Sterowanie wejściem/wyjściem w GNU Octave: C-style I/O Functions

Wprowadzenie

GNU Octave oferuje bogaty zestaw funkcji do obsługi wejścia i wyjścia danych, które są podobne do tych znanych z języka C. Te funkcje umożliwiają interakcję z użytkownikiem oraz odczytywanie i zapisywanie danych do plików. W tym dokumencie omówimy kluczowe funkcje wejścia/wyjścia w stylu C dostępne w GNU Octave.

Podstawowe funkcje wejścia/wyjścia

Funkcje wejścia/wyjścia standardowego

Najczęściej używane funkcje wejścia/wyjścia to te związane z konsolą standardową (stdin, stdout, stderr). Oto kilka najważniejszych funkcji:

disp

Funkcja `disp` wyświetla tekst lub wartość zmiennej bez dodatkowych informacji, takich jak nazwa zmiennej czy znaki nowej linii.

```
disp('Hello, World!');
```

printf

Funkcja `printf` działa podobnie do funkcji `printf` w C. Pozwala na formatowanie i wyświetlanie danych.

```
a = 5;
b = 10;
printf('Suma %d + %d wynosi %d\n', a, b, a+b);
```

scanf

Funkcja `scanf` pozwala na wczytanie danych od użytkownika w stylu C.

```
num = scanf('%d');
disp(['Wczytana liczba: ', num2str(num)]);
```

Operacje na plikach

GNU Octave oferuje wiele funkcji do pracy z plikami. Oto kilka kluczowych funkcji:

fopen i fclose

Funkcje `fopen` i `fclose` służą do otwierania i zamykania plików.

```
fid = fopen('data.txt', 'r'); % Otwórz plik do odczytu
if fid == -1
    disp('Nie udało się otworzyć pliku.');
else
    disp('Plik został pomyślnie otwarty.');
    fclose(fid); % Zamknij plik
end
```

fprintf

Funkcja `fprintf` pozwala na formatowane zapisywanie danych do pliku.

```
fid = fopen('output.txt', 'w');
if fid ~= -1
    fprintf(fid, 'Liczba: %d\n', 42);
    fclose(fid);
else
    disp('Nie udało się otworzyć pliku.');
end
```

fscanf

Funkcja `fscanf` pozwala na formatowane odczytywanie danych z pliku.

```
fid = fopen('data.txt', 'r');
if fid ~= -1
    data = fscanf(fid, '%f'); % Odczytaj wszystkie liczby zmiennoprzecinkowe z pliku
    fclose(fid);
    disp(data);
else
    disp('Nie udało się otworzyć pliku.');
end
```

Przykłady użycia funkcji wejścia/wyjścia

Przykład 1: Wczytanie i wyświetlenie danych z pliku

Rozważmy przypadek, gdy chcemy wczytać dane z pliku tekstowego i wyświetlić je w konsoli.

```
fid = fopen('data.txt', 'r');
if fid == -1
    disp('Nie udało się otworzyć pliku.');
else
    % Wczytaj dane z pliku
    data = fscanf(fid, '%f'); % Załóżmy, że dane to liczby zmiennoprzecinkowe
    fclose(fid); % Zamknij plik

    % Wyświetl dane
    disp('Dane z pliku:');
    disp(data);
end
```

Przykład 2: Zapisanie danych do pliku

Teraz rozważmy przypadek, gdy chcemy zapisać dane do pliku tekstowego.

```
x = linspace(0, 2*pi, 100);
y = sin(x);

% Otwórz plik do zapisu
fid = fopen('sin_data.txt', 'w');
if fid == -1
    disp('Nie udało się otworzyć pliku.');
else
    % Zapisz dane do pliku
```

```
    for i = 1:length(x)
        fprintf(fid, '%f %f\n', x(i), y(i));
    end
    fclose(fid); % Zamknij plik

    disp('Dane zostały zapisane do pliku sin_data.txt.');
end
```

Zarządzanie strumieniami wejścia/wyjścia

GNU Octave umożliwia również zarządzanie strumieniami wejścia/wyjścia poprzez funkcje takie jak `fopen`, `fclose`, `fseek`, `ftell` itp.

Przykład: Przesunięcie wskaźnika pliku

Można użyć funkcji `fseek` do przesunięcia wskaźnika pliku.

```
fid = fopen('data.txt', 'r');
if fid == -1
    disp('Nie udało się otworzyć pliku.');
else
    fseek(fid, 10, 'bof'); % Przesuń wskaźnik o 10 bajtów od początku pliku (BOF)
    data = fscanf(fid, '%f', 1); % Odczytaj jedną liczbę z pozycji wskaźnika
    fclose(fid);

    disp(['Wczytana liczba: ', num2str(data)]);
end
```

Podsumowanie

GNU Octave oferuje bogaty zestaw funkcji wejścia/wyjścia, które są podobne do tych znanych z języka C. Dzięki nim można efektywnie realizować operacje na plikach oraz komunikować się z użytkownikiem. Funkcje takie jak `disp`, `printf`, `fopen`, `fclose`, `fprintf` i `fscanf` pozwalają na precyzyjną kontrolę nad danymi wejściowymi i wyjściowymi. Dodatkowo, możliwości zarządzania strumieniami dają większą elastyczność w pracy z plikami.