Sprawozdanie z przedmiotu
Programowanie Równoległe i Rozproszone

Mechanizmy programowania współbieżnego w SO UNIX/LINUX

Wykonał:

Imię i nazwisko	Magdalena Paszko
Nr indeksu	76024
Grupa	L2

Prowadzący : dr inż. Krzysztof Szerszeń

1. Wprowadzanie

Funkcja systemowa fork() jest jedna z najważniejszych funkcji w systemach unixowych, także w Linuxie. Za jej sprawa, i tylko za jej sprawa, mogą sie pojawiać w systemie nowe procesy. Jedynym procesem, który jest powoływany do życia w inny sposób jest proces o numerze (identyfikatorze) 1, czyli init.

Z punktu widzenie programisty, fork() jest bardzo prosta funkcja. Wywołuje sie go tak:

```
pid = fork();
```

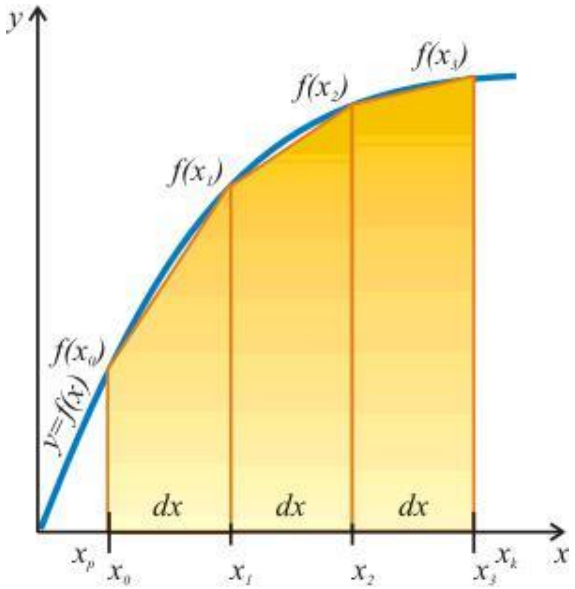
i w wyniku dostaje dwie różne wartości. Proces macierzysty, który wywołał forka, dostaje identyfikator dziecka. Dziecko dostaje w wyniku 0. Dzięki temu procesy mogą poznać "który jest który" bez uciekania sie do sztuczek. Dziecko jest prawie dokładną kopia rodzica - różni sie tylko tym, co zwraca fork(). Oczywiście w przypadku niepowodzenia dziecko nie jest tworzone a rodzic otrzymuje informacje o bledzie.

2. Definicja

Całkowanie numeryczne

Metoda polegająca na przybliżonym obliczaniu całek oznaczonych.
Proste metody całkowania numerycznego polegają na przybliżeniu całki za pomocą odpowiedniej sumy ważonej wartości całkowanej funkcji w kilku punktach.
Do uzyskania dokładniejszego przybliżenia dzieli się przedział całkowania na niewielkie fragmenty.
Ostateczny wynik to suma oszacowań całek w poszczególnych podprzedziałach.
Przedział zwykle dzieli się na równe podprzedziały.

Metoda trapezów



Przedział całkowania <xp,xk> dzielimy na n+1 równo odległych punktów x0,x1,x2,...,xn. Punkty te

wyznaczamy w prosty sposób wg wzoru:

dla i = 0,1,2,...,n

$$x_i = x_p + \frac{i}{n}(x_k - x_p)$$

Obliczamy odległość między dwoma sąsiednimi punktami - będzie to wysokość każdego trapezu:

$$dx = \frac{x_k - x_p}{n}$$

Dla każdego wyznaczonego w ten sposób punktu obliczamy wartość funkcji f(x) w tym punkcie:

$$f_i = f(x_i), \text{ dla } i = 1,2,...,n$$

Pole pod wykresem funkcji przybliżane jest polami n trapezów. Pole i-tego trapezu obliczamy wg wzoru:

dla i=1,2,...,n

$$f_i = f(x_i), \text{ dla } i = 1,2,...,n$$

Przybliżona wartość całki jest sumą pól wszystkich otrzymanych w ten sposób trapezów:

$$s = P_1 + P_2 + \ldots + P_n$$

czyli:

$$s = \frac{f_0 + f_1}{2}dx + \frac{f_1 + f_2}{2}dx + \frac{f_2 + f_3}{2}dx + \ldots + \frac{f_{n-2} + f_{n-1}}{2}dx + \frac{f_{n-1} + f_n}{2}dx$$

$$s = \frac{dx}{2}(f_0 + f_1 + f_1 + f_2 + f_2 + f_3 + \ldots + f_{n-2} + f_{n-1} + f_{n-1} + f_n)$$

$$s = \frac{dx}{2}(f_0 + 2f_1 + 2f_2 + \ldots + 2f_{n-1} + f_n)$$

$$s = dx(f_1 + f_2 + \ldots + f_{n-1} + \frac{f_0 + f_n}{2})$$

Wyprowadzony na końcu wzór jest podstawą przybliżonego wyliczania całki w metodzie trapezów.

