Systemy Operacyjne 2

Projekt - 2025

Damian Raczkowski

- DevOps inżynier w Intel
- Wcześniej stażysta DevOps w CERN
- Absolwent kierunku ITE na specjalności IGM zarówno na I jak i II stopniu.
- Prywatnie szachista, nerd, amator gotowania

Plan prezentacji

- Projekt(y)
- Wymagania
- Plan pracy
- Kryteria oceniania
- Konsultacje

Projekt(y)

Celem zajęć jest wykonanie aplikacji wykorzystującej wielowątkowość oraz wykorzystanie mechanizmów zabezpieczających sekcje krytyczne: mutexy, semafory.

W ramach kursu będą do wykonania dwa projekty:

- Problem jedzących filozofów (jako rozgrzewka, daje wam ocenę max 3.5 z kursu) - solo
- Wielowątkowy serwer czatu (dla tych którzy chcą wyższą ocenę 4.0-5.0) - solo bądź w parach.
    - Dopuszczam też inne tematy po wcześniejszej konsultacji

Amir Shevat

@ashevat

A programmer had a problem. He thought to himself, "I know, I'll solve it with threads!". has Now problems. two he

10:00 PM · 16 Jan 21 · Twitter for iPhone

Wymagania

- Język programowania: C/C++, w ostateczności Python (dopuszczam tylko w drugim projekcie)
  - **Nie możecie** korzystać z modułów które mają **gotową synchronizację**! Ideą projektu jest to, że zaimplementujecie wątki i sekcje krytyczne **sami**.
  - W przypadku C++ zostawiam dowolność czy skorzystacie z pthreads czy std::thread
  - W przypadku Pythona musicie skorzystać z modułu threading (ignorujemy GIL).
  - Kod i komentarze muszą być w języku angielskim.
- Dokumentacja tego co zrobicie w postaci README.md na repozytorium
  - Maks 3 strony A4
  - Instrukcje uruchomienia projektów
  - Opis problemu/ów
  - Wylistowanie:
    - Wątków i co reprezentują
    - Sekcje krytyczne i ich rozwiązanie

Wymagania - projekty

Problem jedzących filozofów

- Każdy z wątków raportuje o stanie w którym się znajduje (Wystarczy wydruk w konsoli)
- W programie nie dochodzi do trwałego zablokowania wątków (ang. deadlock)
- Program otrzymuje jako argument liczbę filozofów.

Wielowątkowy serwer czatu

- Serwer tworzy osobny wątek dla każdego połączenia od klienta
- Serwer dba o synchronizację wiadomości od klientów
- Klient widzi wiadomości w chacie
- Klient ma możliwość wysyłania wiadomości

Tyczące się obu projektów

- Wydruk w konsoli musi być czytelny

Wymagania - projekty

W przypadku chęci realizacji własnego pomysłu na drugi projekt oczekuję następujących rzeczy:

- Krótko przedstawić problem jaki chcecie rozwiązać
- Wskazać miejsce gdzie będzie występował race condition
- Wizualizację działania projektu

Plan pracy

1. Wysłanie linku do repozytorium. (czas **1 tydzień** od dzisiaj) Formularz
2. Stworzenie działającego rozwiązania 1 projektu (**3 zajęcia**)
3. Stworzenie działającego rozwiązania 2 projektu (**7 zajęcia**) oraz wystawienie oceny końcowej z kursu.

Na punktach kontrolnych (3 i 7 zajęcia) obecność **obowiązkowa**. Pozostałe terminy służą jako możliwość konsultacji projektu.

W przypadku, kiedy student odda projekt szybciej, **nie musi** być obecnym na punkcie kontrolnym.

Kod wysyłamy do **24h** przed spotkaniem kontrolnym.

