Mechanizmy synchronizacyjne: semafory całkowitoliczbowe/zliczające.

Lab Systemy Operacyjne

Jan Pielesiak

Semafor

Krótki opis

Semafor jest rozwiązaniem problemu ewentualnego współdzielenia przez różne procesy zasobów/plików, które nie mogą być współdzielone.

Semafor jest zmienną, która informuje, czy do danego elementu (pliku, zasobu) jest możliwy obecnie dostęp.

Semafor działanie

Proces chcący użyć np. pliku wywołuje funkcję wait(S) , gdzie S jest odpowiedzialnym za ten plik semaforem. Funkcja ta sprawdza, czy wartość semafora jest większa od zera. Jeśli tak, zmniejsza semafor i kończy się, pozwalając procesowi wykonywać swoje działania. Jeśli nie, blokuje proces dopóki wartość semafora nie zostanie powiększona powyżej zera.

Gdy proces zakończy działanie na pliku, podnosi wartość semafora funkcją signal(S), umożliwiając kolejnym procesom dostęp.

Semafor działanie

Ilość procesów mogących naraz korzystać z danego zasobu jest ustalona w momencie tworzenia semafora, poprzez ustawienie jego początkowej wartości.

Procesy oczekujące na semaforze są przechowywane w kolejce i budzone podczas operacji signal(S).

Przykład

Załóżmy, że mamy plik p1, który modyfikowany może być przez jeden program naraz. Tworzymy odpowiedzialny zań semafor S1, którego wartość początkową ustawiamy na 1.

Proces o1 chce sobie coś w p1 pozmieniać. Wywołuje wait(S1),jest przepuszczany by robić swoje a, wartość semafora spada do 0.

Wtem proces o2 także próbuje zacząć działanie na p1. Wywołuje wait(S1), lecz wartość semafora jest równa 0, więc proces ten czeka i dodany jest do kolejki oczekujących procesów.

Przykład

o1 kończy pracę na p1, wywołuje signal(S1), zwiększając wartość S1 do 1 oraz budząc znajdujący się na początku kolejki o2. Semafor zmniejszany jest przez o2 znów do 0 i o2 przechodzi do wykonywania operacji na p1.

Zarys implementacji

Klasa *Semafor* posiadająca:

zmienną typu **int** *wartosc*, która informuje o wartości semafora

kolejkę procesów *kolejka,* będącą najprawdopodobniej typu **ArrayDeque** i zawierającą odnośniki do procesów czekających pod semaforem (numer id procesu lub odnośnik do pozycji w której jest przechowywany proces)

Konstruktor umożliwiający ustalenie początkowej wartości zmiennej *wartosc*.

Zarys implementacji

Metody:

void wait(Semafor S) - zmniejsza wartość pola *wartosc*, jeśli jest mniejsza równa zeru wywołuje funkcję **block()** oraz dodaje proces do kolejki *kolejka*.

void signal(Semafor S) – podnosi wartość pola *wartosc*, jeśli jest większa od zera zdejmuje kolejny proces z kolejki *kolejka* i za pomocą funkcji **wakeup()** pozwala mu pracować.

void block() - zatrzymuje działanie procesu.

void wakeup() - wznawia działanie procesu.

Metody **wait** oraz **signal** są metodami publicznymi wywoływanymi przez procesy podczas próby dostępu do plików, a parametrem podczas tych wywołań będzie semafor odpowiedzialny za dany plik. Operacje **block()** oraz **wakeup()** są elementarnymi funkcjami systemowymi dostarczonymi przez osobę odpowiedzialną za obsługę działania plików i wywoływane są wewnątrz metod

Koncepcja interfejsu pracy krokowej

Podczas wywoływania funkcji **wait** wypisywana na ekran będzie wartość początkowa i końcowa semafora oraz zawartość kolejki, wraz z informacją, czy wykorzystano operację block().

Podczas wywoływania funkcji **signal** wypisywana będzie wartość początkowa i końcowa semafora oraz zawartość kolejki, oraz informacja czy wykorzystano wakeup().

Dziękuję za uwagę!