

**Otwarto:** poniedziałek, 26 lutego 2024, 00:00

**Wymagane do:** niedziela, 17 marca 2024, 23:59

## Środowisko pracy

W tym ćwiczeniu zgłębimy kluczowe narzędzia i praktyki niezbędne do efektywnego tworzenia, budowania i debugowania aplikacji w języku C.

Pliki Makefile są nieodłącznym elementem procesu kompilacji w środowiskach Unix-like. Pozwalają one na zdefiniowanie zestawu instrukcji, które są potrzebne do budowania aplikacji, włączając w to kompilację źródeł, linkowanie bibliotek oraz zarządzanie zależnościami między plikami. Podczas tego ćwiczenia dowiemy się, jak tworzyć i zarządzać plikami Makefile, co pozwoli nam na zautomatyzowanie procesu kompilacji i ułatwi pracę nad projektami programistycznymi.

Środowiska programistyczne, zwłaszcza IDE (Integrated Development Environment), są niezastąpionymi narzędziami dla każdego programisty. W ćwiczeniu przyjrzymy się popularnym IDE dla języka C, jak również poznajemy ich funkcje i możliwości, które pomogą nam w efektywnym tworzeniu, debugowaniu i testowaniu aplikacji.

Techniki debugowania są nieodzowne w procesie rozwoju oprogramowania, zwłaszcza gdy napotykamy na błędy i trudności w naszym kodzie. W trakcie tego ćwiczenia zgłębimy różne metody debugowania w języku C, w tym korzystanie z debuggera, wstawianie punktów kontrolnych oraz analizowanie zmiennych działającego programu.

## Zadanie

1. Zapoznaj się z koncepcją plików Makefile: [https://www.gnu.org/software/make/manual/html\\_node/Introduction.html](https://www.gnu.org/software/make/manual/html_node/Introduction.html)
2. Zainstaluj/skonfiguruj IDE, w którym będziesz pracować przez resztę laboratoriów. (np. VS Code, Vim, etc.)
3. Stwórz nowy projekt w IDE.
4. Napisz prosty program countdown.c, który będzie w pętli odliczał od 10 do 0 i wypisywał aktualną liczbę na konsolę (każda liczba w nowej linii).
5. Stwórz plik Makefile, za pomocą którego skompilujesz swój program.  
Makefile powinien zawierać co najmniej trzy targety: all, countdown, clean.

all — powinien budować wszystkie targety (na razie tylko countdown, w kolejnych zadaniach będziemy dodawać nowe targety).

countdown — buduje program countdown.c

clean — usuwa wszystkie pliki binarne, czyści stan projektu

Użyj zmiennych CC oraz CFLAGS do zdefiniowania kompilatora (gcc) i flag kompilacji (-Wall, -std=c17, ...).

Dodatkowo w Makefile powinien być specjalny target .PHONY.

6. Skompiluj i uruchom program.
7. Korzystając z gdb (lub lldb) zademonstruj poniższe:
  1. zatrzymywanie programu (breakpoint) wewnątrz pętli
  2. podejrzenie aktualnego indeks pętli
  3. kontynuację wykonywania programu
  4. wypisanie kolejnego indeksu
  5. usunięcie breakpoint-a
  6. kontynuowanie działania programu do końca

<https://sourceware.org/gdb/current/onlinedocs/gdb.html/>

<https://sourceware.org/gdb/current/onlinedocs/gdb.html/Sample-Session.html#Sample-Session>

<https://lldb.lvm.org/use/tutorial.html>

8. Skonfiguruj swoje IDE do współpracy z Makefile. Postaw breakpoint (graficznie, klikając) w środku pętli. Uruchom program w trybie Debug i zademonstruj wszystkie podpunkty z pkt. 7.