



Politechnika Białostocka  
Wydział Elektryczny  
Katedra Elektrotechniki Teoretycznej i Metrologii

Instrukcja  
do pracowni specjalistycznej z przedmiotu

### **Informatyka 1**

Kod przedmiotu: **EZ1D200 008**  
(studia niestacjonarne)

## **JĘZYK C - ŚLEDZENIE WYKONANIA PROGRAMU, DEBUGGER**

Numer ćwiczenia

**INF09Z**

Autor:  
dr inż. Jarosław Forenc

Białystok 2017

## **Spis treści**

|   |           |
|---|-----------|
| <b>1. Opis stanowiska .....</b>           | <b>3</b>  |
| 1.1. Stosowana aparatura .....            | 3         |
| 1.2. Oprogramowanie.....                  | 3         |
| <b>2. Wiadomości teoretyczne.....</b>     | <b>3</b>  |
| 2.1. Program uruchomieniowy.....          | 3         |
| 2.2. Punkty kontrolne.....                | 4         |
| 2.3. Śledzenie wykonania programu.....    | 5         |
| 2.4. Obserwowanie wartości zmiennych..... | 6         |
| <b>3. Przebieg ćwiczenia.....</b>         | <b>9</b>  |
| <b>4. Literatura.....</b>                 | <b>10</b> |
| <b>5. Pytania kontrolne .....</b>         | <b>10</b> |
| <b>6. Wymagania BHP.....</b>              | <b>11</b> |

---

**Materiały dydaktyczne przeznaczone dla studentów Wydziału Elektrycznego PB.**

© Wydział Elektryczny, Politechnika Białostocka, 2017 (wersja 3.1)

Wszelkie prawa zastrzeżone. Żadna część tej publikacji nie może być kopiowana i odtwarzana w jakiegokolwiek formie i przy użyciu jakichkolwiek środków bez zgody posiadacza praw autorskich.

## 1. Opis stanowiska

### 1.1. Stosowana aparatura

Podczas zajęć wykorzystywany jest komputer klasy PC z systemem operacyjnym Microsoft Windows (XP/7/10).

### 1.2. Oprogramowanie

Na komputerach zainstalowane jest środowisko programistyczne Microsoft Visual Studio 2008 Standard Edition lub Microsoft Visual Studio 2008 Express Edition zawierające kompilator Microsoft Visual C++ 2008.

## 2. Wiadomości teoretyczne

### 2.1. Program uruchomieniowy

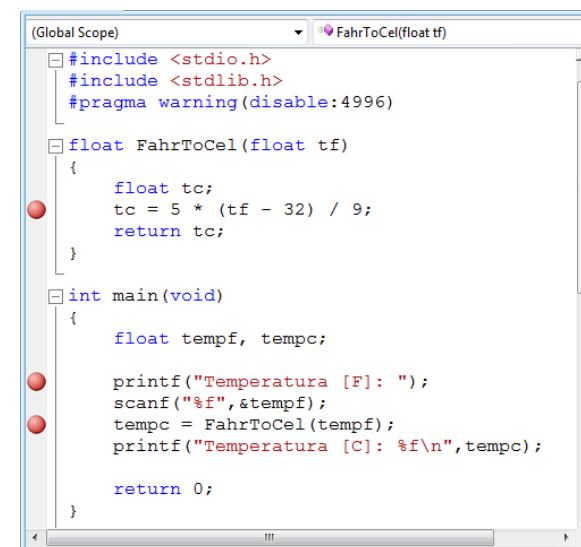
Podczas kompilacji programu w języku C kompilator sygnalizuje wszystkie odnalezione błędy składniowe. Niestety w programach mogą występować także błędy logiczne. W takim przypadku program kompiluje się, ale nie działa poprawnie. Do wykrycia i poprawienia tego typu błędów stosowany jest program uruchomieniowy (ang. *debugger*), czasami nazywany „odpluskwiaczem”.

Program uruchomieniowy umożliwia m.in.

- obserwowanie zachowania się programu podczas jego wykonania;
- krokowe wykonanie programu;
- ustalenie punktów kontrolnych;
- przerwanie lub wstrzymanie wykonania programu w celu sprawdzenia kodu;
- sprawdzenie i modyfikację wartości zmiennych występujących w programie;
- sprawdzenie zawartości rejestrów procesora;
- odczytanie zawartości pamięci zajmowanej przez program;
- wyświetlenie listy odwołań do funkcji.

