

Systemy operacyjne projekt

Temat 2 – Cegielnia, Antoni Jaros

Spis treści

1. Opis projektu	2
2. Ogólne zasady funkcjonowania kodu	2
Kod pracowników (Podstawowa funkcjonalność)	3
Kod pracowników (Mechanizmy synchronizacji)	4
Kod ciężarówek (Podstawowa funkcjonalność)	4
Kod ciężarówek (Mechanizmy synchronizacji)	5
Kod dyspozytora (Podstawowa funkcjonalność)	6
Kod dyspozytora (Mechanizmy synchronizacji)	6
Dodatkowe pliki	6
3. Opisy użytych funkcji	7
Plik: dyspozytor_utils.c	7
Plik: utils.c	8
Plik: pracownik_ciezarowka_utils.c	9
4. Testy i ich wyniki	10
Test pierwszy – normalne warunki pracy	10
Test drugi – wszystkie funkcje sleep usunięte	17
Test trzeci - bardzo dużo ciężarówek bez sleep	19
Test czwarty – bardzo duża pojemność taśmociągu i jedna ciężarówka bez sleep	22
Test piąty – jedna ciężarówka z bardzo małą pojemnością z sleep	23
4. Linki do istotnych fragmentów kodu	26
Tworzenie procesów (funkcje fork(), exec())	26
Obsługa sygnałów (handlers, signal(), kill())	26
Obsługa plików	26
Tworzenie i obsługa semaforów	27
Tworzenie i obsługa kolejki komunikatów	27

1. Opis projektu

Temat 2 – Cegielnia

W cegielni przy taśmie transportowej pracuje trzech pracowników oznaczonych przez P1, P2 i P3. Pracownicy wrzucają na taśmę cegły o masach odpowiednio 1, 2 i 3 jednostki. Na końcu taśmy stoi ciężarówka o ładowności C jednostek, którą należy zawsze załadować do pełna. Wszyscy pracownicy starają się układać cegły na taśmie najszybciej jak to możliwe. Taśma może przetransportować w danej chwili maksymalnie K sztuk cegieł. Jednocześnie jednak taśma ma ograniczony udźwig: maksymalnie M jednostek masy, tak, że niedopuszczalne jest położenie np. samych tylko cegieł najcięższych ($3K > M$). Cegły „zjeżdżające” z taśmy muszą od razu trafić na samochód dokładnie w takiej kolejności jak zostały położone na taśmie. Po zapelnieniu ciężarówki na jej miejsce pojawia się natychmiast (jeżeli jest dostępna!) nowa o ładowności C. Łączna liczba ciężarówek wynosi N, każda o ładowności C. Ciężarówki rozwożą cegły i po czasie T_i wracają do cegielni. Na polecenie dyspozytora (sygnał 1) ciężarówka, która w danym momencie stoi przy taśmie może odjechać z cegielni z niepełnym ładunkiem. Po otrzymaniu od dyspozytora polecenia (sygnał 2) pracownicy kończą pracę. Ciężarówki kończą pracę po rozwiezieniu wszystkich wyprodukowanych cegieł. Napisz programy symulujące działanie dyspozytora, pracowników i ciężarówek.

2. Ogólne zasady funkcjonowania kodu

Cały projekt podzielony jest na trzy główne moduły:

- Dyspozytor
- Ciężarówka
- Pracownik

których główny kod znajduje się odpowiednio w plikach `dyspozytor.c`, `ciezarowka.c` i `pracownik.c`. Symulacja rozpoczyna się od wywołania programu `dyspozytor`, który następnie tworzy wszystkie inne procesy (pracowników i ciężarówki) używając wywołań systemowych `fork()` i `exec()`. Synchronizacja opiera się na semaforach, kolejce komunikatów oraz na sygnałach. Semaforey oraz kolejka komunikatów mają minimalne wymagane prawa – 0600.

Aby skompilować program należy użyć załączonego pliku `Makefile` i wywołać polecenie `make all`. Następnie należy uruchomić symulację poleceniem `./dyspozytor`

Aby wysyłać sygnały do symulacji można posłużyć się załączonym skryptem `sigusr.sh`. `./sigusr.sh 1` – wysyła do symulacji SIGUSR1, `./sigusr.sh 2` – wysyła do symulacji SIGUSR2.

W następnej sekcji znajduje się omówienie zasady działania każdego modułu programu.

Kod pracowników (Podstawowa funkcjonalność)

Zasada działania pracownika:

1. Tworzenie cegły i przygotowanie do załadunku:

- Każdy pracownik (P1, P2, P3) odpowiada za generowanie cegieł o określonej masie:
 - P1: cegły o masie 1 jednostki.
 - P2: cegły o masie 2 jednostek.
 - P3: cegły o masie 3 jednostek.
- Pracownicy starają się jak najszybciej dodawać cegły na taśmę, jednocześnie respektując ograniczenia udźwigu taśmociągu, co jest osiągnięte poprzez odpowiednie obniżanie semaforów podczas układania cegieł.

