

# Systemy operacyjne

## Ćwiczenia 4

Należy przygotować się do zajęć czytając następujące rozdziały książek:

- Stallings: 5.1 - 5.5, 6.1, 6.2
- Tanenbaum: 2.3
- Silberschatz: 5.1, 5.2, 5.4 - 5.6, 5.8, 5.10, 7.1 - 7.4

### Zadanie 1

Podaj przykład (w pseudokodzie) szkodliwej **rywalizacji** procesów o dostęp do **współdzielonych** danych. Wykaż, że wynik modyfikacji tych danych może zależeć od arbitralnych decyzji planisty krótkoterminowego. Czym jest **sytuacja wyścigu**, a czym **zakleszczenie**?

### Zadanie 2

Sformułuj problem **sekcji krytycznej** i podaj warunki, jakie spełniać musi rozwiązanie tego problemu. Szczegółowo uzasadnij istotność każdego z powyższych warunków. Czym różni się **przetwarzanie równoległe** od **przetwarzania współbieżnego**? Czym charakteryzują się funkcje **wielobieżne**?

### Zadanie 3

Co to są **semafony** i jakie zadanie realizują? Wymień różnice między następującymi typami: **semafor binarny**, **semafor zliczający**, **słaby semafor**, **silny semafor**, **POSIX.1 mutex**. Podaj implementację (w pseudokodzie) semafora zliczającego używając semaforów binarnych i uzasadnij jej poprawność.

### Zadanie 3½ [nieobowiązkowe]

Podaj programowy sposób na implementację sekcji krytycznej – **algorytm Petersona** – i wyjaśnij jego działanie. Czemu taki sposób synchronizacji jest niepraktyczny nawet, jeśli użyjemy bardziej zaawansowanego **algorytmu piekarnianego** Lamporta? Spróbuj w przystępny sposób wyjaśnić pojęcie **bariery pamięciowej** i **sekwencyjnego modelu spójności pamięci**.

### Zadanie 4

Podaj kilka sprzętowych mechanizmów wspierania **synchronizacji**. Dlaczego rozróżnienie pomiędzy maszynami wielo- i jednoprocessorowymi jest tu istotne? Używając **instrukcji atomowej** typu `compare-and-swap` (podaj semantykę w pseudokodzie) zaimplementuj w asemblerze procesorów x86 operacje `lock` i `unlock` dla **blokad wirujących**.

### Zadanie 5

Proces oczekujący pod semaforem może zostać wstrzymany do czasu jego podniesienia lub **aktywnie czekać** na to zdarzenie. Jakie są zalety i wady obydwu rozwiązań? Na podstawie mechanizmu **futex** systemu Linux opisz rozwiązanie pośrednie (tj. semafony adaptacyjne). Jaka jest semantyka operacji `FUTEX_WAIT` i `FUTEX_WAKE`?

### Zadanie 6

Wymień cztery warunki konieczne do zaistnienia **zakleszczenia** i wyjaśnij jak im zapobiegać. Opisz zjawisko **odwrócenia priorytetów**. W jaki sposób jądro systemu może zapobiegać takiemu problemowi? Podaj przykład kosztownego projektu, który został poważnie zagrożony wskutek wystąpienia tego zjawiska.

### Zadanie 7

Opisz semantykę operacji `wait`, `signal`, `broadcast` na **zmiennych warunkowych**. Czemu operacja `wait` wymaga przekazania blokady? Co pamięta obiekt zmiennej warunkowej? Jedną z wysokopoziomowych konstrukcji zapewniających synchronizację jest **monitor** – wytłumacz to pojęcie z użyciem diagramu. Podaj przykład zastosowania monitora i wyjaśnij jak do jego implementacji można użyć zmiennych warunkowych. Czym różnią się **monitory Hoare’a** od **monitorów Mesa**?

### Zadanie 8

**Wymiana komunikatów** wymaga implementacji przynajmniej dwóch metod: `send(dest, msg)` i `recv(src, msg)`. Komunikat ma strukturę rekordu – jakie pola mogą w nim występować? W jaki sposób można **adresować nadawcę / odbiorcę**? **Skrzynki pocztowe** mogą mieć różne implementacje, od których zależy semantyka operacji `send` i `recv`. Podaj przykłady zachowania wymienionych funkcji – czym w tym kontekście są **punkty schadzek** (ang. *rendezvous*).

### Zadanie 9

Okazuje się, że istnieją techniki zachowania **spójności** struktur danych bez stosowania blokad. Jeśli założymy, że jedną strukturę może przeglądać wielu czytających i aktualizować co najwyżej jeden piszący, to możemy użyć mechanizmu **RCU** (ang. read-copy-update). Wyjaśnij zasadę działania RCU. Jakie dodatkowe założenia należy przyjąć w stosunku do wątków czytających? W jaki sposób wykorzystywany jest fakt oddzielenia **fazy usuwania** elementów od **fazy zwalniania** ich pamięci.