Dokumentácia k projektu do kurzu ISA

Zoznam služieb bežiacich na zadaných počítačoch

19. novembra 2012

Autor: Fridolín Pokorný, xpokor32@stud.fit.vutbr.cz
Fakulta Informačních Technologií
Vysoké Učení Technické v Brně

Obsah

1 Úvod

Cieľom úlohy bolo vytvoriť program `tcpsearch`, ktorý vyhľadáva služby na zadaných počítačoch v zadanom rozsahu TCP portov. Pre komunikáciu má program využívať knihovňu pre prácu s `BSD` sockets dostupnú na operačných systémoch Unixového typu. Program má pracovať v príkazovom riadku, po spustením s užívateľom neinteraguje. Informácie pre užívateľa majú byť vypisované na štandardný výstup, informácie o chybách a problémoch majú byť vypisované na štandardný chybový výstup.

Užívateľ musí špecifikovať pomocou parametrov súbor, v ktorom sú doménové mená, prípadne adresy IPv4 alebo IPv6 serverov, ktoré budú skúmané z pohľadu bežiacich služieb na portoch zadaných ako parameter programu. Špeciálny prípad súboru udáva parameter -, kedy sa namiesto textového súboru použije štandardný vstup.

Program má byť implementovaný v programovacom jazyku C alebo C++, preložiteľný štandardnými nástrojmi.

2 Implementácia

Program `tcpsearch` je implementovaný v jazyku C++ s využitím štandardnej knihovni jazyka C++ a knihovnou pre prácu s `BSD` sockets pod Unixom.

Pri implementácii bol zohľadnený objektovo orientovaný návrh aplikácie. Nakoľko je aplikácia implementovaná v objektovo orientovanom jazyku C++ podľa štandardu ISO/IEC 14882:1998, je i program implementovaný objektovo.

2.1 Parametre programu

Informácie o podporovaných parametroch je možné získať pri spustení programu s parametrom `-h`:

```
$ ./tcpsearch -h
tcpsearch - port discover tool
Written by Fridolin Pokorny <xpokor32@stud.fit.vutbr.cz>

Usage:
        ./tcpsearch [-t TIME] [-v] -p PORT_RANGE FILE

Options:
        FILE            file whith domain name or IP address
        -t TIME         specify wait time
        -p PORT_RANGE   comma-separated list of ports and port ranges
        -v              verbose info messages
```

Pokiaľ je programu namiesto názvu súboru predaný znak `-`, program číta jednotlivé záznamy zo štandardného vstupu. Parametrom `-t` je možné zadať dobu v sekundách, po ktorú program čaká na odpoveď od servru. Ak je tento čas prekročený, program sa automaticky odpojí od skúmaného portu a pokračuje v zisťovaní dostupných služieb.

Povinný parameter `-p` umožňuje zadať porty pomocou výpisu, prípadne pomocou rozsahu. Napríklad volaním programu pomocou `-p 80-89`, program preskúma porty od 80 do 89 (vrátane 80 a 89). Na druhú stranu pri volaní `-p 80,89`, program preskúma porty 80 a 89. Obe formy je možnéžné ľubovoľne kombinovať pri oddelovaní čiarkou. V prípade zadávania rozsahu však musí byť rozsah zadaný vždy od menšieho čísla portu k väčšiemu. Inak je program ukončený chybovým hlásením.