2. Dodawanie cegły do taśmy:

- Pracownik próbuje zarezerwować miejsce na taśmie (liczba cegieł oraz maksymalna masa) za pomocą semaforów:
 - Obniżenie wartości semaforów odpowiadających za liczbę cegieł oraz masę cegieł na taśmie, oraz obniżenie semafora kontrolującego dostęp do taśmy między pracownikami i ciężarówkami.
- Jeśli miejsce zostało pomyślnie zarezerwowane, cegła zostaje dodana do kolejki komunikatów z informacją o masie cegły i ID pracownika.
- Informacja o dodaniu cegły jest wyświetlana na ekranie oraz zapisywana w pliku.

3. Czekanie na kolejną operację:

- Pracownik wykonuje losową przerwę (1-5 sekund), symulując czas potrzebny na wytworzenie kolejnej cegły.

4. Obsługa zakończenia pracy:

- Na sygnał SIGUSR2 pracownik kończy swoje działanie, wypisując odpowiednie logi o zakończeniu pracy na ekranie.
-

Kod pracowników (Mechanizmy synchronizacji)

1. Sygnały:

- **SIGUSR2:** Pracownik kończy pracę, wywołując handler sygnału i ustawiając odpowiednią zmienną na 0, co kończy główną pętlę pracownika.

2. Semaforey:

- Semaforey zarządzają liczbą dostępnych miejsc na taśmie i jej udźwigiem:
 - Semafor odpowiadający za maksymalną liczbę cegieł na taśmie jest zmniejszany podczas dodawania cegły i zwiększany przy odbiorze cegły przez ciężarówkę.
 - Semafor odpowiadający za udźwig taśmy działa w podobny sposób, redukując lub zwiększając masę cegieł na taśmie.
 - Semafor kontrolujący dostęp do zapisu oraz odczytu cegieł z taśmy, jest opuszczany przez pracownika, gdy ten chce umieścić cegłę na taśmie i podnoszony po operacji umieszczenia cegły.

3. Kolejki komunikatów:

- Każda cegła jest reprezentowana jako wiadomość w kolejce komunikatów. Pracownik dodaje wiadomość zawierającą:
 - Typ cegły.
 - ID pracownika, który ją umieścił.
 - Masę cegły.
-

Kod ciężarówek (Podstawowa funkcjonalność)

Zasada działania ciężarówki:

1. Odbieranie cegieł z taśmy:

- Ciężarówka podjeżdża do taśmy i rezerwuje dostęp za pomocą semafora, który pozwala tylko jednej ciężarówce operować na taśmie.

- Pobiera cegły z kolejki komunikatów zgodnie z kolejnością ich dodania (FIFO). Pętla załadunku kończy się, gdy ciężarówka osiągnie maksymalny ładunek lub gdy pracownicy nie wytwarzają już cegieł, a taśma jest pusta.

2. Rozwożenie cegieł:

- Po załadowaniu ciężarówka opuszcza cegielnię, co jest symulowane funkcją sleep. Po określonym czasie (np. 30 sekund) wraca do cegielni, aby ponownie rozpocząć cykl.

3. Obsługa poleceń dyspozytora:

- **SIGUSR1:** Na polecenie dyspozytora ciężarówka odjeżdża z niepełnym ładunkiem.
- **SIGUSR2:** Ciężarówka kończy swoją pracę po rozwiezieniu cegieł i upewnieniu się, że na taśmie nie został

Kod ciężarówek (Mechanizmy synchronizacji)

1. Sygnały:

- **SIGUSR1:** Ciężarówka opuszcza cegielnię niezależnie od stanu załadunku.
- **SIGUSR2:** Kończy działanie ciężarówki po rozwiezieniu wszystkich cegieł.

2. Semafony:

- Pierwszy semafor zarządza dostępem ciężarówek do taśmy, pozwalając na jednoczesną pracę tylko jednej ciężarówki.
- Drugi semafor pozwala ciężarówce na zablokowanie pracownikom dostępu do taśmy. Jest to przydatne, gdy ciężarówka musi zwrócić cegłę na taśmę, w przypadku, gdy masa cegły przekracza już ładowność ciężarówki.

3. Kolejki komunikatów:

- Ciężarówki pobierają cegły z kolejki, zachowując kolejność ich dodania przez pracowników.

4. Obsługa przerw:

- Jeśli ciężarówka nie może pobrać cegły (np. przekroczony ładunek), cegła jest zwracana na taśmę, aby była odebrana przez następną ciężarówkę.