$$\int_{x_p}^{x_k} f(x)dx \approx \frac{x_k - x_p}{n}\left(\sum_{i=1}^{n-1} f(x_p + i\frac{x_k - x_p}{n}) + \frac{f(x_p) + f(x_k)}{2}\right)$$

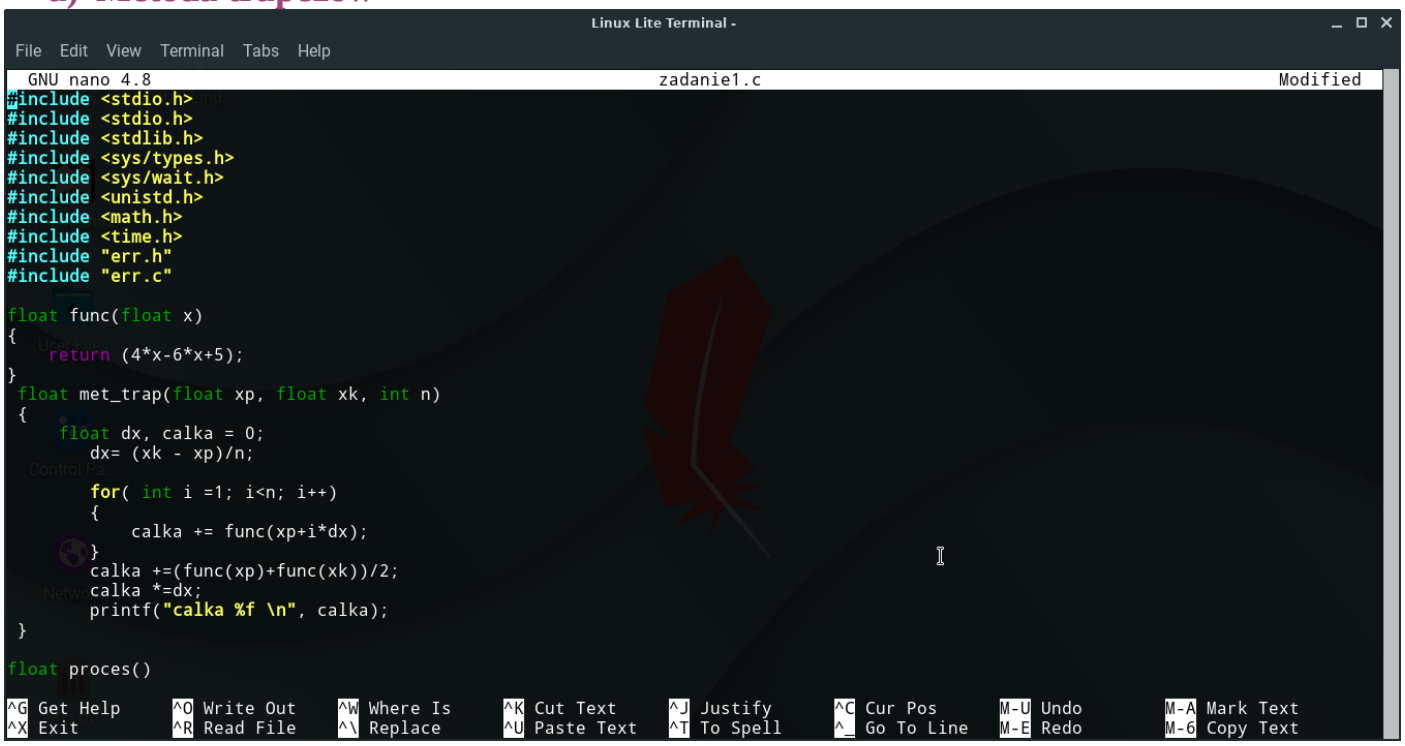
Wzór Leibniza

Wzór pozwalający obliczyć n-tą pochodną iloczynu funkcji.
Został wprowadzony przez niemieckiego matematyka Gottfrieda Leibniza.

