

Systemy operacyjne 2016

Lista zadań nr 5

Na zajęcia ?

Należy przygotować się do zajęć czytając następujące rozdziały książek:

- Stallings (wydanie siódme): 5.1 - 5.5, 6.1, 6.2
- Tanenbaum (wydanie czwarte): 2.3

UWAGA! W trakcie prezentacji rozwiązań należy zdefiniować i wyjaśnić pojęcia, które zostały oznaczone **wytłuszczoną** czcionką.

Zadanie 1. Podaj w pseudokodzie przykład szkodliwej **rywalizacji** procesów o dostęp do współdzielonych danych. Czemu odpluskwanie programów, w których występuje **sytuacja wyścigu** jest szczególnie uciążliwa?

Zadanie 2. Sformułuj problem **sekcji krytycznej**. Podaj i szczegółowo uzasadnij istotność warunków, jakie musi spełniać rozwiązanie tego problemu. Następnie wymień trzy sprzętowe mechanizmy umożliwiające implementację sekcji krytycznej. Podaj w pseudokodzie semantykę instrukcji atomowej **compare-and-swap** i pokaż jak użyć jej do implementacji **blokad wirujących**.

Zadanie 3. Jakie zadanie realizują semafony? Podaj różnice między następującymi parami semaforów: **binarny** i **zliczający**, **słaby** i **silny**. Czym różni się **mutex** od semafora binarnego? Czy system *Linux* implementuje słabe semafony? Podaj w pseudokodzie implementację semafora zliczającego, używając semaforów binarnych, i uzasadnij jej poprawność.

Zadanie 4. Rozważmy proces, który próbuje wejść pod opuszczony semafor. Ma dwie opcje: albo **aktywnie czeka** na podniesienie semafora, albo prosi jądro o **uśpienie** do czasu gdy stan semafora zostanie zmieniony. Porównaj te rozwiązania i powiedz w jakich warunkach się sprawdzają. Opisz rozwiązanie pośrednie, czyli **semafony adaptacyjne**. Czy mechanizm **futex** systemu *Linux* implementuje semafony adaptacyjne? Jaka jest semantyka operacji **FUTEX_WAIT** i **FUTEX_WAKE**?

Zadanie 5. Wymień cztery warunki konieczne do zaistnienia **zakleszczenia** (ang. *deadlock*). Czym różni się zakleszczenie od **ukrytego zakleszczenia** (ang. *livelock*) i **głodzenia** (ang. *starvation*)? W jaki sposób programista może przeciwdziałać zakleszczeniom?

Zadanie 6. Podaj sygnaturę i semantykę operacji **wait**, **signal** i **broadcast** dla **zmiennych warunkowych**. Co przechowuje obiekt zmiennej warunkowej? W jaki sposób **monitory** upraszczają programowanie współbieżne? Wytłumacz różnice między **monitorami Hoare'a**, a **monitorami Mesa**. Używając zmiennych warunkowych podaj w pseudokodzie implementację semafora zliczającego.

Zadanie 7. Mechanizm **przekazywania komunikatów** wymaga implementacji co najmniej dwóch funkcji: **send(dest,msg)** i **recv(src,msg)**. Wymień niezbędne składowe komunikatu. Jakie problemy stwarza synchronizacja z użyciem przekazywania komunikatów w stosunku do semaforów? Podaj różne warianty semantyki operacji **send** i **recv** dla **skrzynek pocztowych**. Jak adresować **nadawcę** lub **odbiorcę**? Czym charakteryzują się **punkty schadzek** (ang. *rendezvous*)?

Zadanie 8. Jedną z technik wymuszania spójności współdzielonych struktur danych bez stosowania blokad jest **RCU** (ang. *read-copy-update*). Działa ona przy założeniu, że strukturę danych może przeglądać wiele wątków czytających i aktualizować co najwyżej jeden piszący. Jakie dodatkowe założenia należy przyjąć w stosunku do wątków czytających? W jaki sposób wykorzystywany jest fakt oddzielenia **fazy usuwania** elementów od **fazy odzyskiwania** pamięci? Biorąc pod uwagę rozpatrzone ograniczenia podaj przykłady w jakich stosowanie tej techniki jest uzasadnione.