Kod dyspozytora (Podstawowa funkcjonalność)

1. Zarządzanie systemem:

- Tworzy pracowników, ciężarówki oraz niezbędne struktury synchronizacyjne (kolejki komunikatów, semafony).
- Monitoruje stan systemu:
 - Liczba cegieł na taśmie.
 - Masa cegieł na taśmie.

2. Obsługa sygnałów:

- **SIGUSR1**: Wysyła polecenie wcześniejszego odjazdu ciężarówki.
- **SIGUSR2**: Rozpoczyna proces zamykania cegielni, kończąc pracę pracowników i ciężarówek.

3. Zamykanie systemu:

- Usuwa kolejki komunikatów oraz semafony.
 - Wysyła sygnały kończące pracę do wszystkich procesów.
-

Kod dyspozytora (Mechanizmy synchronizacji)

1. Sygnały:

- Zarządza sygnałami, wysyłając je do pracowników i ciężarówek.

2. Semafony:

- Kontrolują dostępność zasobów w systemie, takich jak miejsce na taśmie i masa cegieł.

3. Kolejki komunikatów:

- Przekazują dane między pracownikami, taśmą i ciężarówkami.

4. Procesy:

- Dyspozytor tworzy i monitoruje procesy dla pracowników i ciężarówek, synchronizując ich działanie.
-

Dodatkowe pliki

- **utils.h i utils.c:**

- Zawierają funkcje pomocnicze, takie jak tworzenie i usuwanie semaforów oraz kolejek komunikatów.
 - **pracownik_ciezarowka_utils.h i pracownik_ciezarowka_utils.c:**
 - Obsługują operacje dodawania cegieł, pobierania cegieł oraz operacje na semaforach.
 - **dyspozytor_utils.h i dyspozytor_utils.c:**
 - Zarządzają procesami, zawierają funkcje tworzenia pracowników i ciężarówek.
-

3. Opisy użytych funkcji

Plik: `dyspozytor_utils.c`

`void remove_message_queue(int msgid)`

Usuwa kolejkę komunikatów IPC.

- **Argumenty:**
 - `msgid (int)`: Identyfikator kolejki komunikatów.
- **Zwracana wartość:** Brak.
- **Opis:** Funkcja usuwa kolejkę komunikatów i obsługuje potencjalne błędy.

`void remove_semaphore(int semid)`

Usuwa zestaw semaforów IPC.

- **Argumenty:**
 - `semid (int)`: Identyfikator zestawu semaforów.
- **Zwracana wartość:** Brak.
- **Opis:** Funkcja usuwa zestaw semaforów i obsługuje potencjalne błędy.

`void create_workers(char *queue_key_string, char *semaphore_key_string)`

Tworzy procesy pracowników.

- **Argumenty:**

- `queue_key_string (char*)`: Klucz kolejki komunikatów jako ciąg znaków.
- `semaphore_key_string (char*)`: Klucz semaforów jako ciąg znaków.
- **Zwracana wartość:** Brak.
- **Opis:** Tworzy procesy potomne pracowników za pomocą `fork` i uruchamia odpowiednie programy przy użyciu `execl`.

`void create_trucks(char *queue_key_string, char *semaphore_key_string, char *truck_semaphore_key_string)`

Tworzy procesy ciężarówek.

- **Argumenty:**
 - `queue_key_string (char*)`: Klucz kolejki komunikatów jako ciąg znaków.
 - `semaphore_key_string (char*)`: Klucz semaforów pracowników jako ciąg znaków.
 - `truck_semaphore_key_string (char*)`: Klucz semaforów ciężarówek jako ciąg znaków.
- **Zwracana wartość:** Brak.
- **Opis:** Tworzy procesy potomne dla ciężarówek za pomocą `fork` i `execl`.

`void initialize_log_file(char *file_name)`

Inicjalizuje plik logów.

- **Argumenty:**
 - `file_name (char*)`: Nazwa pliku logów.
- **Zwracana wartość:** Brak.
- **Opis:** Tworzy lub nadpisuje plik logów, przygotowując go do zapisu danych.

Plik: `utils.c`

`int create_message_queue(key_t key)`

Tworzy lub otwiera kolejkę komunikatów IPC.

- **Argumenty:**

- key (key_t): Klucz do kolejki komunikatów.
- **Zwracana wartość:** Identyfikator kolejki komunikatów (int).
- **Opis:** Funkcja inicjalizuje kolejkę komunikatów z odpowiednimi uprawnieniami.

int create_semaphore(key_t key, int nsems, int init_val[])

Tworzy lub otwiera zestaw semaforów IPC.

- **Argumenty:**
 - key (key_t): Klucz do semaforów.
 - nsems (int): Liczba semaforów w zestawie.
 - init_val (int[]): Tablica wartości początkowych dla każdego semafora.
- **Zwracana wartość:** Identyfikator zestawu semaforów (int).
- **Opis:** Tworzy semafony i inicjalizuje je wartościami początkowymi, jeśli podano.