### 2.2. Punkty kontrolne

Punkty kontrolne (punkty przerwania, punkty wstrzymania, ang. *breakpoints*) pozwalają na przerwanie wykonania programu w określonych miejscach i sprawdzenie jego aktualnego stanu (wartości zmiennych, zawartość rejestrów procesora, itp.). W środowisku Microsoft Visual Studio 2008 wiersze kodu programu, w których zostały ustawione punkty kontrolne, oznaczane są symbolem czerwonej kropki umieszczonej na lewym marginesie okna z kodem źródłowym (Rys. 1).

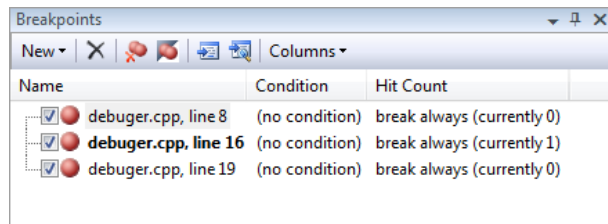


Rys. 1. Punkty kontrolne ustawione w kodzie programu

Punkt kontrolny może być ustawiony na kilka sposobów:

- kliknięcie lewym przyciskiem myszki na lewym marginesie okna z kodem programu;
- kliknięcie prawym przyciskiem myszki na wybranym wierszu kodu i wybranie z menu podręcznego opcji **Breakpoint** → **Insert Breakpoint**;
- ustawienie kursora w wybranym wierszu kodu i wybranie z menu głównego **Debug** → **Toggle Breakpoint (F9)**.

Podczas śledzenia wykonania programu wszystkie ustawione punkty kontrolne wyświetlane są w oknie Breakpoints (Rys. 2).



Rys. 2. Lista ustawionych punktów kontrolnych

Do usunięcia punktu kontrolnego można zastosować jeden z poniższych sposobów:

- kliknięcie lewym przyciskiem myszki na symbolu czerwonej kropki umieszczonym na lewym marginesie okna z kodem programu;
- kliknięcie prawym przyciskiem myszki na wierszu kodu z ustawionym punktem kontrolnym i wybranie z menu podręcznego **Breakpoint → Delete Breakpoint**;
- ustawienie kursora w wierszu kodu z ustawionym punktem kontrolnym i wybranie z menu głównego **Debug → Toggle Breakpoint (F9)**.

Można usunąć także wszystkie punkty kontrolne. W takim przypadku należy z menu głównego wybrać **Debug → Delete All Breakpoints (Ctrl + Shift + F9)**.

### 2.3. Śledzenie wykonania programu

Rozpoczęcie śledzenia wykonania programu może nastąpić w następujący sposób:

- wybranie z menu głównego polecenia **Debug → Start Debugging (F5)** - program zostanie uruchomiony i wykona się do pierwszego punktu kontrolnego;
- wybranie z menu głównego polecenia **Debug → Step Into (F11)** - program zostanie uruchomiony i zatrzyma się na pierwszej linii kodu funkcji **main()**;

- wybranie z menu głównego polecenia **Debug → Step Over (F10)** - program zostanie uruchomiony i zatrzyma się na pierwszej linii kodu funkcji **main()**;
- kliknięcie prawym przyciskiem myszki na wybranym wierszu kodu i wybranie z menu podręcznego **Run To Cursor** - program zostanie uruchomiony i wykona się do wybranego wiersza kodu lub do pierwszego punktu kontrolnego (zależnie od tego, co wystąpi wcześniej).

Kolejną linię kodu, która będzie wykonana, pokazuje żółta strzałka umieszczona na lewym marginesie okna. Dalsze śledzenie wykonania programu możliwe jest po wybraniu z menu głównego jednego z następujących poleceń:

- **Debug → Step Into (F11)** - wykonuje jedną linię kodu programu wchodząc do funkcji (zatrzymuje się na pierwszej linijce kodu funkcji);
- **Debug → Step Over (F10)** - wykonuje jedną linię kodu programu nie wchodząc do funkcji (wykonuje całą funkcję od razu);
- **Debug → Step Out (Shift + F11)** - wykonuje wszystkie instrukcje do końca funkcji i powraca do miejsca jej wywołania;
- **Debug → Continue (F5)** - wykonuje program do kolejnego punktu kontrolnego.

Śledzenie wykonania programu może być przerwane w dowolnym momencie. Służy do tego polecenie z menu głównego **Debug → Stop Debugging (Shift + F5)**. Można także przerwać śledzenie wykonania i rozpocząć je ponownie - polecenie **Debug → Restart (Ctrl + Shift + F5)**.

