BOTTEGA IT minds

Program szkolenia:

Programowanie obiektowe i wielowątkowe w C na systemie Linux

Informacje:

Nazwa:	**Programowanie obiektowe i wielowątkowe w C na systemie Linux**
Kod:	**ccpp-multi**
Kategoria:	C i C++
Odbiorcy:	developerzy
Czas trwania:	3 dni
Forma:	35% wykłady / 65% warsztaty

Szkolenie prezentuje faktyczną możliwość stosowania paradygmatu obiektowego podczas implementacji kooperujących aplikacji wieloprocesowych i wielowątkowych. Wszystkie techniki przedstawiane są na przykładach z rzeczywistych projektów.

W opozycji do akademickiego/książkowego podejścia, uczestnicy szkolenia zdobywaną stopniowo wiedzę jednocześnie wykorzystują i testują przy okazji implementacji zadań warsztatowych, które odzwierciedlają wyzwania napotykane w rzeczywistości, tworząc przy tym skalowalny kod, otwarty na zmiany i podatny na testowanie.

Zalety szkolenia:

- Skupienie uwagi na "fizyce" problemu programowania współbieżnego i rozproszonego
- Wybór jedynie użytecznych wzorców i technik
- Rzeczywiste przykłady

BOTTEGA IT minds

Szczegółowy program:

1. OOP in C

1.1. Powrót do podstaw składni języka C

1.1.1. Koncept wartości i wskaźnika

1.1.2. Rzutowanie po raz pierwszy

1.1.3. Aliasing wskaźników

1.1.4. Reguła „strict aliasing'u”

1.1.5. Jak nad tym panować? Rozszerzenie rzutowania

1.1.6. Słowa kluczowe: const, static, volatile i restrict

1.2. Wykorzystanie składni języka i części paradygmatu OOP na rzecz bezpiecznej implementacji re-używalnych obiektów

1.2.1. Abstrakcja – projektowanie interfejsu

1.2.2. Enkapsulacja – ukrywanie szczegółów implementacyjnych i zapobieganie nieoczekiwanym zmianom stanu

1.2.3. Polimorfizm – specjalizowanie zachowania

1.2.4. SOLID-ne programowanie z wykorzystaniem GRASP i DDD

1.3. Inwersja kontroli

1.3.1. Paradygmat IoC i zasada Hollywood

1.3.2. Wstrzykiwanie zależności vs. podejście tradycyjne

2. Programowanie współbieżne

2.1. Wstęp

2.1.1. Po co zrównoleglać i rozpraszać?

2.1.2. Problemy i komplikacje

2.2. Podstawowa nomenklatura

2.2.1. Aplikacja, program, proces wątek

BO·TT·EGA
IT minds

2.2.1.1. Poprawne rozumienie podstawowych idiomów w kontekście architektury Linux

2.2.2. Synchronizacja

2.2.2.1. Mechanizmy blokad, oczekiwania i notyfikacji

2.2.2.2. Poprawne projektowanie sekcji krytycznych

2.3. Programowanie współbieżne pod maską

2.3.1. Prawa Amdahla i Gustafsona

2.3.2. Konfrontacja prawideł matematycznych, a szara rzeczywistość

2.3.3. C językiem niskopoziomowym

2.3.4. POSIX vs. reszta świata

2.3.5. Narzędzia wspomagające

3. Testy deweloperskie

3.1. Testy jednostkowe vs. testy modułowe

3.1.1. Które i jak stosować

3.1.2. Biblioteki usprawniające testowanie

3.2. Stress testy

3.3. Testy funkcjonalne

3.4. Diagnostyka aplikacji wielowątkowych: śledzenie i profilowanie

3.4.1. Zestaw narzędzi wspomagających

3.4.2. Poprawna interpretacja wyników