Opóźnienia naliczane są **po zajęciach kontrolnych**. Opóźnienie = **0.5** oceny w **dół**

Kryteria oceniania

Kryteria oceniania

- Terminowość: Wysłanie zadań w terminie, Obecność na punktach kontrolnych.
- Czytelność kodu i dokumentacji: Czy kod da się zrozumieć? Czy dokumentacja jest czytelna i spełnia swoje zadanie?
- Czy student zna swój kod :P
- Historia commitów
- Kod działa: sekcje krytyczne zostały pokryte.
- Bonusy działające na **korzyść** studenta:
  - kod spełnia wymagania lintera (pylinta/cpplinta)
  - Automatyczna analiza kodu na repozytorium
  - Dodatkowe feature'y w drugim projekcie (np zamiast linii komend jest interfejs graficzny)
  - Automatyzacja budowy projektu (np używanie Makefile'a)
  - I wiele innych


THE DEATH STAR FROM STAR TREK.

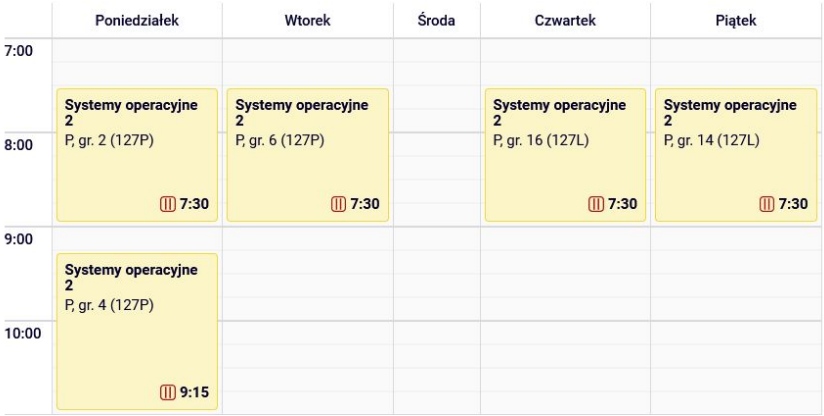
Konsultacje

- Zdalnie na serwerze Discord Link
- bądź w czasie zajęć nie będącymi terminem kontrolnym

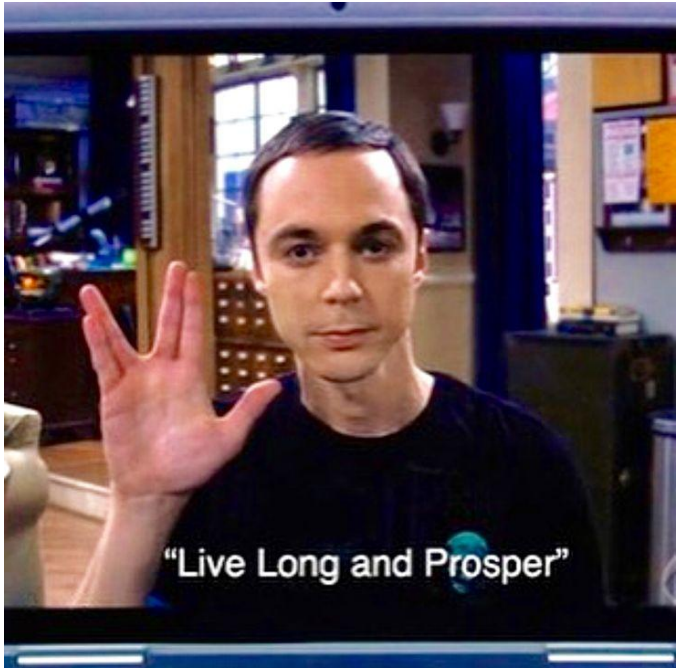
Potrzebę konsultacji (zdalnie bądź w terminie zajęć) zgłaszamy co najmniej **1 dzień przed** za pomocą maila.

Podczas pisania maila proszę o opis problemu jaki macie, to dacie mi szansę się przygotować :-)

Jeżeli problem jest pilny możecie się pojawić na zajęciach innej grupy. Ale uprzedzam, że studenci tej grupy będą mieli pierwszeństwo.



Pytania?


“Live Long and Prosper”