### 2.4. Obserwowanie wartości zmiennych

W środowisku Microsoft Visual Studio 2008 istnieje kilka metod obserwowania wartości zmiennych podczas śledzenia wykonania programu. Najprostsza metoda to zastosowanie **DataTips**. Jeśli podczas śledzenia wykonania programu najedziemy kursorem myszki nad nazwą dowolnej zmiennej, to zostanie wyświetlone małe okienko zawierające jej nazwę i aktualną wartość (Rys. 3). Operacja taka możliwa jest tylko w stosunku do zmiennych będących w aktualnym zasięgu widoczności. Istnieje możliwość zmiany wartości zmiennej.

W tym celu należy kliknąć prawym przyciskiem myszki na małym okienku i z menu podręcznego wybrać **Edit Value**.

```
int main(void)
{
    float tempf, tempc;

    printf("Temperatura [F]: ");
    scanf("%f",&tempf);
    tempc = FahrToCel(tempf);
    printf("Temperatura [C]: %f\n", tempc);

    return 0;
}
```

Rys. 3. Zastosowanie **DataTips** do wyświetlenia wartości zmiennej

Jeśli analizowana zmienna jest tablicą lub strukturą, to lista jej wartości zostanie wyświetlona po naciśnięciu znaku + umieszczonego przed nazwą zmiennej (Rys. 4).

```
int main(void)
{
    float tempf, tempc;
    int tab[5] = {4,3,2};

    printf("Temperatura [F]: ");
    scanf("%f",&tempf);
    tempc = FahrToCel(tempf);
    printf("Temperatura [C]: %f\n", tempc);

    return 0;
}
```

Rys. 4. Wyświetlenie elementów tablicy w **DataTips**

Program uruchomieniowy zawiera kilka typów okien, w których mogą być wyświetlane, obliczane i edytowane wartości zmiennych lub wyrażeń arytmetycznych. Każde z tych okien zawiera trzy kolumny:

- **Name** - nazwa zmiennej lub zapis wyrażenia arytmetycznego;
- **Value** - wartość zmiennej lub obliczonego wyrażenia arytmetycznego;
- **Type** - typ zmiennej lub typ wyniku wartości wyrażenia arytmetycznego.

Dostęp do okien możliwy jest z menu głównego po wybraniu **Debug → Windows**.

Do podstawowych okien należą:

- **Locals** - wyświetla wartości zmiennych występujących w aktualnym zakresie widoczności (np. zakres funkcji) (Rys. 5);

| Name  | Value     | Type  |
|-------|-----------|-------|
| tempc | 25.555555 | float |
| tempf | 78.000000 | float |

Rys. 5. Okno **Locals**

- **Autos** - wyświetla wartości zmiennych używanych w bieżącej i poprzednich liniach kodu (Rys. 6);

| Name  | Value     | Type  |
|-------|-----------|-------|
| tempc | 25.555555 | float |
| tempf | 78.000000 | float |

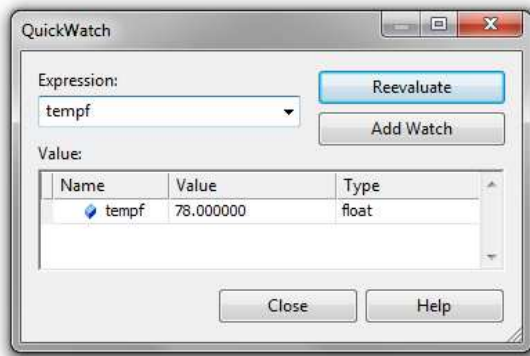
Rys. 6. Okno **Autos**

- **Watch** - wyświetla wartości zmiennych oraz wyrażeń wprowadzanych przez użytkownika (Rys. 7);

| Name           | Value                | Type   |
|----------------|----------------------|--------|
| tempc          | 25.555555            | float  |
| tempf          | 78.000000            | float  |
| 5/9*(tempf-32) | 0.000000000000000000 | double |
| (tempf-32)*5/9 | 25.555555555555557   | double |

Rys. 7. Okno **Watch**

- **QuickWatch** - wyświetla wartość tylko jednej zmiennej lub wyrażenia wprowadzanego przez użytkownika (Rys. 8).



Rys. 8. Okno **QuickWatch**

### 3. Przebieg ćwiczenia

Na pracowni specjalistycznej należy wykonać wybrane zadania wskazane przez prowadzącego zajęcia. W różnych grupach mogą być wykonywane różne zadania.