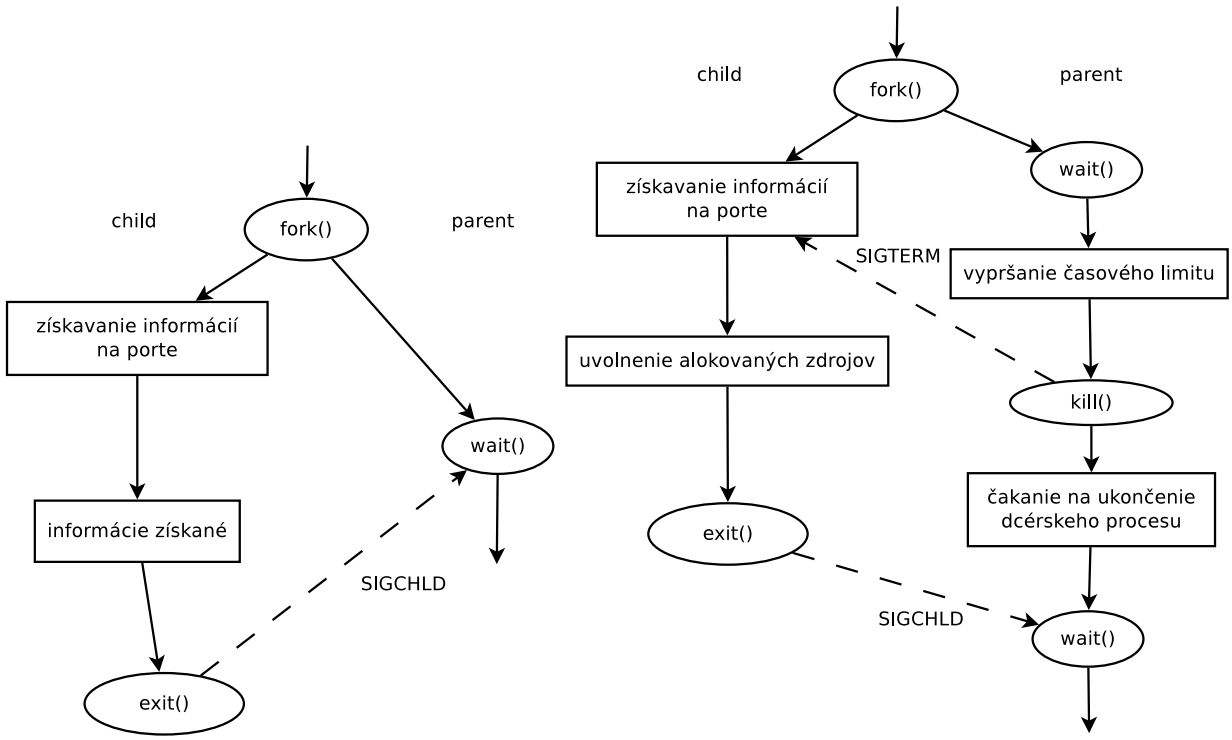
Program bol ďalej rozšírený o parameter `-v` (od slova *verbose*), ktorý umožňuje podporu podrobnejšieho výpisu v prípade neúspechu zistenia služby na zadanom počítači. Užívateľ tak je informovaný, či došlo k vypršaniu časového limitu, prípadne nemožnosť nadviazať spojenie na zadanom TCP porte a pod.

2.2 Tok behu programu

Po spustení je v hlavnom procese inicializovaný jedináčik (angl. *singleton*) `Arg`, ktorý spracuje parametre príkazového riadku. Následne dôjde k inicializácii singletonu `Host`, ktorý spracováva jednotlivé záznamy IP adries alebo doménových mien zo vstupného súboru, prípadne zo štandardného vstupu. Prázdne a zakomentované riadky sú vo vstupnom súbore ignorované. Komentáre vo vstupnom súbore začínajú znakom `#` a končia znakom nového riadku.

Po úvodných inicializáciach je zavolaná funkcia `tcpsearch()` pre každý záznam na vstupe. Vo funkcii sa následne vytvára proces, ktorý sa snaží pripojiť na zadaný port a získať tak informácie o bežiacej službe. Ak sa potomku nepodarí získať požadované informácia do stanoveného času, ktorý je zadaný pomocou parametra `-t`, rodičovský proces zašle signál `SIGTERM` a ukončí tak nadviazané spojenie. Všetky alokované zdroje sú pred ukončíčením dcérskeho procesu uvolnené.

Pokiaľ sa dcérskemu procesu podarí získať informácie o službe do zadaného času, po uvolnení systémových prostriedkov a ukončení dcérskeho procesu je zaslaný signál `SIGCHLD` operačným systémom. Rodičovský proces ako reakciu na tento signál ukončí čakanie na dcérsky proces a umožňuje tak pokračovať vo výpočte.



