Sprawozdanie z projektu ‘Life‘

Jan Salwowski

5 kwietnia 2014

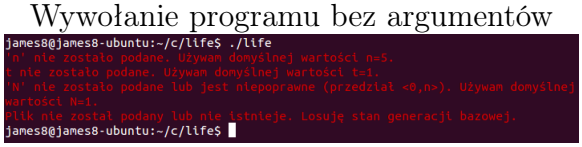
1 Testowanie

Porgram był testowany z użyciem pliku generation.txt, złączonego z projektem. Testy obejmowały zarówno podawanie w argumentach nieistniejącego pliku jak i brak podawania tego argumentu wywołania programu. W obu tych prapadkach uruchamiana jest funkcja makeLife, która tworzy losową konfigurację pierwszej generacji. Program jest zabezpieczony na wypadek podania niepoprawnych argumentów wywołania programu według następujących kryteriów:

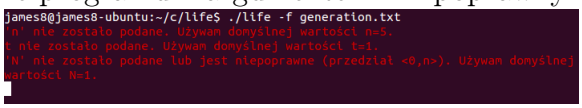
- **-n** - jeśli jest podany mniejszy od *0* lub wartość nie jest liczbą to używana jest wartość domyślna **5**
- **-N** - jeśli jest podany mniejszy od *0* lub większy od argumentu **n**, lub nie jest liczbą to używana jest wartość domyślna **1**
- **-t** - jeśli jest podany mniejszy od *0* lub większy od *30*, lub wartość nie jest liczbą to używana jest wartość domyślna **1**
- **-f** - jeśli nie istnieje plik o podanej nazwie to tworzona jest losowa generacja bazowa
- **-s** - domyślna wartość to **0**, czyli niewyświetlanie generajci w czasie tworzenia, w przypadku podania wartość to **1** i generacje są wyświetlane

Przykładowe screeny z testów:

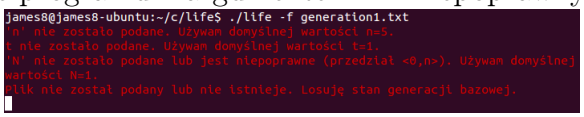
Wywołanie programu bez argumentów



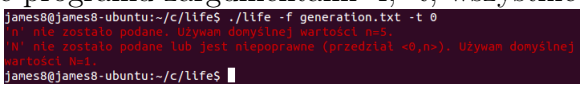
Wywołanie programu z argumentem -f i poprawnym plikiem



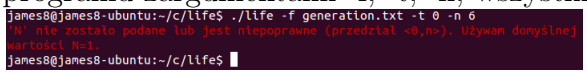
Wywołanie programu z argumentem -f i niepoprawnym plikiem



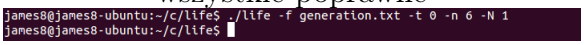
Wywołanie programu zargumentami -f, -t, wszystkie poprawne



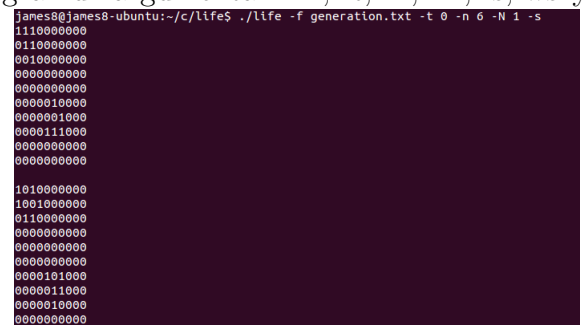
Wywołanie programu zargumentami -f, -t, -n, wszystkie poprawne



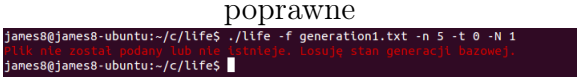
Wywołanie programu zargumentami -f, -t, -n, -N, bez prametru -s, wszystkie poprawne



Wywołanie programu zargumentami -f, -t, -n, -N, -s, wszystkie poprawne



Wywołanie programu zargumentami -f, -t, -n, -N, wszystkie poza -f poprawne



2 Problemy

Podczas tworzenia projektu problemem okazał się układ tablicy z generacją, początkowo było to *tab[x][y]*, który po wstępnych testach został zmieniony na *tab[y][x]* ze względu na łatwiejsze poruszanie się po tablicy. Można było zostawić to w pierwszej formie, lecz dla mnie układ *tab[y][x]* okazał się dużo czytelniejszy i łatwiejszy do pisania oraz prowadzneia testów.
Innym problemem okazało się napisanie oddzielnej funkcji do tworzenia folderów, kod bezpośrednio w głównej funkcji działa, natomiast przy próbie napisania i wywołania funkcji do tworzenia katalogów kod nie działał, ze

względu na małą wagę tego problemu i tylko jednokrotne tworzenie folderu, problem został zignorowany i tworzenie folderu odbywa się bezpośrendio w głównej funkcji, nie wpływa to w żadnym stopniu na działanie programu.

3 Optymalizacja

Nie zostały wprowadzone większe elementy optymalizacji. Drobną optymalizacją jest wczytywanie z pliku i zapisywanie do niego tylko i wyłącznie komórek żywych, inicjalizacja tablicy do przechowywania generacji poza alokacją zawiera wypełnianie jej zerami (komórkami martwymi).

4 Cechy wyróżniające i ograniczające

Program działa zarówno w środowisku **Windows** jak i **Linux**, został napisany i testowany w środowisku Linux, system *Ubuntu 13.10*, lecz jest również możliwa kompilacja na systemie *Windows*, wersja *32* i *64 bit*. Obsługa plików i folderów jest napisana tak, by działało to poprawnie na wcześniej wymienionych systemach.
Do programu dołączony jest plik makefile, był on używany podczas testów ze względu na usprawnienia kompilacji i testowania.