1. Dla programu wskazanego przez prowadzącego:
  - prześledź wykonanie programu instrukcja po instrukcji, bez wchodzenia do funkcji;
  - prześledź wykonanie programu instrukcja po instrukcji, z wchodzeniem do funkcji;
  - ustaw kilka punktów kontrolnych i prześledź wykonanie programu korzystając z tych punktów; usuń wszystkie punkty kontrolne.
  - ustaw kursor w dowolnej linii kodu programu i wykonaj program do tego miejsca; prześledź wykonanie dalszej części programu instrukcja po instrukcji, bez wchodzenia do funkcji.
2. Dla programu wskazanego przez prowadzącego prześledź jego wykonanie instrukcja po instrukcji. Jednocześnie zaobserwuj zmiany wartości zmiennych

występujących w programie (zastosuj **DataTip**, okno **Locals**, okno **Autos**, okno **Watch**, okno **QuickWatch**).

3. Programy wskazane przez prowadzącego zawierają błędy logiczne (wyświetlają niepoprawne wyniki). Wykorzystując program uruchomieniowy znajdź te błędy i je popraw.

### 4. Literatura

- [1] Kernighan B.W., Ritchie D.M.: Język ANSI C. Programowanie. Wydanie II. Helion, Gliwice, 2010.
- [2] Prata S.: Język C. Szkoła programowania. Wydanie VI. Helion, Gliwice, 2016.
- [3] Prinz P., Crawford T.: Język C w pigułce. APN Promise, Warszawa, 2016.
- [4] King K.N.: Język C. Nowoczesne programowanie. Wydanie II. Helion, Gliwice, 2011.
- [5] Kochan S.G.: Język C. Kompendium wiedzy. Wydanie IV. Helion, Gliwice, 2015.
- [6] Wileczek R.: Microsoft Visual C++ 2008. Tworzenie aplikacji dla Windows. Helion, Gliwice, 2009.

### 5. Pytania kontrolne

1. Wyjaśnij, do czego przeznaczony jest program uruchomieniowy (*debugger*).
2. Omów sposoby śledzenia wykonania programu dostępne w środowisku Microsoft Visual Studio 2008.
3. Co to są punkty kontrolne (*breakpoints*) i w jaki sposób można je ustawiać?
4. Wyjaśnij, w jaki sposób można śledzić wartości zmiennych występujących w programie.

## 6. Wymagania BHP

Warunkiem przystąpienia do praktycznej realizacji ćwiczenia jest zapoznanie się z instrukcją BHP i instrukcją przeciw pożarową oraz przestrzeganie zasad w nich zawartych.

W trakcie zajęć laboratoryjnych należy przestrzegać następujących zasad.

- Sprawdzić, czy urządzenia dostępne na stanowisku laboratoryjnym są w stanie kompletnym, nie wskazującym na fizyczne uszkodzenie.
- Jeżeli istnieje taka możliwość, należy dostosować warunki stanowiska do własnych potrzeb, ze względu na ergonomię. Monitor komputera ustawić w sposób zapewniający stałą i wygodną obserwację dla wszystkich członków zespołu.
- Sprawdzić prawidłowość połączeń urządzeń.
- Załączenie komputera może nastąpić po wyrażeniu zgody przez prowadzącego.
- W trakcie pracy z komputerem zabronione jest spożywanie posiłków i picie napojów.
- W przypadku zakończenia pracy należy zakończyć sesję przez wydanie polecenia wylogowania. Zamknięcie systemu operacyjnego może się odbywać tylko na wyraźne polecenie prowadzącego.
- Zabronione jest dokonywanie jakichkolwiek przełączeń oraz wymiana elementów składowych stanowiska.
- Zabroniona jest zmiana konfiguracji komputera, w tym systemu operacyjnego i programów użytkowych, która nie wynika z programu zajęć i nie jest wykonywana w porozumieniu z prowadzącym zajęcia.
- W przypadku zaniku napięcia zasilającego należy niezwłocznie wyłączyć wszystkie urządzenia.
- Stwierdzone wszelkie braki w wyposażeniu stanowiska oraz nieprawidłowości w funkcjonowaniu sprzętu należy przekazywać prowadzącemu zajęcia.

- Zabrania się samodzielnego włączania, manipulowania i korzystania z urządzeń nie należących do danego ćwiczenia.
- W przypadku wystąpienia porażenia prądem elektrycznym należy niezwłocznie wyłączyć zasilanie stanowiska. Przed odłączeniem napięcia nie dotykać porażonego.