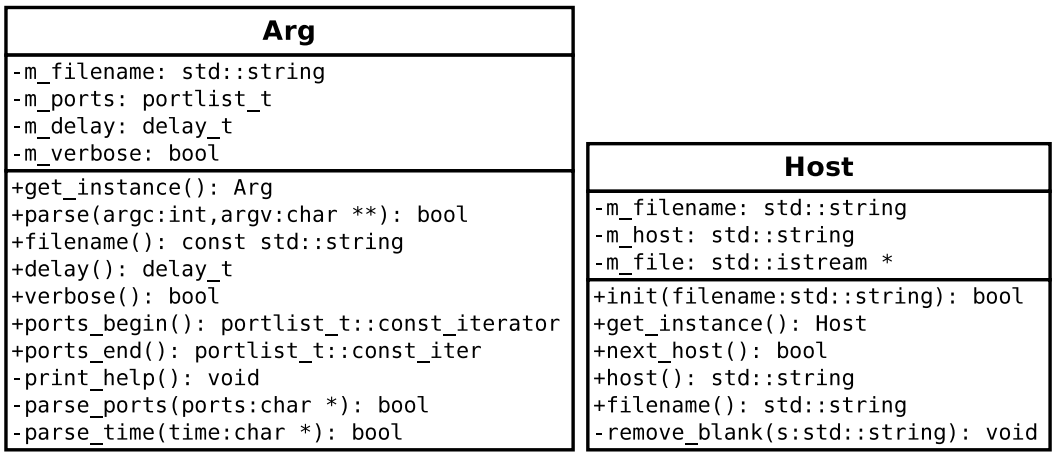
Obr. 1: Hlavný tok programu. Vľavo je informácia o bežiacej službe na zadanom porte získaná do vypršania časového limitu. Vpravo časový limit vypršal a dcérsky proces bol tak ukončený zaslaním signálu `SIGTERM`

Pri výstupe daných služieb je v prípade nájdenia záznamu IPv4 alebo IPv6 pomocou lokálneho resolveru vypísaná i nájdená IP adresa, pričom je uprodnostňovaný tvar v IPv6.

2.3 Implementované triedy

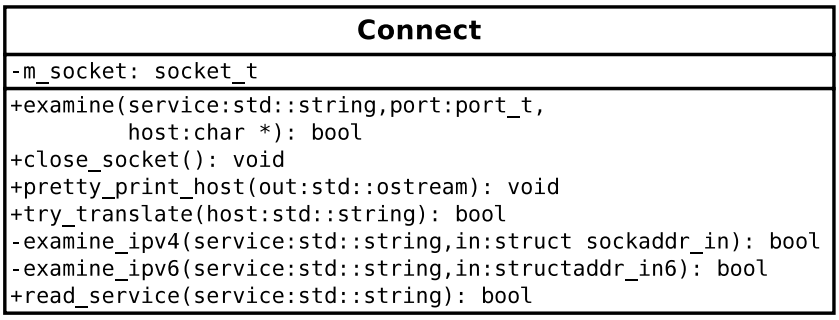
Triedy `Host` a `Arg` sú navrhnuté podľa návrhového vzoru jedináčik (angl. *singleton*). V oboch prípadoch by boli tieto triedy inštanciované iba raz, naviac je tým možné zabrániť vytvoreniu globálnych premenných, ktoré by bolo nutné využívať pri zaslaní signálov od operačného systému.

V triede `Arg` je najdôležitejšou metódou metóda `parse()`, ktorá spracuje parametre príkazového riadku. Jednotlivý rozsach portov, ktorý bude skúmaný trieda `Arg` zapuzdruje a poskytuje výhradne pomocou verejného rozhrania vo forme sprístupnenia konštantným iterátorom.



Obr. 2: UML diagramy tried `Host` a `Arg`

Trieda `Host` je stavová a uchováva informácie o stave vo vstupnom súbore. Pre úsporu pamäťových nárokov aplikácie aplikácia postupne číta záznamy v otvorenom vstupnom súbore. Pomocou metódy `next_host()` je tak možné získať ďalejšie doménové meno zo vstupného súboru (prípadne IP adresu). Pomocou metódy `host()` je následne možné spracovanú adresu sprístupniť. Implementované metódy v tejto triede sa starajú o spracovanie adresy, ignorovanie zakomentovaných a prázdnych riadkov vo vstupnom súbore tak, aby sprístupnená adresa bola pripravená na ďalšiu prácu.