[Plik: pracownik_ciezarowka_utils.c](#)

void place_brick(int id, int mass, int type, int queue_id)

Umieszcza cegłę na taśmie transportowej.

- **Argumenty:**
 - id (int): ID pracownika lub ciężarówki.
 - mass (int): Masa cegły.
 - type (int): Typ cegły (np. zwykła lub pozostawiona).
 - queue_id (int): Identyfikator kolejki komunikatów.
- **Zwracana wartość:** Brak.
- **Opis:** Wysyła informacje o cegle do kolejki komunikatów.

void get_bricks(int truck_id, int queue_id, int semaphore_id, int *current_load, int *are_there_bricks, int *is_interrupted)

Obsługuje załadunek cegieł na ciężarówkę.

- **Argumenty:**

- truck_id (int): ID ciężarówki.
- queue_id (int): Identyfikator kolejki komunikatów.
- semaphore_id (int): Identyfikator semaforów.
- current_load (int*): Wskaźnik na bieżący ładunek ciężarówki.
- are_there_bricks (int*): Wskaźnik na flagę dostępności cegieł.
- is_interrupted (int*): Wskaźnik na flagę przerwania operacji.
- **Zwracana wartość:** Brak.
- **Opis:** Pobiera cegły z taśmy, synchronizując się za pomocą semaforów i kolejek.

void safe_sleep(int seconds)

Implementuje bezpieczny sen.

- **Argumenty:**
 - seconds (int): Liczba sekund do odczekania.
- **Zwracana wartość:** Brak.
- **Opis:** Przerywa sen tylko w przypadku obsługi sygnałów i wznowia po zakończeniu.

4. Testy i ich wyniki

Test pierwszy – normalne warunki pracy

W pierwszym teście symulacja działa w warunkach „normalnych”.

```
#define CONVEYOR_MAX_NUMBER 20 // Maksymalna ilość cegieł na taśmie
#define CONVEYOR_MAX_LOAD 30 // Maksymalny udźwig taśmy
#define NUMBER_OF_TRUCKS 3 // Ilość ciężarówek

#define TRUCK_MAX_LOAD 30 // Maksymalna ładowność ciężarówki
#define TRUCK_GONE_TIME 30 // Czas rozwożenia cegieł przez ciężarówkę
#define TRUCK_LOADING_TIME 1 // Czas potrzebny do załadunku cegły na ciężarówkę
```

Dostępne są 3 ciężarówki, każda ma 30 jednostek pojemności. Na taśmie może znajdować się maksymalnie 20 cegieł, a maksymalny udźwig wynosi 30 jednostek.

Uruchomienie symulacji:

```

jaros.antoni.151439@torus:~/projekt$ ./dyspozytor
Stworzono pracownika 1
Stworzono pracownika 2
Stworzono pracownika 3
Stworzono ciezarowke 1
Stworzono ciezarowke 2
Stworzono ciezarowke 3

```

Proces dyspozytor tworzy trzech pracowników oraz trzy ciężarówki

```

Ciezarowka 1 podjezdza do tasmy
Brak zostawionych cegiel: No message of desired type
Obecny ladunek 0/30
Liczba cegiel: 1/20
Masa cegiel: 1/30
Pracownik 1 dodal cegle o masie 1
Liczba cegiel: 0/20
Masa cegiel: 0/30
Ciezarowka 1 odebrala cegle o masie 1
Liczba cegiel: 1/20
Masa cegiel: 0/30
Liczba cegiel: 1/20
Masa cegiel: 2/30
Pracownik 2 dodal cegle o masie 2
Obecny ladunek 1/30
Liczba cegiel: 2/20
Masa cegiel: 2/30
Liczba cegiel: 2/20
Masa cegiel: 5/30
Pracownik 3 dodal cegle o masie 3
Liczba cegiel: 1/20
Masa cegiel: 3/30
Ciezarowka 1 odebrala cegle o masie 2
Liczba cegiel: 2/20