$$\pi = 4 \cdot \sum_{n=1}^{\infty} \frac{(-1)^{n-1}}{2 \cdot n - 1} = 4 \cdot \left(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \ldots\right)$$

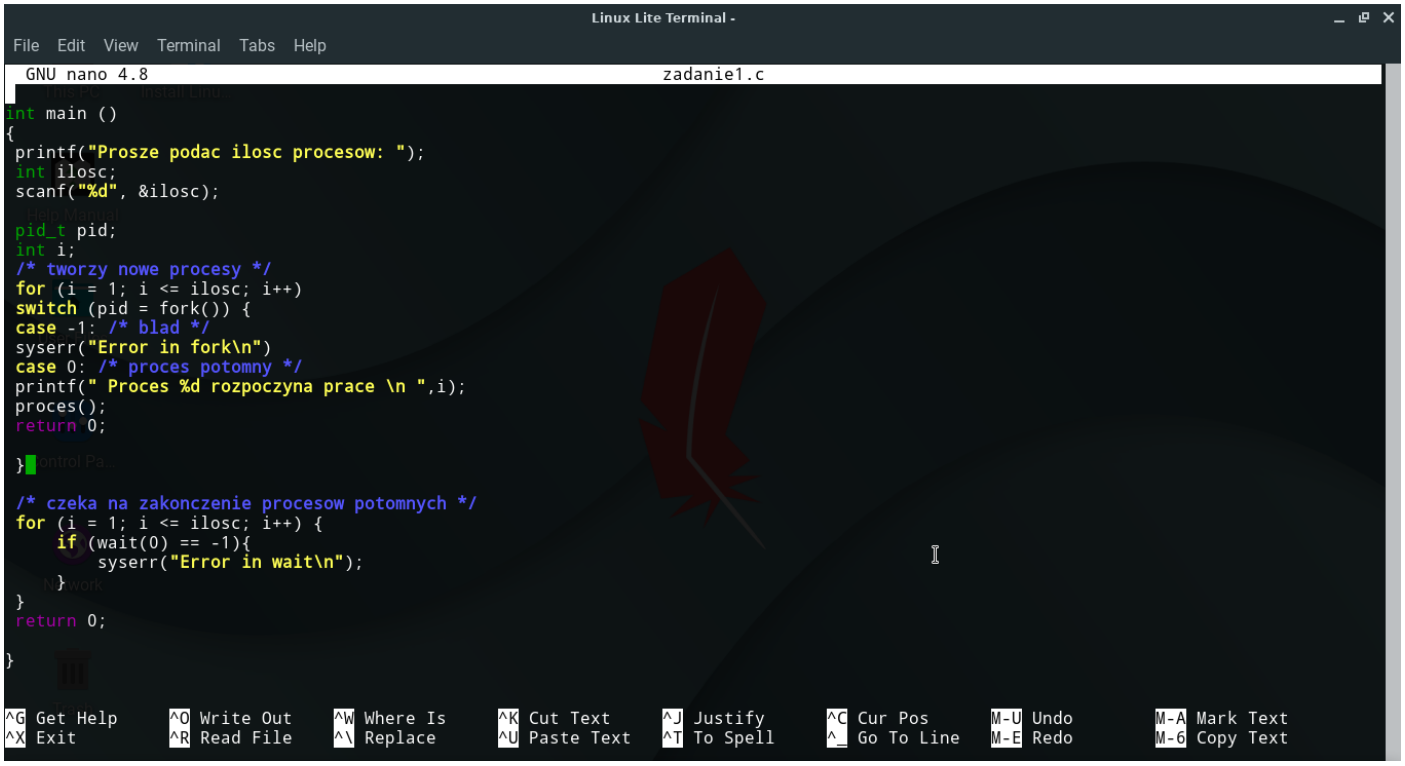
3. Implementacja algorytmów

a) Metoda trapezów



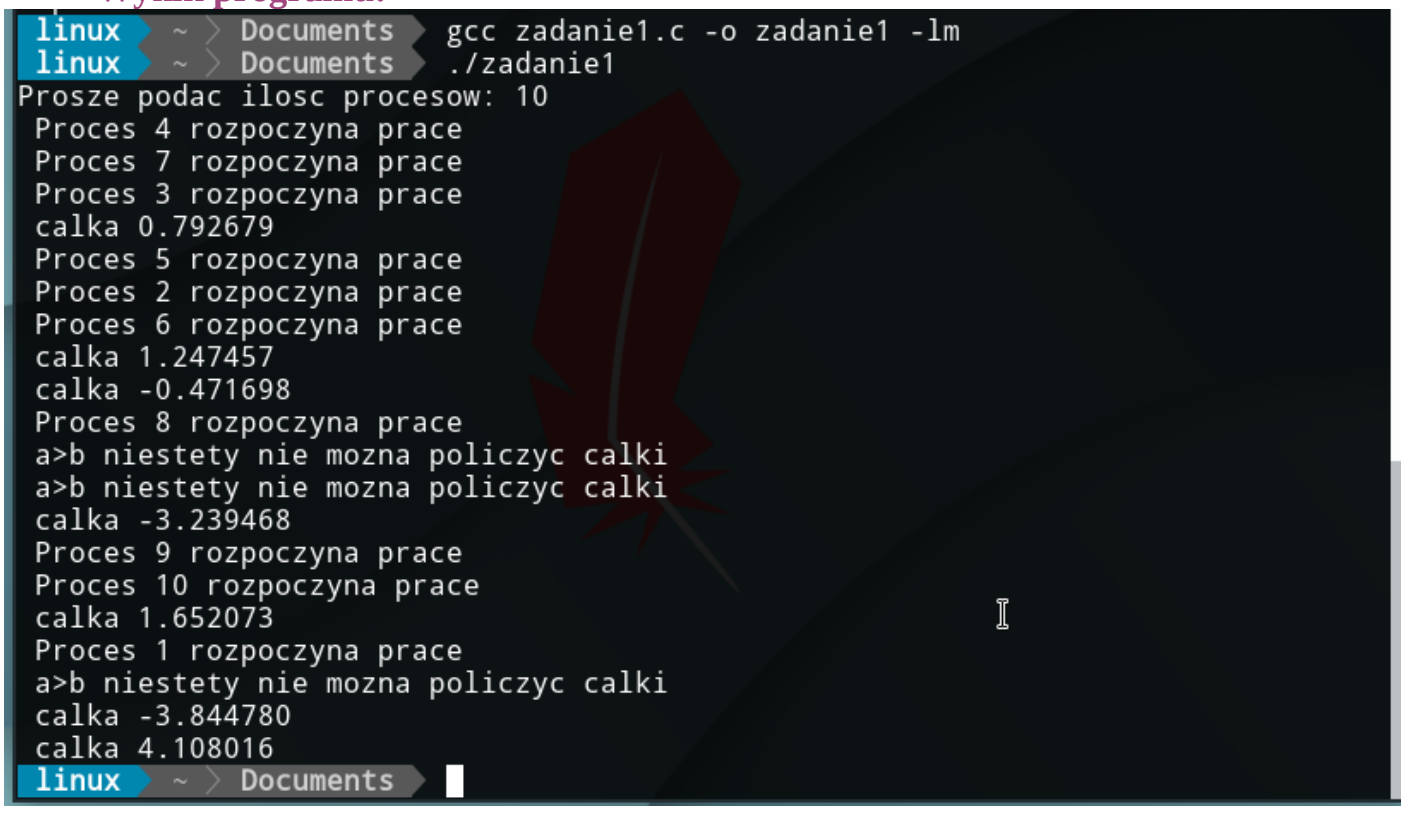
Funkcja met_trap przyjmuje przedział <a,b> oraz n liczbę trapezów do obliczenia całki.
Oblicza całkę z wzoru metody trapezów która opisałam powyżej.

Funkcja proces ustawia punkt startowy, który jest stosowany do generowania serii pseudo losowych liczb, jednak wstawiając NULL i wywołując wątki liczba zawsze była taka sama. Idealnym rozwiązaniem okazało sie mieć inne ziarno dla każdego wywołania programu. Jednym ze sposobów jest użycie srand (getpid ()); gdzie getpid () zwraca numer identyfikacyjny bieżącego procesu. Lepsze ziarno można jednak uzyskać, łącząc getpid () czyli identyfikatorem rodzica:srand(getpid () + getppid ()). Następnie rand zwraca pseudolosowa liczbę zmiennoprzecinkowa a i b oraz całkowitą n oraz sprawdza czy a<b jeśli tak to wywołuje met_tap.

