ZADÁNÍ:
1) Navrhněte a implementujte síťový analyzátor v C/C++/C#, který bude schopný na určitém síťovém rozhraním zachytávat a filtrovat pakety (13 b)
2) Vytvořte relevantní manuál/dokumentaci k projektu (7b)

UPŘESNĚNÍ ZADÁNÍ:
Ad 1)
Volání programu:

./ipk-sniffer -i *rozhraní* [-p *port*] [--tcp|-t] [--udp|-u] [-n *num*]

kde

- -i *eth0* (rozhraní, na kterém se bude poslouchat. Nebude-li tento parametr uveden, vypíše se seznam aktivních rozhraní)
- -p *23* (bude filtrování paketů na daném rozhraní podle portu; nebude-li tento parametr uveden, uvažují se všechny porty)
- -t nebo --tcp (bude zobrazovat pouze tcp pakety)
- -u nebo --udp (bude zobrazovat pouze udp pakety)
- Pokud nebude -tcp ani -udp specifikováno, uvažují se TCP a UDP pakety zároveň
- -n *10* (určuje počet paketů, které se mají zobrazit; pokud není uvedeno, uvažujte zobrazení pouze 1 paket)

Formát výstupu:

čas IP|FQDN : *port* > *IP|FQDN* : *port*

počet_vypsaných_bajtů: *výpis_bajtů_hexa výpis_bajtů_ASCII*

(takto vypíšete úplně celý paket)

Příklady volání:

./ipk-sniffer -i eth0 -p 23 --tcp -n 2
./ipk-sniffer -i eth0 --udp
./ipk-sniffer -i eth0 -n 10
./ipk-sniffer -i eth0 -p 22 --tcp --udp .... stejné jako:
./ipk-sniffer -i eth0 -p 22
./ipk-sniffer -i eth0

Příklady výstupu:

11:52:49.079012 pcvesely.fit.vutbr.cz : 4093 > 10.10.10.56 : 80

0x0000: 00 19 d1 f7 be e5 00 04 96 1d 34 20 08 00 45 00 ....4 ..
0x0010: 05 a0 52 5b 40 00 36 06 5b db d9 43 16 8c 93 e5 ..R[@.6. [..C....
0x0020: 0d 6d 00 50 0d fb 3d cd 0a ed 41 d1 a4 ff 50 18 .m.P..=. ..A...P.

0x0030: 19 20 c7 cd 00 00 99 17 f1 60 7a bc 1f 97 2e b7 . ...`z.....
0x0040: a1 18 f4 0b 5a ff 5f ac 07 71 a8 ac 54 67 3b 39 ....Z._. .q..Tg;9
0x0050: 4e 31 c5 5c 5f b5 37 ed bd 66 ee ea b1 2b 0c 26 N1._.7. .f...+.&
0x0060: 98 9d b8 c8 00 80 0c 57 61 87 b0 cd 08 80 00 a1 .....W a.....

Netisknutelné znaky budou nahrazeny tečkou. Kde nepůjde zjistit doménové jméno, ponechte IP adresu.

Ad 2)
V dobré dokumentaci se OČEKÁVÁ následující: titulní strana, obsah, logické strukturování textu, výcuc relevantních informací z nastudované literatury, popis zajímavějších pasáží implementace, sekce o testování (ve které kromě vlastního programu otestujete nějaký obecně známý open-source nástroj), bibliografie, popisy k řešení bonusových zadání.

DOPORUČENÍ:

- Při implementaci použijte knihoven pcap / libnet
  Pcap: http://www.tcpdump.org/pcap3_man.html
  Libnet: http://www.packetfactory.net/projects/libnet/
- U syntaxe vstupních voleb jednotlivým programům složené závorky {} znamenají, že volba je nepovinná (pokud není přítomna, tak se použíje implicitní hodnota), oproti tomu [] znamená povinnou volbu. Přičemž pořadí jednotlivých voleb a jejich parametrů může být libovolné. Pro jejich snadné parsování se doporučuje použít funkci getopt().
- Výsledky vaší implementace by měly být co možná nejvíce multiplatformní mezi OS založenými na unixu, ovšem samotné přeložení projektu a funkčnost vaší aplikace budou testovány na referenčním Linux image pro síťové předměty (přihlašovací údaje student / student).

ODEVZDÁNÍ:
Součástí projektu budou zdrojové soubory přeložitelné na referenčním operačním systému, funkční Makefile (či pomocné provozy C#), soubor manual.pdf a README (viz obecné pokyny). Projekt odevzdejte jako jeden soubor xlogin00.tar, který vytvoříte programem tar.