

Wydział Elektroniki i Technik Informacyjnych
Politechnika Warszawska

Zaawansowane programowanie w C++

Dokumentacja wstępna

Kamil Gabryjelski, Antoni Róžański

Prowadzący: Konrad Grochowski

Warszawa, 2017

1. Temat projektu

Tematem projektu jest gra Core Wars. Dwóch graczy pisze programy (tak zwanych "wojowników") w języku podobnym do assemblera, które rywalizują między sobą o kontrolę zasobów wirtualnej maszyny. Wygrywa ten gracz, którego procesy zajmą całą pamięć lub wyeliminują wszystkie procesy przeciwnika.

2. Funkcjonalność

2.1. Instrukcje

Językiem, w którym gracze będą pisali swoje programy, będzie Redcode. Instrukcje, które zamierzamy zaimplementować, to: DAT, MOV, ADD, SUB, MUL, DIV, MOD, JMP, JNZ, JMN, DJN, SPL, CMP, SNE, SLT, LDP, STP, NOP. Jeśli postępy w pracy nad projektem będą zadowalające, graczom udostępnione zostaną modyfikatory instrukcji.

Pamięć, w której operować będą instrukcje, zostanie zrealizowana jako tablica instrukcji. Adresowi w pamięci odpowiadać będzie indeks tablicy.

2.2. Interfejs graficzny

Planujemy udostępnić użytkownikowi interfejs graficzny z poziomu przeglądarki internetowej. Pierwszym etapem gry będzie stworzenie wojownika. Chcemy umożliwić graczowi pisanie programu poprzez zaznaczanie i wstawianie pożądaných instrukcji. Drugim sposobem na tworzenie wojownika będzie wczytanie go z pliku. W początkowych iteracjach projektu założymy, że program jest napisany poprawnie (bez błędów składniowych). Jeżeli starczy nam czasu, dodamy integrację z bazą danych NoSQL, dzięki czemu możliwe będzie zapisywanie i wczytywanie utworzonych wcześniej wojowników.

Po stworzeniu programu przez obu użytkowników rozpocznie się "walka". Zamierzamy zwizualizować stan rywalizacji jako dwuwymiarową tablicę, której każda komórka odpowiadać będzie adresowi w pamięci. Kolory tych komórek odpowiadać będą procesom użytkowników, co umożliwi ocenę, jaką część pamięci kontroluje każdy z graczy.

3. Narzędzia

Narzędzia, z których zamierzamy korzystać:

1. Kontrola wersji - git/GitHub
2. Środowisko programistyczne - CLion (Linux) lub Visual Studio (Windows)
3. Optymalizacja kodu - gprof
4. Generowanie dokumentacji - Doxygen
5. Komunikacja - Slack