Obr. 3: UML diagram triedy `Connect`

Trieda `Connect` zapuzdruje operácie nad `BSD` socketmi a aplikácii umožňuje pristupovať prostredníctvom rozhrania k metódam, prípadne i k triednym metódam. Programu je tak prístupné len verejné rozhranie.

Pred samotným skúmaním portov je spracovaná adresa najprv preložená na IP adresu, aby bolo zaručené, že preklad prebehol správne. Pokiaľ sa preklad nepodarí, je o neúspešnosti informovaný užívateľ len raz.

2.4 Dokumentácia

Okrem dokumentácie `manual.pdf` je program distribuovaný spolu s textovým súborom `README`, ktorý obsahuje krátky popis programu, prípadne príklad spustenia programu.

Zdrojové kódy aplikácie sa snažia byť samodokumentačné. Okrem rozdelenia problému na podproblémy, ktoré sú implementované pomocou funkcií a metód, program poskytuje doku-

mentáciu vo forme komentárov podľa dokumentačného nástroja `doxygen`. Je tak možné vytvoriť implementačnú dokumentáciu priamo zo zdrojových súborov aplikácie.

3 Ukážky

Vybrané ukážky reprezentujú výstup programu pri rôznych zadaných parametroch programu. Vstupný súbor `in` obsahuje po rade tieto záznamy:

```
80.79.29.98
merlin6.fit.vutbr.cz
neexistujuci
```

3.1 Ukážka č. 1

Užívateľ je informovaný o získaní informácií výpisom na štandardný výstup. O chybe pri preklade je užívateľ informovaný výpisom na štandardný chybový výstup. Každá chybová hláška začína prefixom `Err:`.

```
$  ./tcpsearch in -p 22
80.79.29.98
22
SSH-2.0-OpenSSH_5.8p1-hpn13v10
2001:67c:1220:8b0::93e5:b013 (merlin6.fit.vutbr.cz)
22
SSH-2.0-OpenSSH_4.3
neexistujuci
Err: unknown service or name
```

3.2 Ukážka č. 2

Jednotlivé riadky výstupu obsahujú len čísla portov, u ktorých sa podarili získať informácie o bežiacej službe.

```
$  ./tcpsearch in -p 1-3
80.79.29.98
2001:67c:1220:8b0::93e5:b013 (merlin6.fit.vutbr.cz)
neexistujuci
Err: unknown service or name
```

3.3 Ukážka č. 4

V prípade zadania parametra `-v` (`verbose`) je užívateľ bližšie informovaný o problémoch spojených so získavaním informácie o bežiacej službe. Takéto výpisy sú považované za varovné a sú uvedené prefixom `Warn:`, vypisované na štandardný chybový výstup.

```
$  ./tcpsearch in -p 1,80 -t 1 -v
80.79.29.98
1
Warn: Cannot connect to given port: Connection refused
80
Warn: Connection timeout!
2001:67c:1220:8b0::93e5:b013 (merlin6.fit.vutbr.cz)
```

```
1
Warn: Cannot connect to given port: Connection refused
80
Warn: Connection timeout!
neexistujuci
Err: unknown service or name
```

3.4 Ukážka č. 3

V prípade akejkoľvek chyby je užívateľ informovaný chybovým hlásením a program je ukončený s odpovedajúcim návratovým kódom (v prípade chyby je nenulový). Takéto riešenie umožňuje program začleniť do programov vytváraných pomocou skriptov.

```
$ ./tcpsearch -p 1 /etc/shadow
Err: /etc/shadow: Permission denied
$ echo $?
2
```

4 Metriky kódu

Metriky boli zisťované pri preklade pomocou prekladača GNU GCC vo verzii 4.7.1 pod operačným systémom GNU/Linux - 64 bitová architektúra. Preklad bol bez ladiacich informácií.

Metrika	Hodnota metriky
Počet riadkov zdrojového súboru	1114
Počet súborov	9
Veľkosť statických dát	1312B
Veľkosť spustitelného súboru	36KB

Tabuľka 1: Metriky kódu.

Literatúra

[1] Študijna opora ku kurzu ISA - Kapitola 2: Programovanie sietí TCP/IP

[2] Manuálové stránky k `BSD` socketom

[3] Prednášky a pomocné materiály kurzu ISA

[4] Prednášky a pomocné materiály kurzu IPK