

Teoria Współbieżności

Ćwiczenie 2

1 Cel ćwiczenia

Celem ćwiczenia jest wprowadzenie do analizy i dowodzenia poprawności algorytmów opisujących procesy działające współbieżnie.

2 Wprowadzenie

Każdy z przedstawionych procesów będzie wykonywał nieskończoną pętlę podzieloną na **sekcje lokalną** oraz **sekcje krytyczną**. Sekcja krytyczna może być wykonywana w danej chwili wyłącznie przez jeden proces a sekcji lokalna jest niezależna i związana ściśle z danym procesem.

Aby przedstawione algorytmy były uznane za działające poprawne muszą spełniać następujące warunki:

1. wzajemnego wykluczania - w sekcji krytycznej nie może działać więcej niż jeden proces
2. braku zakleszczenia - jeśli procesy chcą wejść do sekcji krytycznych w końcu któryś dostaje dostęp
3. braku zagłodzenia - brak scenariusza, w którym proces nigdy nie dostanie się do sekcji krytycznej

Aby formalnie pokazać, że przedstawione procesy współbieżne nie spełniają powyższych punktów należy na przykład:

- podać listę instrukcji,
- diagram stanów (nie musi być cały, wystarczy część),
- tabelkę w stylu debbugera,

która nie spełni 1. 2. albo 3. . Zamiast całych instrukcji można napisać tylko numer instrukcji procesu; należy pamiętać, że zmienne również przedstawiają stan systemu.

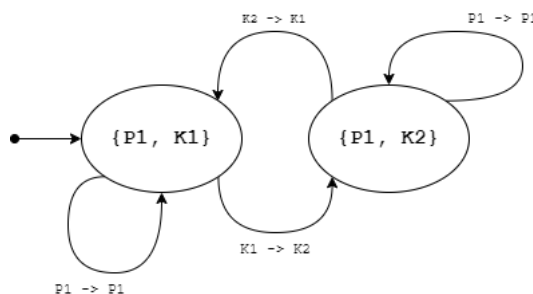
3 Przykłady

1. Przykład z diagramem stanów

<i>Proces P</i>	<i>Proces K</i>
while true:	while true:
P1: sekcja lokalna	K1: sekcja lokalna
	K2: sekcja krytyczna

Sekcja lokalna oraz sekcja krytyczna są zapisane skótowno; mogą w sobie zawierać wiele instrukcji. Px i Kx to numery instrukcji; dla uproszczenia główny while nie będzie numerowany.

System może być maksymalnie tylko w dwóch różnych stanach $\{P1, K1\}$ albo $\{P1, K2\}$. Algorytm jest poprawny ponieważ system spełnia warunki 1. 2. i 3. - widać to na diagramie stanów:



2. Przykład z tabelką

<i>Proces P</i>	<i>Proces K</i>
while true:	while true:
P1: sekcja lokalna	K1: sekcja lokalna
P2: sekcja krytyczna	K2: sekcja krytyczna

Algorytm nie spełnia warunku wzajemnego wykluczania ponieważ procesy mogą równocześnie znaleźć się w sekcjach krytycznych czyli w stanie $\{P2, K2\}$. Dowód:

<i>Czas/Krok</i>	<i>Proces P</i>	<i>Proces K</i>	<i>Zmienne globalne</i>
T1	P1	K1	brak
T2	P1	K2	-
T3	P1	K1	-
T4	P1	K2	-
T5	P2	K2	-

4 Ćwiczenia

Dla poniższych algorytmów działających współbieżnie sprawdź czy działają poprawnie - czy spełniają warunki 1. 2. oraz 3. lub pokaż, że nie spełniają któregośkolwiek z nich.

1.

<i>Zmienne globalne</i>	
działaj \leftarrow true	
<i>Proces P</i>	<i>Proces K</i>
while true:	while true:
P1: sekcja lokalna	K1: sekcja lokalna
P2: działaj \leftarrow false	K2: if działaj :
P3: sekcja krytyczna	K3: sekcja krytyczna
P4: działaj \leftarrow true	

Oprócz numerów instrukcji przy opisie stanu systemu należy również uwzględnić zmienne globalne.

2.

<i>Zmienne globalne</i>	
działaj \leftarrow true	
<i>Proces P</i>	<i>Proces K</i>
while true:	while true:
P1: sekcja lokalna	K1: if działaj :
P2: if działaj :	K2: działaj \leftarrow false
P3: działaj \leftarrow false	K3: sekcja krytyczna
P4: sekcja krytyczna	K4: działaj \leftarrow true
P5: działaj \leftarrow true	K5: sekcja lokalna

3.

Zmienne globalne	
czyja_kolej \leftarrow "P"	
Proces P	Proces K
while true:	while true:
P1: sekcja lokalna	K1: sekcja lokalna
P2: await czyja_kolej = "P"	K2: await czyja_kolej = "K"
P3: sekcja krytyczna	K3: sekcja krytyczna
P4: czyja_kolej \leftarrow "K"	K4: czyja_kolej \leftarrow "P"

Instrukcja await czeka aż warunek będzie prawdziwy.

4.

Zmienne globalne	
turaP \leftarrow false	
turaK \leftarrow false	
Proces P	Proces K
while true:	while true:
P1: sekcja lokalna	K1: sekcja lokalna
P2: await turaK = false	K2: await turaP = false
P3: turaP \leftarrow true	K3: turaK \leftarrow true
P4: sekcja krytyczna	K4: sekcja krytyczna
P5: turaP \leftarrow false	K5: turaK \leftarrow false

5.

Zmienne globalne	
turaP \leftarrow false	
turaK \leftarrow false	
Proces P	Proces K
while true:	while true:
P1: sekcja lokalna	K1: sekcja lokalna
P2: turaP \leftarrow true	K2: turaK \leftarrow true
P3: await turaK = false	K3: await turaP = false
P4: sekcja krytyczna	K4: sekcja krytyczna
P5: turaP \leftarrow false	K5: turaK \leftarrow false

6.

<i>Zmienne globalne</i>	
turaP \leftarrow false turaK \leftarrow false	
<i>Proces P</i>	<i>Proces K</i>
while true:	while true:
P1: sekcja lokalna	K1: sekcja lokalna
P2: turaP \leftarrow true	K2: turaK \leftarrow true
P3: while turaK :	K3: while turaP :
P4: turaP \leftarrow false	K4: turaK \leftarrow false
P5: turaP \leftarrow true	K5: turaK \leftarrow true
P6: sekcja krytyczna	K6: sekcja krytyczna
P7: turaP \leftarrow false	K7: turaK \leftarrow false

Literatura

- [1] Ben-Ari, Mordechai. Podstawy programowania współbieżnego i rozproszonego. Wydawnictwa Naukowo-Techniczne, 2009.