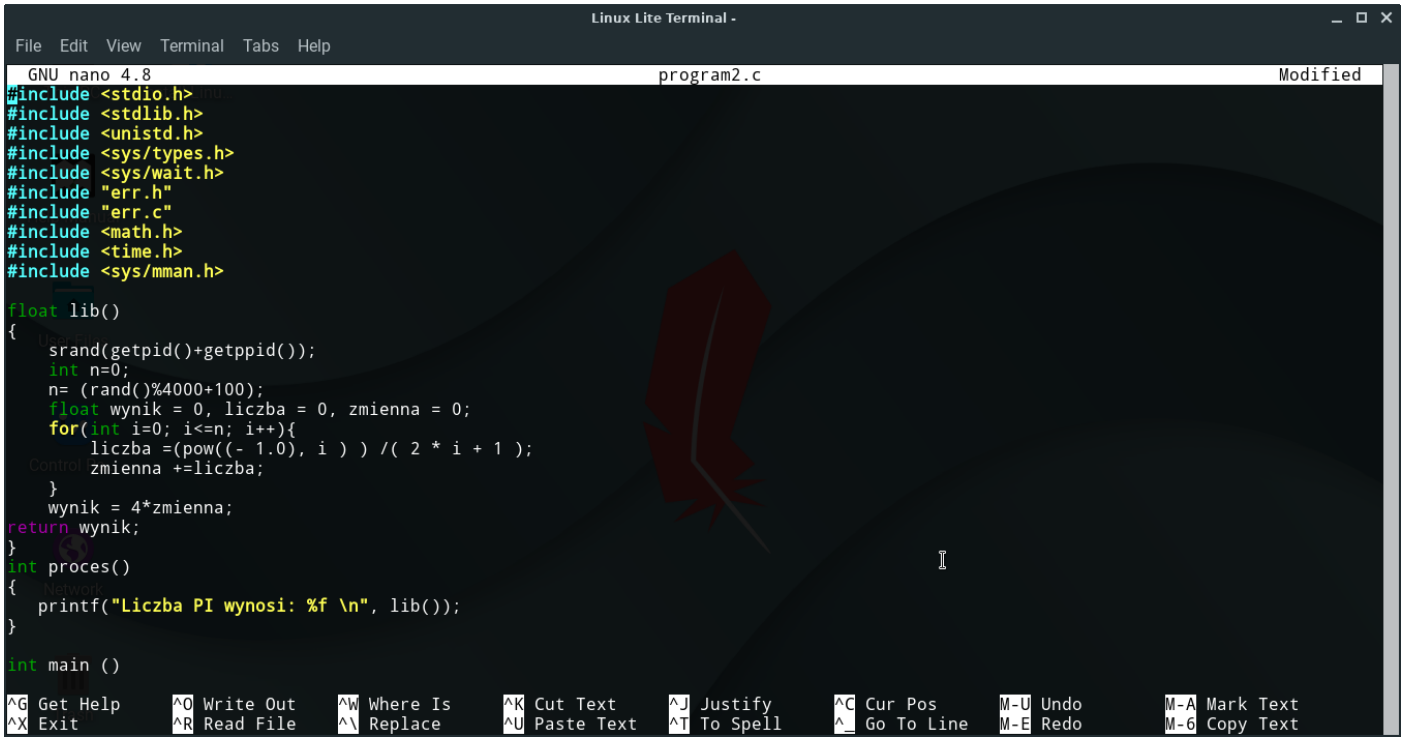


W main tak jak widać po komentarzach musimy na początku podać ilość procesów jaka ma wykonać. Tworzymy nowe procesy.
Jeśli wartość zwracana wynosi -1 oznacza to, że proces sprawdzający wartość zwracaną to proces-rodzic, ale nie udało się stworzyć procesu potomnego (dziecka) – błąd.
Jeśli wartość zwracana wynosi 0 oznacza to, że proces sprawdzający wartość zwracaną to proces-dziecko w którym funkcje proces a one met_trapezow – proces potomny.
Na koniec mamy funkcje wait co oznacza oczekiwanie na zakończenie procesu potomnego. Funkcja zwraca identyfikator (pid) procesu, który się zakończył. Gdy zwraca -1 oznacza ze jakiś proces nie zakończył się.

Wynik programu:



b) Wzór Leibniza



Funkcja lib ustawia punkt startowy, który jest stosowany do generowania serii pseudo losowych liczb analogicznie jak w pierwszym przykładzie. Oblicza przybliżenie liczby PI wzorem Leibniza do liczby która jest generowana pseudolosowo zwracając na koniec wynik przybliżenia.

Main jest analogiczny jak w przykładzie pierwszym funkcja proces wywołuje funkcje lib która oblicza liczbę PI

Wynik programu: