

Zadanie 2.

Instrukcja atomowa-operacja niepodzielna, tzn. proces nie może być wywłaszczony podczas jej wykonywania.

compare-and-swap- it compares the contents of a memory location with a given value and, only if they are the same, modifies the contents of that memory location to a new given value.

```
function cas(p : pointer to int, old : int, new : int) returns bool {
    if *p != old {
        return false
    }
    *p := new
    return true
}
```

Blokada wirująca- sposób realizacji synchronizacji międzyprocesowej, w którym oczekiwanie na zwolnienie blokady polega na ciągłym badaniu jej stanu. Spinlock działa więc na zasadzie aktywnego oczekiwania, tj. zużywa czas procesora, wykonując cały czas pustą pętlę:

```
while blokada aktywna do
    {nic nie rób};
```

Kod:

```
void lock(spin_t *m) {
    while (!cas(m, 0, 1));
}

void unlock(spin_t *m) {
    cas(m, 1, 0);
}
```

W systemach wieloprocessorowych spinlock może być bardziej efektywny, ponieważ pętla wykonuje się w kontekście procesu - unika się kosztownego czasowo wstrzymania procesu i przełączenia kontekstu na kod systemowy.

- Blokad wirujące zużywają bezsensownie CPU
- Blokad usypiające wymagają zmiany kontekstu, przejścia do jądra itp.
- Problem z blokadami wirującymi także pojawia się, gdy procesy mają priorytety wykonania. Załóżmy, że wątek T1 ma niższy priorytet niż T2, tzn. T1 działa tylko wtedy, gdy T2 czeka na coś. Sytuacja: T2 jest na razie zablokowane, bo czeka na jakieś IO. T1 działa, bierze spin locka i wchodzi do sekcji krytycznej i T2 skończył czekać na IO. Scheduler wywłaszcza T1 i daje CPU T2. T2 idzie do spin locka i widzi, że jest zajęty, więc się kręci. To kręcenie się nie zatrzyma, bo T1 nigdy nie dostanie już czasu CPU.

Zadanie 4.

Mamy 4. wątki oraz $v=2$. Wątek 1 i 2 wołają P i wychodzą. Teraz wątki 3, 4 wołają P i zatrzymują się między instrukcjami 17. i 18. Teraz 1. i 2. wchodzi do V. To oznacza, że `delay` zostanie zwiększony dwa razy. To jest niemożliwe, bo to jest binarny semafor. Wynikiem tego będzie, że któryś z wątków 3. lub 4. zawiśnie na 18 instrukcji.