a response() přijme zprávu a vytvoří odpověď.
- Odpověď pošle a spojení uzavře, poté se cyklus vykonává znovu.

Funkce string receive(int socket):
- Načítá data do bufferu velikosti definované makrem BUFFER, které je 16384, což odpovídá maximální velikosti jednoho packetu. pokud nebude buffer stačit, nebo dojde zpráva po menších částech, načítá se zpráva dále v cyklu, dokud není nalezen konec hlavičky, nebo dokud není načtena maximální délka zprávy – definována makrem MAX_MSG_LENGTH.
- Case-insensitive část hlavičky se přepíše na malá písmena, poté se zkontroluje přítomnost „content-length“ a případně se načítá tělo.

Funkce string response(string msg):
- Jako statickou proměnnou uchovává nástěnky ve slovníku vektorů.
- Pomocí regulárních výrazů se odstraní nadbytečné mezery pro usnadnění práce. Pomocí soustavy regulárních výrazů se poté vybírá akce a na základě akce a aktuálního stavu proměnné s nástěnkami se vrací odpověď.

## 3.2 Client

Makro handle_error(msg):
- Vypíše chybové hlášení, nápovědu a vrátí 1.

Makro perr(msg):
- Vypíše předdefinovanou chybu pomocí perror(), nápovědu a vrátí 1.

Funkce int main(int arg, char *argv[]):
- Zavolá funkci arghelp() pro případný výpis nápovědy a ukončení programu.
- Zkontroluje, zda jsou zadány všechny potřebné argumenty a jejich validitu.
- Vrací funkci client(), které předá IP adresu z funkce HostToIp(), port a zprávu na základě argumentů vytvořenou pomocí funkce msg_create().

Funkce int arghelp(int argc, char *argv[]):
- Snaží se najít argument pro nápovědu. („-h“ nebo „—help“)
- V případě úspěchu vypíše nápovědu pomocí funkce help().

Funkce void help(void):
- Vypíše nápovědu

Funkce char *HostToIp(char *host):
- Pokusí se získat IP adresu zadané adresy, v případě úspěchu ji vrátí, jinak vrátí NULL. Pokud byla zadána IP adresa, přeloží se sama na sebe.

Funkce string msg_create(char *host, char *args[]):
- Pomocí soustavy podmínek s porovnáním řetězců s argumenty vybere variantu odesílané zprávy a z argumentů ji poskládá.
- Tato funkce předpokládá správně zadané pole argumentů, proto je potenciálně nebezpečná a je nutné, aby byla ošetřena před zavoláním.

Funkce int client(char *host, int port, string msg):
- Vytvoří a inicializuje potřebné struktury pro síťovou komunikaci.
- Pro připojení k serveru se využívají funkce socket(), inet_ptin() a connect().
- Pokud nenastala chyba a dostalo se na odeslání dat, vrácí se funkce process(), které je pomocí funkce receive() předána obdržená odpověď.

Funkce string receive(int socket):
- Načítá data do bufferu velikosti definované makrem BUFFER, které je 16384, což odpovídá maximální velikosti jednoho packetu. pokud nebude buffer stačit, nebo dojde zpráva po menších částech, načítá se zpráva dále v cyklu, dokud není nalezen konec hlavičky a načteno celé tělo.

Funkce int process(string msg):
- Zkontroluje, zda načtená zpráva obsahuje konec hlavičky, pokud ano, vypíše hlavičku na standardní chybový výstup a tělo na standardní výstup, v opačném případě vypíše celou zprávu na chybový výstup. Pokud byla obdržena positivní odpověď od serveru (2XX), vrátí 0, jinak vrátí -1.

# 4 Základní informace o programu

## 4.1 Isaserver

### 4.1.1 Rozšíření

- Jména nástěnek a zprávy mohou obsahovat také znak ´_´.
- Přidaný volitelný argument verbose zadaný jako „--verbose“, nebo „-v“.
- Nápověda se vypisuje i pro dlouhou variantu „--help“.
- Argumenty nevyžadují pevné pořadí.

### 4.1.2 Omezení

- Přijímá zprávy pouze do délky definované makrem MAX_MSG_LENGTH, které je nastaveno na 32768. Důvodem je, že není žádoucí, aby mohl někdo server obsadit na příliš dlouho, nebo aby nástěnky obsahovaly až příliš dlouhé texty. Zároveň to slouží jako obrana proti posílání nesmyslných extrémně dlouhých dat, která se odmítnou.
- Pro úspěšné zkompilování aplikace je vyžadován standard alespoň C++11.
- Porty 0-1023 jsou vymezené pro uživatele s oprávněním root.

## 4.2 Isaclient

### 4.2.1 Rozšíření

- Nápověda se vypisuje i pro dlouhou variantu „--help“.
- Lze zaměnit pořadí argumentů host a port.
- Oproti serveru přijímá neomezeně dlouhé zprávy.

### 4.2.2 Omezení

- Část argumentů <command> musí být zadána jako poslední.
- Pro úspěšné zkompilování aplikace je vyžadován standard alespoň C++11.
- Porty 0-1023 jsou vymezené pro uživatele s oprávněním root.

# 5 Návod k použití

## 5.1 Isaserver

- Překlad: příkazem „make“ pomocí přiloženého souboru Makefile.
- Spuštění: ./isaserver -p <port> [-v | --verbose]
- Popis argumentů:
  - <port> je celé číslo v rozsahu 0-65535.
  - -v/--verbose zapne „upovídaný režim“ pro výpis zpráv na serveru.
  - S argumentem -h/--help se vypíše nápověda.
- Příklad spuštění:
  - ./isaserver -p 2025 -v
    - Spustí server na portu 2025 s výpisem zpráv

## 5.2 Isaclient

- Překlad: příkazem „make“ pomocí přiloženého souboru Makefile.
- Spuštění: ./isaclient -H <host> -p <port> <command>
- Popis argumentů:
  - <host> je doménové jméno nebo IP adresa.
  - <port> je celé číslo v rozsahu 0-65535.
  - S argumentem -h/--help se vypíše nápověda.
  - <command> je směs argumentů podle které se tvoří zpráva, detailně viz README.
- Příklady spuštění:
  - ./isaserver -H merlin.fit.vutbr.cz -p 2025 board add hello
    - Připojí se na server merlin.fit.vutbr.cz na port 2025
    - Vytvoří nástěnku „hello“
  - ./isaserver -H merlin.fit.vutbr.cz -p 2025 item add hello "Hello world!"
    - Přidá do nástěnky „hello“ zprávu „hello world“
  - ./isaserver -H merlin.fit.vutbr.cz -p 2025 boards
    - Pošle žádost k vrácení seznamu nástěnek.
  - ./isaserver -H merlin.fit.vutbr.cz -p 2025 board list hello
    - Pošle žádost k vrácení obsahu nástěnky „hello“.

# 6 Studijní literatura


- https://www.root.cz/clanky/protokol-http-1-1-pod-lupou/
- https://www.sallyx.org/sally/c/linux/socket
- https://www.sallyx.org/sally/c/linux/http
- https://www.w3.org/Protocols/rfc2616/rfc2616-sec4.html
- https://linux.die.net/man/3/setsockopt
- http://man7.org/linux/man-pages/man2/socket.2.html
- http://man7.org/linux/man-pages/man2/bind.2.html
- http://man7.org/linux/man-pages/man2/listen.2.html
- http://man7.org/linux/man-pages/man2/accept.2.html
- http://man7.org/linux/man-pages/man2/connect.2.html
- http://man7.org/linux/man-pages/man2/read.2.html