TFTP w Coq-u

8 marca 2018

1 Wstęp

Zadanie polega na napisaniu docelowo w języku oprogramowania OCaml serwera protokołu TFTP [1], przy czym samo jądro serwera, obsługujące jego logikę ma zostać napisane w Coq-u i tamże zweryfikowane.

2 Etapy pracy

Praca nad zbudowaniem tego serwera będzie składała się z kilku etapów.

1. Zbudowanie w języku OCaml otoczki, która będzie odwoływała się do logiki, jaką napiszemy w Coq-u.
2. Napisanie w Coq-u implementacji logiki oraz specyfikacji tej logiki w postaci odpowiednich definicji oraz lematów.
3. Wykonanie dowodów, że podana implementacja zachowuje własności podane w lematach.

3 Wymagania dotyczące etapów

3.1 Otoczka w OCamlu

Serwer TFTP musi wykonywać następujące operacje:

- Parsować parametry wywołania.
- Wchodzić w pętlę obsługi połączeń.
- W pętli obsługi połączeń przyjmować połączenie.
- Obsłużyć połączenie.
- Powrócić do początku pętli obsługi połączeń.

Dodatkowo serwer musi reagować na polecenia wydane z klawiatury oraz pozwalać na odczytywanie zawartości pliku. Zależy nam, aby część programu napisana bezpośrednio w OCaml-u była funkcjonalna, ale nie zależy nam na tym, aby była ona rozbudowana.

3.2 Implementacja i specyfikacja w Coq-u

Sama implementacja w Coq-u powinna realizować funkcjonalność opisaną w sekcjach 2, 4, 6, 7 specyfikacji TFTP [1]. Siłą rzeczy wymagać to będzie zamodelowania pakietów opisanych w sekcji 5.

Zaimplementowane w Coq-u funkcje będą zależały od stanu wewnętrznego serwera. Stan ten obejmuje:

- stan protokołu (automat stanowy można odczytać z opisu w sekcjach 2, 4, 6, 7),
- opcjonalnie przysłany od klienta pakiet,
- informacje o pliku lokalnym, który jest przesyłany w komunikacji.

Funkcje będą produkowały nowy stan wewnętrzny oraz, opcjonalnie, pakiet, jaki ma zostać wysłany do klienta. Pakiety mają być reprezentowane jako wektory ośmiobitowych znaków.

3.3 Dowody

Chcielibyśmy, aby udowodnione zostały wszystkie oczekiwane własności napisanych programów.

Literatura

[1] K. Sollins. The TFTP protocol (revision 2). Technical report, Network Working Group, 1992. RFC1350.