Zadanie 4

Należy rozpatrzyć współbieżne procesy P2, P3, P4, P5. Proces P2 zawiera instrukcję S2, proces P3 instrukcję S3 i po niej instrukcję S6, proces P4 instrukcję S4 a proces P5 instrukcję S5 i po niej instrukcję S1. Wykorzystując semafory należy podać rozwiązanie zadania synchronizacji polegającego na wykonaniu instrukcji w następującej kolejności: instrukcje S2 i S4, powinny się wykonać po instrukcji S5; instrukcja S6 po S4; a instrukcja S1 - gdy wykonane są już instrukcje S2, S3 i S4. Podać uzasadnienie.

Rozwiązanie

W pierwszej kolejności musimy utworzyć macierz grafu powiązań między instrukcjami:

		Instrukcje poprzedzane					
		S1	S2	S3	S4	S5	S6
Instrukcje poprzedzające	S1	0	0	0	0	0	0
	S2	1	0	0	0	0	0
	S3	1	0	0	0	0	0
	S4	1	0	0	0	0	1
	S5	0	1	0	1	0	0
	S6	0	0	0	0	0	0

W macierzy powyżej zawarte są informacje z zadania:

– S1 jest poprzedzany przez S2, S3, S4

- S2 jest poprzedzany przez S5

- S4 jest poprzedzany przez S5

- S6 jest poprzedzany przez S4

Semafory będą reprezentowane tutaj jako rozdzielne prostokąty składające się z samych zer. W tak przedstawionej macierzy możemy mieć wrażenie że potrzebne będą 4 semafory – jednak poprzez przestawianie wierszy i kolumn możemy dojść do układu, w którym będziemy potrzebowali jedynie 3 semaforów. Pozbędę się również kolumn S3 i S5, jako że instrukcje S3 i S5 nie są poprzedzane przez żaden inny proces.

	S1	S2	S4	S6	
S1	0	0	0	0	
S2	1	0	0	0	Po S2 wstawić sygnalizuj_cz()
S3	1	0	0	0	Po S3 wstawić sygnalizuj_cz()
S4	1	0	0	1	Po S4 wstawić sygnalizuj_cz() oraz sygnalizuj_n()
S5	0	1	1	0	Po S5 wstawić dwa razy sygnalizuj_z()
S6	0	0	0	0	
	Przed S1 wstawić czekaj_cz()	Przed S2 wstawić czekaj_z()	Przed S4 wstawić czekaj_z()		

Pozostała teraz kwestia wartości początkowych semaforów – zakładamy, że po wykonaniu wszystkich procesów wartość wszystkich semaforów powinna być równa zero. Operacja sygnalizuj() podwyższa wartość semafora o 1, operacja czekaj() – obniża wartość semafora o 1.

Ponieważ semafor czerwony jest sygnalizowany 3 razy, a oczekiwany tylko raz, jego wartość początkowa powinna być równa -2.

Wszystkie pozostałe semafory będą miały wartość początkową 0.

W końcu możemy pisać kod procesów (pozwolę sobie użyć pythonowego pseudokodu):

```
s_cz = -2
s_z = 0
s_n = 0

def p2():
        czekaj_z()
        s2
        sygnalizuj_cz()
def p3():
        s3
        sygnalizuj_cz()
        s6
def p4():
        czekaj_z()
        s4
        sygnalizuj_cz()
        sygnalizuj_n()
def p5():
        s5
        sygnalizuj_z()
        sygnalizuj_z()
        czekaj_cz()
        s1
```