Tytuł szkolenia: Programowanie współbieżne w języku C na system UNIX

Kod szkolenia: C-IPC

Wprowadzenie

Trudno obecnie o nietrywialną aplikację nie korzystającą w żaden sposób ze współbieżności. Nietrywialne aplikacje systemowe, jak i wszelkie rozwiązania na styku aplikacji z systemem operacyjnym (sterowniki, biblioteki) wymagają uwzględnienia interakcji między poszczególnymi procesami i odpowiedniej ich synchronizacji. Wiele aplikacji, nie wyłączając zwykłych aplikacji użytkowych, dla wygody i dla wykorzystania w pełni mocy wieloprocesorowych maszyn projektuje się z kolei jako aplikacje wielowątkowe. Naturalne jest też, że programy komunikują się miedzy sobą przez sieć albo używając innych mechanizmów. Dla wszystkich tych zastosowań istnieją pewne zasady poprawnego programowania, które przenoszą się na dowolne platformy i technologie.

System UNIX był od początku projektowany jako system wieloprocesowy i jest pierwszym szeroko stosowanym systemem tego typu. Obecnie wiele systemów opartych o UNIX-a, w tym Linux, pozwala korzystać z ukształtowanych przez lata rozwoju tej rodziny systemów mechanizmów komunikacji i synchronizacji międzyprocesowej (IPC – *inter-process communication*). Wątki w systemie UNIX i pochodnych pojawiły się później. Spośród dostępnych bibliotek pozwalających korzystać z wątków najbardziej popularną dla czystego języka C jest pthreads.

Adresaci szkolenia

Adresatami szkolenia są programiści pragnący poznać mechanizmy systemu UNIX (i pochodnych, np. Linux) służące do realizacji wieloprocesowości i wielowątkowości.

Szkolenie polecamy przede wszystkim programistom przygotowującym się do pracy w projektach wymagających skorzystania z technologii będących przedmiotem szkolenia. Może ono być także kolejnym krokiem w naturalnym rozwoju programisty języka C platformy UNIX, który pozwoli lepiej rozumieć i w większym stopniu wykorzystywać możliwości systemu (polecamy w tym miejscu cykl szkoleń **C-PD** → **C-IPC** → **C-TCP**).

Technologie prezentowane na szkoleniu można potraktować dosłownie i korzystać z nich bezpośrednio w pracy pod systemem UNIX/Linux lub tylko jako przykładową realizację mechanizmów IPC, które w innych systemach (np. Windows, Android) są podobne w idei, ale różne jeśli chodzi o szczegóły i API.

Wymagania wstępne:

- programowanie w języku C (np. dzięki szkoleniu **C-PD**) lub ewentualnie w C++.

Wiedza i umiejętności, które mogą podnieść efektywność tego szkolenia:

- znajomość podstawowych pojęć i zasad programowania współbieżnego,
- znajomość mechanizmów IPC lub wielowątkowości z innych języków i platform.

Cel szkolenia

Szkolenie, prowadzone na platformie Linux, obejmuje te mechanizmy programowania współbieżnego, które są obecnie uważane za standardowe i są przenośne między różnymi wersjami systemów opartych o UNIX. Gdzie możliwe, opieramy się o standard POSIX. Omawiane są także podstawowe zasady poprawnego programowania współbieżnego, a także typowe błędy i zagrożenia. Prezentowane jest podstawowe API mechanizmów, dostępne z języka C i korzystające bezpośrednio z odpowiednich zasobów i funkcji systemowych. Używanie ich w ten sposób daje największą przenośność rozwiązań i największą kontrolę nad szczegółami.

Po zakończeniu szkolenia aktywny uczestnik potrafi:

- tworzyć i zamykać procesy, a także uruchamiać na różne sposoby nowe programy,
- tworzyć procedury obsługi sygnałów oraz maskować sygnały,
- współbieżnie korzystać z plików używając takich mechanizmów jak blokowanie plików
- czy mapowanie plików do pamięci,
- tworzyć proste rozwiązania wieloprocesowe w modelu rozproszonym oparte o
- komunikację za pomocą łącz nazwanych i nienazwanych,
- tworzyć proste rozwiązania wieloprocesowe w modelu scentralizowanym oparte o
- współdzielenie pamięci i synchronizację za pomocą semaforów,
- korzystać z kolejek komunikatów,
- tworzyć proste aplikacje wielowątkowe w oparciu o bibliotekę pthreads,
- wykorzystywać do synchronizacji między wątkami mutexy i zmienne warunkowe.

Szkolenie nie obejmuje m.in. następujących tematów:

- komunikacja sieciowa (patrz szkolenie **C-TCP**),
- analogiczne mechanizmy w innych systemach operacyjnych, w szczególności Windows,
- opakowanie poznawanych tu mechanizmów systemowych w nakładki języka C++.

Czas i forma szkolenia

- 21 godzin (3 dni x 7 godzin), w tym wykłady i warsztaty praktyczne.

Plan szkolenia

1. Procesy w systemie UNIX.
    a. Tworzenie nowych procesów: funkcja fork, hierarchia procesów.
    b. Kończenie procesów, funkcja wait, procesy *zombie*.
    c. Ładowanie nowych programów, grupa funkcji exec_.
2. Sygnały.
    a. Wysyłanie sygnałów.
    b. Rejestrowanie kodu obsługi sygnału.
    c. Maskowanie sygnałów.
3. Komunikacja między procesami za pomoczą plików.
    a. Deskryptory plików a funkcje fork i exec.
    b. Oczekiwanie na zdarzenia wejścia/wyjścia – funkcje select i poll.
    c. Blokowanie dostępu do pliku (do całości, do części).
    d. Mapowanie plików do pamięci.
4. Komunikacja między procesami za pomoczą łącz.
    a. Łącza nienazwane („pipe”).
    b. Duplikowanie otwartych deskryptorów i podmiana standardowego wejścia i wyjścia.
    c. Łącza nazwane („fifo”).
5. Synchronizacja między procesami za pomocą semaforów.
    a. Definicja teoretyczna i działanie semaforów.
    b. Semafory standardu POSIX.
    c. Zastosowanie semaforów do synchronizacji między procesami.
6. Mechanizmy IPC Systemu V.
    a. Wspólne podstawy.
    b. Tablice semaforów („sem”).
    c. Segmenty pamięci dzielonej („shm”).
    d. Kolejki komunikatów („msg”).
7. Tworzenie aplikacji wielowątkowych za pomocą biblioteki pthreads.
    a. Uruchamianie wątków.
    b. Kończenie wątków, wątki detached i joinable.
    c. Anulowanie wątków, wstrzymywanie anulowania i stos bloków czyszczących.
    d. Synchronizacja wątków za pomocą mutexów.
    e. Synchronizacja wątków za pomocą zmiennych warunkowych.