

Projekty Zaliczeniowe Laboratorium Sieci Komputerowych

Rok Akademicki 2013/2014

Michał Kalewski, Łukasz Piątkowski

Revision : 2.2

Spis treści

1	Proponowane tematy projektów zaliczeniowych	1
1.1	Uwagi	3
2	Zasady zaliczania projektów	3
3	Kryteria oceny projektów	3

1 Proponowane tematy projektów zaliczeniowych

Poniżej znajdują się lista proponowanych tematów projektów zaliczeniowych laboratorium **Sieci Komputerowych 2**:

1. Sieciowa turowa gra logiczna, np.: reversi, szachy, warcaby itp. (**1** osoba). ■□□□
 - jeden z graczy wystawia serwer, drugi się do niego podłącza
2. Zdalne zamykanie systemów operacyjnych (**1** osoba). ■□□□
 - część serwerowa uruchomiona na węzłach podlegających zarządzaniu
 - graficzna część kliencka służąca do sterowania zamykaniem systemów
 - mile widziane wsparcie również dla *Wake On LAN*
3. System równoważenia obciążenia dla grupy serwerów sieciowych (**1** osoba). ■□□□
 - równoważenie obciążenia przez rozkładanie połączeń TCP
4. Zdalna konsola — *telnet* (**1** osoba). ■□□□
 - wsparcie dla uwierzytelniania użytkowników
5. System komunikacji grupowej typu *IRC* lub komunikator internetowy typu *GG* (**2** osoby). ■■□□
 - klient w wersji graficznej
 - wsparcie dla uwierzytelniania użytkowników
6. System wymiany komunikatów typu *publish/subscribe* (**2** osoby). ■■□□
 - klient w wersji graficznej
 - możliwość zarządzania tematami z GUI
 - wsparcie dla uwierzytelniania użytkowników

7. Prosty serwer protokołu HTTP (2 osoby). ■■□□
- metody GET i HEAD
 - obsługa nagłówków http
 - zgodny z RFC2616
8. Prosty serwer protokołu FTP (2 osoby). ■■□□
- komendy: ascii, binary, mkdir, rmdir, put, get
 - zgodny z RFC959
9. System nadzoru serwerów za pomocą protokołu ICMP i połączeń TCP (2 osoby). ■■□□
- GUI do definiowania parametrów monitorowania (częstotliwość, czułość)
 - powiadamianie zainteresowanych stanem maszyn użytkowników przez e-mail
10. System poczty elektronicznej — *e-mail* (2 osoby). ■■□□
- obsługa protokołu SMTP
 - część nadawcza i odbiorcza MTA
 - wykorzystanie rekordów MX z DNS
11. System skanowania i monitorowania sieci (2 osoby). ■■■□
- jednorazowe i okresowe odkrywanie węzłów działających w sieci
 - odkrywanie usług działających na węzłach
 - monitorowanie zmian i informowanie o nich przez e-mail
12. Sieciowa implementacja gry planszowej lub karcianej (2 osoby). ■■■□
- serwer i klient
 - implementacja reguł gry
 - przykładowo: Dominion, Sabotażysta, Dixit
13. Prosty system proxy www (2 osoby). ■■■□
- wbudowany cache obiektów
 - obsługa nagłówków
14. System wymiany plików w architekturze *peer-to-peer* (2 osoby). ■■■□
15. Internetowy system nadzoru wizyjnego (2 osoby). ■■■□
16. Implementacja protokołu routingu dynamicznego (2 osoby). ■■■□
- *RIPv2* lub *OSPF*
17. Zaawansowany system równoważenia obciążenia dla grupy serwerów sieciowych (2 osoby). ■■■■
- równoważenie obciążenia przez rozkładanie połączeń TCP
 - connection pool
 - dynamiczny skład grupy serwerów
 - GUI do zarządzania składem grupy
18. Zdalnie konfigurowalna zapora sieciowa (ang. *firewall*) (2 osoby). ■■■■
- wsparcie dla tablic *mangle* i *filter*
 - GUI do konfiguracji
19. Implementacja dowolnej (lecz uzgodnionej) usługi sieciowej zgodnej z dokumentami RFC¹. ■■■■

1.1 Uwagi

- Istnieje możliwość realizacji praktycznie dowolnych innych projektów sieciowych po konsultacji z prowadzącym.
- Wszystkie projekty należy zrealizować w **architekturze klient-serwer** (z wyjątkiem zadań, gdzie któraś ze stron nie występuje), przy użyciu protokołu TCP (chyba, że ustalono inaczej, np. w wypadku gier).
- Implementacje serwerów (**współbieżnych**) należy dokonać wyłącznie z zastosowaniem interfejsu gniazd BSD oraz biblioteki pthread w systemie Linux, a implementacje klientów w środowisku graficznym QT4/5 (chyba, że wyraźnie dla każdej ze stron komunikacji ustalono inaczej).
- Implementacje serwerów należy wykonać z użyciem języka C lub C++, a klientów z użyciem języka C, C++ i biblioteki QT4/5 (w uzgodnionych przypadkach C# lub Java). Aplikacje klienckie muszą działać minimum pod system Linux i Windows.
- Dodatkowe informacje na temat projektów można uzyskać podczas każdego z zajęć laboratoryjnych, podczas konsultacji lub drogą elektroniczną.
- Należy przestrzegać poniższych **zasad zaliczania projektów** (punkt nr 2)!

2 Zasady zaliczania projektów

Oto podstawowe zasady zaliczania projektów w ramach laboratorium **Sieci Komputerowych**:

- wszystkie aplikacje serwerowe muszą się kompilować bez ostrzeżeń (z opcją `-Wall`) na komputerach laboratoryjnych (na których będą zaliczane);
- do kompilacji programów napisanych w języku C należy stosować kompilator gcc, a do programów napisanych w języku C++ kompilator g++;
- programy muszą być napisane w sposób czytelny i przejrzysty, należy stosować komentarze; **uwaga**: programy nieczytelne nie będą przyjmowane!;
- do weryfikacji poprawności programu można użyć narzędzie splint² – taki test może być przeprowadzony także podczas zaliczania;
- do każdego projektu należy dołączyć krótkie (max. 3 strony A4) sprawozdanie (w formacie pdf), które powinno zawierać następujące informacje:
 - treść zadania,
 - przyjęta metoda rozwiązania problemu, w tym **opis protokołu komunikacyjnego**,
 - sposób realizacji i implementacji, w tym krótki opis zawartości plików źródłowych,
 - krótka informacja na temat obsługi programu;
- **kody źródłowe projektu należy trzymać w systemie gitlab, a sprawozdanie należy wysłać pocztą elektroniczną najpóźniej dwa dni przed terminem zaliczania!**

3 Kryteria oceny projektów

Ocena zrealizowanych projektów będzie uwzględniała następujące kryteria:

¹<http://www.ietf.org/rfc.html>

²<http://www.splint.org/>

- poprawność implementacji aplikacji, w szczególności komunikacji sieciowej;
- budowa protokołu komunikacyjnego;
- zgodność funkcjonalności aplikacji z uzgodnionymi założeniami;
- przejrzystość i czytelność kodu;
- inicjatywa i pomysłowość autora/ów w rozbudowywaniu i polepszaniu projektu;
- przestrzeganie zasad zaliczania projektów (punkt nr 2).