

## Winiarze uproszczone

Krzysztof Krzyżaniak 145334  
Michał Kwart 145192

Studenci zawsze biorą całe wino, winiarze więc produkują po prostu "jedną jednostkę" wina. Winiarz może zdecydować, komu dać wino.

Parametry:

- S – liczba studentów
- W – liczba winiarzy
- B – liczba bezpiecznych miejsc

Zasoby lokalne:

- **clock** – aktualna wartość zegara Lamporta
- **winiarze\_announcements** – tablica ogłoszeń winiarzy
- **studenty\_requests** – posortowana po clock, pid tablica żądań wina
- **bezpieczne\_miejsce\_status** – (tylko dla winiarzy) tablica o rozmiarze B statusów zajęcia bezpiecznych miejsc
- **bezpieczne\_miejsce\_requests** – posortowana tablica żądań o bezpieczne miejsce po clock oraz pid

Proces winiarza:

1. Zegary **clock** procesów są inicjalizowane na wartość 0, statusy zajęcia bezpiecznego miejsca są ustawiane na **EMPTY**.
2. Winiarz odczeka losowy czas i po jego upływie produkuje 1 beczkę wina.
3. Winiarz wysyła do wszystkich winiarzy (łącznie ze sobą, wtedy wrzuca do swojej kolejki) żądanie rezerwacji jednego bezpiecznego miejsca (wraz ze swoim zegarem oraz pid).
4. Jeżeli żądanie winiarza jest na szczycie kolejki **bezpieczne\_miejsce\_requests** i otrzymał potwierdzenie od wszystkich innych winiarzy oraz istnieje minimum jedno wolne bezpieczne miejsce w kolejce **bezpieczne\_miejsce\_requests**, winiarz wysyła do innych winiarzy informację o zajęciu jednego z bezpiecznych miejsc oraz wysyła ogłoszenie o wyprodukowanym winie do studentów.
5. Obsługa wiadomości:
  1. Jeżeli dostaniemy żądanie o bezpieczne miejsce, dołączamy żądanie do kolejki **bezpieczne\_miejsce\_requests** (i ją sortujemy po clock, pid). Aktualizujemy zegar lamporta (clock z żądania + 1) oraz wysyłamy potwierdzenie odbioru do nadawcy żądania wraz z wartością zegara lamporta.
  2. Jeżeli dostaniemy komunikat o zajęciu bezpiecznego miejsca, aktualizujemy **bezpieczne\_miejsce\_status** oraz usuwamy żądanie zajmującego z kolejki **bezpieczne\_miejsce\_requests**.
  3. Jeżeli dostaniemy komunikat o zwolnieniu bezpiecznego miejsca, aktualizujemy **bezpieczne\_miejsce\_status**.
  4. Jeżeli dostaniemy komunikat od studenta o przejęciu beczki wina, (zakładając, że sama czynność utylizacji trwa 0) wysyłamy komunikat o zwolnieniu bezpiecznego miejsca do innych winiarzy i przechodzimy do kroku 2.

5. W przypadku wystąpienia scenariusza 6.1, 6.2 lub 6.3, wykonujemy ponownie (tylko) krok 4 i oczekuje na przejęcie beczki przez studenta.

**Proces studenta:**

1. Zegary **clock** procesów są inicjalizowane na wartość 0, wszystkie kolejki są puste.
2. Student wysyła do wszystkich studentów (łącznie ze sobą, wtedy wrzuca do swojej kolejki **studenty\_requests**) żądanie (wraz ze swoim zegarem oraz pid) utylizacji jednej beczki wina.
3. Jeżeli żądanie studenta jest na szczycie kolejki **studenty\_requests** i otrzymał potwierdzenie od wszystkich innych studentów oraz istnieje minimum jedna dostępna beczka wina w **winiarze\_announcements**, student wysyła do winiarza oraz pozostałych studentów informację o przejęciu beczki wina.
4. Student po utylizacji beczki wina jest losowy czas niezdolny do dalszej pracy.
5. Obsługa wiadomości:
  1. Jeżeli dostaniemy żądanie o beczkę wina, dołączamy żądanie do kolejki **studenty\_requests** (i ją sortujemy po clock, pid). Aktualizujemy zegar lamporta ( $\text{clock} = \text{clock} + 1$ ) oraz wysyłamy potwierdzenie odbioru do nadawcy żądania wraz z wartością zegara lamporta.
  2. Jeżeli otrzymamy informację o przejęciu beczki wina, usuwamy żądanie nadawcy z kolejki **studenty\_requests** oraz zmieniamy status winiarza w kolejce **winiarze\_announcements** na EMPTY.
  3. Jeżeli otrzymamy ogłoszenia o beczce wina od winiarza, uzupełniamy odpowiedni wpis w **winiarze\_announcements**.
  4. W przypadku wystąpienia scenariusza 5.1, 5.2 lub 5.3 skaczemy do kroku nr 3.