

# Programowanie systemowe w języku Go

## *Linux w Go dla zaawansowanych*

---

### Cele szkolenia

Celem szkolenia jest zapoznanie uczestników z aspektami programowania systemowego w języku Go.

### Umiejętności

Dzięki szkoleniu uczestnik będzie:

- Rozumieć wewnętrzne mechanizmy Go, działanie kompilatora, garbage collector, modele pamięci
- Umieć pisać wydajny, niskopoziomowy kod w Go na Linuxie
- Kontrolować procesy i demony z poziomu Go
- Komunikować się między procesami w Go
- Używać różnych formatów kodowania do serializacji złożonych struktur danych
- Komunikować się przez sieć z innymi aplikacjami

### Profil uczestników

Szkolenie przeznaczone jest dla programistów, projektantów i architektów oprogramowania, którzy mają już znać podstawy składni języka Go i chcieliby pogłębić swoją wiedzę związaną z programowaniem systemowym w tym języku. Wymagana jest podstawowa znajomość architektury systemu Linux.

### Przygotowanie uczestników

Szkolenie przeznaczone jest dla osób, które umieją już programować w Go i mają doświadczenie w kompilacji, uruchamianiu i debugowaniu kodu w tym języku. Wymagana jest umiejętność pracy w systemie Linux. Wcześniejsza instalacja Go oraz VS Code i plugina do Go jest mile widziana, ale nie wymagana. Szkolenie prowadzone będzie przy użyciu VS Code, GoLand oraz środowiska Go Playground na systemie Linux.

### Szczegółowy program szkolenia

1. *Wprowadzenie problematyki programowania systemowego*
  - 1.1. *Programowanie systemowe a aplikacyjne*
  - 1.2. *Linux - podstawy architektury systemu: system calls: kontrola procesów, zarządzanie plikami, zarządzanie urządzeniami, komunikacja, zarządzanie pamięcią, zarządzanie uprawnieniami*
2. *Wnętrza Go*
  - 2.1. *Kompilator Go: go-build, skaner/parser/SSA (static single assignment), linker, metadane, bootstrap, assembler Go*

- 2.2. *Garbage collector, algorytm tricolor, model pamięci*
- 3. *Operacje we/wy*
  - 3.1. *Praca z systemem plików, pakiet os, prawa dostępu, pakiety io i bufio, pakiet syscall*
  - 3.2. *Strumienie, pakiet bytes, pakiet flag*
  - 3.3. *Pseudoterminale*
  - 3.4. *Praca z plikami systemu operacyjnego*
- 4. *Komunikacja między procesami*
  - 4.1. *Podstawy: procesy, procesy potomne, demony, usługi, pakiet flag*
  - 4.2. *Sygnały, pakiety signal*
  - 4.3. *Pipes, anonimowe pipes, , standardowe pipes we/wy*
  - 4.4. *Kody wyjścia, kody wyjścia i deferred functions, kody wyjścia i goroutines*
- 5. *Reflections - podstawowe mechanizmy, obsługa złożonych typów, refleksje na refleksje*
- 6. *Pakiet unsafe*
- 7. *Pakiet CGO*
  - 7.1. *Kod C w Go; kod Go w C*
  - 7.2. *Typy w C i Go*
  - 7.3. *Praca ze strukturami: ręczny padding, packed i unpacked w C*
  - 7.4. *Dobre praktyki CGO*
- 8. *Kodowanie danych w Go*
  - 8.1. *Kodowania tekstowe - CSV, JSON, XML*
  - 8.2. *Kodowania binarne - gob, protobuf, BSON*
- 9. *Komunikacja sieciowa w Go*
  - 9.1. *Podstawy: TCP/IP, UDP, podstawowe narzędzia*
  - 9.2. *Pakiet net*
  - 9.3. *Sockets i RPC w Go*
  - 9.4. *Serwery i klienci: TCP, UDP, RCP*
  - 9.5. *net/http, serwer i klient web*

## Metoda realizacji szkolenia

Szkolenie ma formę następujących po sobie wykładów i ćwiczeń praktycznych, wykonywanych indywidualnie lub w ramach programowania w parach (pair programming).

## Liczba dni, liczba godzin

2 dni po 8 godzin lekcyjnych

## Ścieżka rozwoju

*Docker DevOps Workshop,*

*Kubernetes – Wprowadzenie*

*Docker SysOps Workshop*

*Docker Extensive Workshop 2*