```

Pracownicy rozpoczynają wrzucanie cegieł na taśmę, co można zaobserwować po prawej stronie ekranu, natomiast ciężarówka 1 odbiera cegły po lewej stronie ekranu

```

logs_trucks.txt
1 Ciezarowka 1 odebrala cegle o masie 1
2 Ciezarowka 1 odebrala cegle o masie 2
3 Ciezarowka 1 odebrala cegle o masie 3
4 Ciezarowka 1 odebrala cegle o masie 1
5 Ciezarowka 1 odebrala cegle o masie 1
6 Ciezarowka 1 odebrala cegle o masie 2
7 Ciezarowka 1 odebrala cegle o masie 1
8 Ciezarowka 1 odebrala cegle o masie 3
9 Ciezarowka 1 odebrala cegle o masie 2
10 Ciezarowka 1 odebrala cegle o masie 3
11 Ciezarowka 1 odebrala cegle o masie 2
12 Ciezarowka 1 odebrala cegle o masie 3
13 Ciezarowka 1 odebrala cegle o masie 1
14 Ciezarowka 1 odebrala cegle o masie 3
15 Ciezarowka 1 odebrala cegle o masie 2
16 Ciezarowka 2 odebrala zostawiona cegle o masie 3
17 Ciezarowka 2 odebrala cegle o masie 1
18 Ciezarowka 2 odebrala cegle o masie 1
19 Ciezarowka 2 odebrala cegle o masie 2
20 Ciezarowka 2 odebrala cegle o masie 3
21 Ciezarowka 2 odebrala cegle o masie 3
22 Ciezarowka 2 odebrala cegle o masie 1
23 Ciezarowka 2 odebrala cegle o masie 2
24 Ciezarowka 2 odebrala cegle o masie 3
25 Ciezarowka 2 odebrala cegle o masie 1

logs_workers.txt
1 Pracownik 1 dodal cegle o masie 1
2 Pracownik 2 dodal cegle o masie 2
3 Pracownik 3 dodal cegle o masie 3
4 Pracownik 1 dodal cegle o masie 1
5 Pracownik 1 dodal cegle o masie 1
6 Pracownik 2 dodal cegle o masie 2
7 Pracownik 1 dodal cegle o masie 1
8 Pracownik 3 dodal cegle o masie 3
9 Pracownik 2 dodal cegle o masie 2
10 Pracownik 3 dodal cegle o masie 3
11 Pracownik 2 dodal cegle o masie 2
12 Pracownik 3 dodal cegle o masie 3
13 Pracownik 1 dodal cegle o masie 1
14 Pracownik 3 dodal cegle o masie 3
15 Pracownik 2 dodal cegle o masie 2
16 Pracownik 3 dodal cegle o masie 3
17 Pracownik 1 dodal cegle o masie 1
18 Pracownik 1 dodal cegle o masie 1
19 Pracownik 2 dodal cegle o masie 2
20 Pracownik 3 dodal cegle o masie 3
21 Pracownik 3 dodal cegle o masie 3
22 Pracownik 1 dodal cegle o masie 1
23 Pracownik 2 dodal cegle o masie 2
24 Pracownik 3 dodal cegle o masie 3
25 Pracownik 1 dodal cegle o masie 1

```

Akcje dodawania i zabierania cegieł z taśmy są zapisywane w plikach co pozwala na łatwą weryfikację, czy wszystkie cegły wyprodukowane cegły zostały rozwieszone.

```

Obecny ładunek 30/30
Cieżarówka 1 rozwozi cegły...
Cieżarówka 2 podjeżdża do taśmy

Liczba cegieł: 15/20
Masa cegieł: 29/30

Liczba cegieł: 16/20
Masa cegieł: 29/30

Liczba cegieł: 16/20
Masa cegieł: 30/30

Pracownik 2 dodał cegle o masie 2

Pracownik 1 dodał cegle o masie 1

Cieżarówka 2 odebrała zostawioną cegle o masie 3
Obecny ładunek 3/30
Cieżarówka 2 odebrała cegle o masie 1

Liczba cegieł: 15/20
Masa cegieł: 29/30

Liczba cegieł: 16/20
Masa cegieł: 29/30

Liczba cegieł: 16/20
Masa cegieł: 30/30

Pracownik 1 dodał cegle o masie 1

Obecny ładunek 4/30
Cieżarówka 2 odebrała cegle o masie 1

Liczba cegieł: 15/20
Masa cegieł: 29/30

Liczba cegieł: 16/20
Masa cegieł: 30/30

Pracownik 1 dodał cegle o masie 1

Obecny ładunek 5/30

Liczba cegieł: 15/20
Masa cegieł: 30/30

```

Po chwili ciężarówka 1 zapełnia się i odjeżdża rozwozić cegły, dzięki czemu do taśmy może podejść ciężarówka 2

```

● jaros.antoni.151439@torus:~/projekt$ ./sigusr.sh 1
Wysłano sygnał SIGUSR1 do procesu 'dyspozytor' (PID: 4171327).

```

Po wystaniu sygnału SIGUSR1 ciężarówka odjeżdża od taśmy mimo niepełnego ładunku

```

Obecny ładunek 4/30
Cieżarówka 2 odjazd na polecenie dyspozytora
Cieżarówka 2 rozwozi cegły...
Cieżarówka 3 podjeżdża do taśmy
Brak zostawionych cegieł: No message of desired type
Obecny ładunek 0/30
Cieżarówka 3 odebrała cegle o masie 3

```

Następnie do taśmy podjeżdża następna wolna ciężarówka, w tym przypadku ciężarówka 3

```

Obecny ładunek 7/30
Cieżarówka 3 odjazd na polecenie dyspozytora
Cieżarówka 3 rozwozi cegły...

