Rozdział 5.1.5

1. Czy bez podprogramu `zaczekaj_na_innych` wszystkie zadania filozofów będą zawsze przetwarzane współbieżnie w każdym uruchomieniu skryptu?

- Nie
- Może dojść do sytuacji, że jeden filozof skończy ucztę zanim drugi ją zacznie, co skutkuje przetwarzaniem sekwencyjnym
- Będzie brak współbieżności

Czy program zawierający rozwiązanie ucztujących filozofów będzie wówczas nadal poprawny?

- Tak
- Podprogram jest tylko po to, żeby zapewnić współbieżność dla wszystkich zadań filozofów

2. Czym jest zatrzask?

- barierą jednokrotnego zastosowania (każdy proces dotrze do pewnego momentu przetwarzania)

Jakie posiada elementy?

- licznik
- kolejka zadań oczekujących

Ile udostępnia operacji?

- jedną: osiągnięcie bariery

wskaż kiedy zadania filozofów synchronizują się z wykorzystaniem zatrzasku.

- gdy liczba zjedzonych posiłków osiągnie wartość $\text{liczba posiłków} - (\text{liczba posiłków}/2)$
- Zatrzymanie procesów filozofów, którzy szybciej zjedli swoją połowę, i wymuszenie na nich oczekiwania, aż wszyscy pozostali filozofowie również dotrą do tego samego etapu uczty.

3. Jakie mechanizmy synchronizacji i komunikacji zostały zastosowane przy budowie zatrzasku?

- synchronizacja - blokada wyłączna (synchronizacja dostępu do plików)
- komunikacja - potok (komunikacja między procesami; przesyłanie danych w sposób jednokierunkowy)

Należy wskazać kod odpowiedzialny za implementację zatrzasku w skryptach z rozwiązaniem problemu ucztujących filozofów.

```
------------------
flock -x -n $3
if [[ $? -eq 0 ]]
then
    komunikat $1 "Założyłem blokadę wyłączną na pliku $4"
    LICZNIK=$2
    while [[ $LICZNIK -gt 1 ]]
    do
        LICZNIK=$(($LICZNIK-$(cat $5|wc -l)))
    done
else
    komunikat $1 "Nie udało się założyć blokady wyłącznej na pliku blokady $4"
    echo &gt;$5
    komunikat $1 "Zakładam blokadę wyłączną na pliku $4"
    flock -x $3
        komunikat $1 "Założyłem blokadę wyłączną na pliku $4"
fi
flock -u $3
------------------

komunikat $1 "Zdjąłem blokadę wyłączną z pliku $4"
```

4. Czym jest spotkanie?

- przesłanie komunikatu od zadania nadawcy do zadania odbiorcy

Kto uczestniczy w pojedynczym spotkaniu?

- pierwszy filozof, który dotrze do bariery

Na czym polega komunikacja synchroniczna w spotkaniu?

- następna instrukcja będzie wykonana dopiero po zakończeniu operacji wysyłania lub odebrania wiadomości

Wymagane przygotowanie eksperymentu przedstawiającego synchronizację w spotkaniu dwóch procesów (utworzonych w osobnych terminalach znakowych), gdy proces nadawcy oczekuje na proces odbiorcy.

```
# tty0:
mkfifo potok
echo komunikat > potok

# tty1:
cat potok
```

5. Dla ustalonej przez nauczyciela liczby X filozofów ile będzie wszystkich spotkań?

- $X - 1$

6. Który filozof uczestniczy w każdym ze spotkań?

- ten, który pierwszy dotarł do bariery

Jaką pełni w nich rolę (nadawcy czy odbiorcy)?

- odbiorcy

7. Dla ustalonej przez nauczyciela liczby X filozofów ile minimalnie a ile maksymalnie komunikatów może odczytać jednorazowo odbiorca z pliku potoku?

Minimalnie

- raz

Maksymalnie

- $X - 1$