```

Możemy znowu wystać sygnał SIGUSR1

Symulacja kontynuuje działanie – pracownicy produkują cegły, a ciężarówki je rozwożą, aż do momentu otrzymania sygnału SIGUSR2

```
● jaros.antoni.151439@torus:~/projekt$ ./sigusr.sh 2  
Wysłano sygnał SIGUSR2 do procesu 'dyspozytor' (PID: 4171327).
```

```
                                     Masa cegieł: 29/30  
Zatrzymywanie pracy                                                         Pracownik 2 kończy pracę  
                                                                              Pracownik 1 kończy pracę  
                                                                                
                                     Liczba cegieł: 17/20  
                                     Masa cegieł: 29/30  
                                                                                
                                     Liczba cegieł: 17/20  
                                     Masa cegieł: 29/30  
                                                                                
                                     Liczba cegieł: 17/20  
                                     Masa cegieł: 29/30  
Obecny ładunek 28/30  
Ciezarowka 1 odebrała cegle o masie 2  
Obecny ładunek 30/30  
Ciezarowka 1 zwróciła cegle o masie 1  
Obecny ładunek 30/30  
Ciezarowka 1 rozwozi cegły...  
Ciezarowka 3 podjeżdża do taśmy  
Ciezarowka 3 odebrała zostawioną cegle o masie 1  
Obecny ładunek 1/30  
Ciezarowka 3 odebrała cegle o masie 2  
Obecny ładunek 3/30  
Ciezarowka 3 odebrała cegle o masie 1  
Obecny ładunek 4/30  
Ciezarowka 3 odebrała cegle o masie 3  
Obecny ładunek 7/30  
Ciezarowka 3 odebrała cegle o masie 1  
Obecny ładunek 8/30  
Ciezarowka 3 odebrała cegle o masie 2  
Obecny ładunek 10/30  
Ciezarowka 3 odebrała cegle o masie 1  
Obecny ładunek 11/30  
Ciezarowka 3 odebrała cegle o masie 2  
Obecny ładunek 13/30
```

Następuje zatrzymanie pracy – pracownicy przestają układać cegły na taśmie, jednak ciężarówki kontynuują rozwożenie cegieł, dopóki wszystkie cegły z taśmy nie zostaną rozwiezione

```

Obecny ladunek 24/30
Ciezarowka 3 odebrala cegle o masie 2
Obecny ladunek 26/30
Ciezarowka 3 odebrala cegle o masie 1
Obecny ladunek 27/30
Brak zostawionych cegiel: No message of desired type
Ciezarowka 3 rozwozi cegly...
Ciezarowka 2 podjezdza do tasmy
Brak zostawionych cegiel: No message of desired type
Obecny ladunek 0/30
Brak zostawionych cegiel: No message of desired type
Ciezarowka 2 konczy prace
Ciezarowka 1 podjezdza do tasmy
Brak zostawionych cegiel: No message of desired type
Obecny ladunek 0/30
Brak zostawionych cegiel: No message of desired type
Ciezarowka 1 konczy prace

                                Liczba cegiel: 0/20
                                Masa cegiel: 0/30

                                Liczba cegiel: 0/20
                                Masa cegiel: 0/30

Ciezarowka 3 konczy prace

                                Liczba cegiel: 0/20
                                Masa cegiel: 0/30

Kolejka komunikatow o ID 6160389 zostala usunieta.
Zbior semaforow o ID 4554778 zostal usuniety.
Zbior semaforow o ID 4554779 zostal usuniety.

```

Gdy nie ma już cegieł na taśmie, ciężarówki kończą pracę, a następnie dyspozytor usuwa kolejkę komunikatów oraz zbiory semaforów i również kończy pracę.

Możemy następnie sprawdzić, czy wszystkie wyprodukowane cegły zostały rozwiezione. W tym celu możemy wejść do plików logs_trucks.txt oraz logs_workers.txt gdzie zapisywana jest każda operacja położenia cegły na taśmę przez pracownika oraz zdjęcia cegły z taśmy przez ciężarówkę.

```
logs_trucks.txt U X logs_workers.txt U
logs_trucks.txt
1 Ciezarowka 1 odebrala cegle o masie 1
2 Ciezarowka 1 odebrala cegle o masie 1
3 Ciezarowka 1 odebrala cegle o masie 3
4 Ciezarowka 1 odebrala cegle o masie 3
5 Ciezarowka 1 odebrala cegle o masie 2
6 Ciezarowka 1 odebrala cegle o masie 3
7 Ciezarowka 1 odebrala cegle o masie 1
8 Ciezarowka 1 odebrala cegle o masie 2
9 Ciezarowka 1 odebrala cegle o masie 1
10 Ciezarowka 1 odebrala cegle o masie 2
11 Ciezarowka 1 odebrala cegle o masie 3
12 Ciezarowka 1 odebrala cegle o masie 1
13 Ciezarowka 1 odebrala cegle o masie 3
14 Ciezarowka 1 odebrala cegle o masie 2
15 Ciezarowka 1 odebrala cegle o masie 2
logs_workers.txt
1 Pracownik 1 dodal cegle o masie 1
2 Pracownik 1 dodal cegle o masie 1
3 Pracownik 3 dodal cegle o masie 3
4 Pracownik 3 dodal cegle o masie 3
5 Pracownik 2 dodal cegle o masie 2
6 Pracownik 3 dodal cegle o masie 3
7 Pracownik 1 dodal cegle o masie 1
8 Pracownik 2 dodal cegle o masie 2
9 Pracownik 1 dodal cegle o masie 1
10 Pracownik 2 dodal cegle o masie 2
11 Pracownik 3 dodal cegle o masie 3
12 Pracownik 1 dodal cegle o masie 1
13 Pracownik 3 dodal cegle o masie 3
14 Pracownik 2 dodal cegle o masie 2
15 Pracownik 2 dodal cegle o masie 2
16 Pracownik 1 dodal cegle o masie 1
17 Pracownik 2 dodal cegle o masie 2
18 Pracownik 3 dodal cegle o masie 3
19 Pracownik 1 dodal cegle o masie 1
20 Pracownik 3 dodal cegle o masie 3
21 Pracownik 2 dodal cegle o masie 2
22 Pracownik 2 dodal cegle o masie 2
```

Porównując ilość linii w każdym pliku możemy sprawdzić czy każda wyprodukowana cegła została odebrana.


```

● jaros.antoni.151439@torus:~/projekt$ wc -l logs_trucks.txt logs_workers.txt
   81 logs_trucks.txt
   81 logs_workers.txt
  162 total

```

Jak widać pliki mają taką samą ilość linii co oznacza, że każda wyprodukowana cegła została odebrana.

Symulacja zakończyła się więc sukcesem, wszystko działało płynnie, nie było żadnych zakleszczeń, a procesy poprawnie reagowały na sygnały.

Wszystkie używane przez symulację zbiory semaforów oraz kolejka komunikatów zostały usunięte, co potwierdza:

```

❌ jaros.antoni.151439@torus:~/projekt$ ipcs | grep $(whoami)

```

Test zakończył się sukcesem

Test drugi – wszystkie funkcje sleep usunięte

W drugim teście usunąłem wszystkie funkcje sleep, aby sprawdzić czy synchronizacja programu nie opiera się na czekaniu procesów przez funkcję sleep.

Parametry symulacji takie same jak w poprzednim teście, ale wszystkie funkcje sleep zostały usunięte.

```

jaros.antoni.151439@torus:~/projekt$ ./dyspozytor
Stworzono pracownika 1
Stworzono pracownika 2
Stworzono pracownika 3
Stworzono ciezarowke 1
Stworzono ciezarowke 2
Stworzono ciezarowke 3

Liczba cegiel: 0/20
Masa cegiel: 0/30

Liczba cegiel: 1/20
Masa cegiel: 1/30

Liczba cegiel: 2/20
Masa cegiel: 2/30
Pracownik 1 dodal cegle o masie 1

Liczba cegiel: 3/20
Masa cegiel: 2/30
Pracownik 1 dodal cegle o masie 1

Liczba cegiel: 3/20
Masa cegiel: 4/30
Pracownik 2 dodal cegle o masie 2

Liczba cegiel: 4/20
Masa cegiel: 5/30
Pracownik 1 dodal cegle o masie 1

Liczba cegiel: 5/20
Masa cegiel: 5/30

```

Na początku symulacji wszystkie procesy tworzą się jak należy, pracownicy

```

Ciezarowka 2 odebrała cegle o masie 3
Liczba cegiel: 15/20
Masa cegiel: 30/30

Obecny ładunek 21/30
Liczba cegiel: 14/20
Masa cegiel: 28/30
Pracownik 3 dodał cegle o masie 3

Ciezarowka 2 odebrała cegle o masie 2
Liczba cegiel: 15/20
Masa cegiel: 28/30
Liczba cegiel: 15/20
Masa cegiel: 30/30

Obecny ładunek 23/30
Liczba cegiel: 14/20
Masa cegiel: 30/30
Pracownik 2 dodał cegle o masie 2

Ciezarowka 2 odebrała cegle o masie 1
Liczba cegiel: 14/20
Masa cegiel: 29/30
Liczba cegiel: 15/20
Masa cegiel: 30/30

Obecny ładunek 24/30

```

Symulacja działa bardzo szybko, ale bez żadnych zakłóceń ani zakleszczeń.

```

Liczba cegiel: 15/20
Masa cegiel: 30/30

Obecny ładunek 18/30
Zatrzymywanie pracy
Pracownik 1 dodał cegle o masie 1
Pracownik 1 kończy pracę
Pracownik 3 kończy pracę
Pracownik 2 kończy pracę

Ciezarowka 3 odebrała cegle o masie 3
Obecny ładunek 21/30
Ciezarowka 3 odebrała cegle o masie 2
Obecny ładunek 23/30
Ciezarowka 3 odebrała cegle o masie 1
Obecny ładunek 24/30
Ciezarowka 3 odebrała cegle o masie 3
Liczba cegiel: 11/20
Masa cegiel: 21/30
Liczba cegiel: 11/20
Masa cegiel: 21/30

Obecny ładunek 27/30
Ciezarowka 3 odebrała cegle o masie 2
Liczba cegiel: 10/20
Masa cegiel: 19/30

Obecny ładunek 29/30
Ciezarowka 3 odebrała cegle o masie 1
Obecny ładunek 30/30

```

Po wystąpieniu SIGUSR2 pracownicy kończą pracę jak należy.

```

Obecny ladunek 17/30
Ciezarowka 2 odebrala cegle o masie 1
Obecny ladunek 18/30
Brak zostawionych cegiel: No message of desired type
Ciezarowka 2 rozwozi cegly...
Ciezarowka 2 konczy prace
Ciezarowka 1 podjezdza do tasmy
Brak zostawionych cegiel: No message of desired type
Obecny ladunek 0/30
Brak zostawionych cegiel: No message of desired type
                                     Liczba cegiel: 0/20
                                     Masa cegiel: 0/30

Ciezarowka 1 konczy prace
Ciezarowka 3 podjezdza do tasmy
Brak zostawionych cegiel: No message of desired type
Obecny ladunek 0/30
Brak zostawionych cegiel: No message of desired type
Ciezarowka 3 konczy prace
                                     Liczba cegiel: 0/20
                                     Masa cegiel: 0/30

                                     Liczba cegiel: 0/20
                                     Masa cegiel: 0/30

Kolejka komunikatow o ID 6357012 zostala usunieta.
Zbior semaforow o ID 4882483 zostal usuniety.
Zbior semaforow o ID 4882484 zostal usuniety.
jaros.antoni.151439@torus:~/projekt$ ipcs | grep $(whoami)
jaros.antoni.151439@torus:~/projekt$ |

```

Ciężarówki kontynuują rozwożenie cegieł, dopóki nie skończą się cegły. Na końcu wszystkie zbiory semaforów i kolejka komunikatów zostają usunięte.

```

• jaros.antoni.151439@torus:~/projekt$ wc -l logs_workers.txt logs_trucks.txt
  16254 logs_workers.txt
  16254 logs_trucks.txt
   32508 total

```

Po porównaniu długości plików z logami można stwierdzić, że wszystkie wyprodukowane przez pracowników cegły zostały rozwiezione, więc test został zakończony pomyślnie.

Test trzeci - bardzo dużo ciężarówek bez sleep

W trzecim teście, podobnie jak w drugim wszystkie funkcje sleep są usunięte, ponadto liczba ciężarówek jest zwiększona do 1000, aby zobaczyć, jak symulacja radzi sobie w takich warunkach, pozostałe parametry są bez zmian.

```

jaros.antoni.151439@torus:~/projekt$ ./dyspozytor
Stworzono pracownika 1
Stworzono pracownika 2
Stworzono pracownika 3
Stworzono ciezarowke 1
Stworzono ciezarowke 2

Stworzono ciezarowke 3
Pracownik 1 dodal cegle o masie 1
Pracownik 1 dodal cegle o masie 1
Pracownik 1 dodal cegle o masie 1

Stworzono ciezarowke 4
Pracownik 2 dodal cegle o masie 2
Pracownik 1 dodal cegle o masie 1

Stworzono ciezarowke 5
Pracownik 3 dodal cegle o masie 3

Ciezarowka 1 podjezdza do tasmy
Stworzono ciezarowke 6
Pracownik 2 dodal cegle o masie 2
Pracownik 1 dodal cegle o masie 1

Brak zostawionych cegiel: No message of desired type
Obecny ladunek 0/30
Stworzono ciezarowke 7
Pracownik 3 dodal cegle o masie 3
Pracownik 2 dodal cegle o masie 2
Pracownik 1 dodal cegle o masie 1

Stworzono ciezarowke 8
Ciezarowka 1 odebrala cegle o masie 1
Stworzono ciezarowke 9
Obecny ladunek 1/30
Pracownik 3 dodal cegle o masie 3
Pracownik 2 dodal cegle o masie 2
Pracownik 1 dodal cegle o masie 1

Stworzono ciezarowke 10
Pracownik 3 dodal cegle o masie 3
Pracownik 2 dodal cegle o masie 2
Pracownik 1 dodal cegle o masie 1

Stworzono ciezarowke 11
Ciezarowka 1 odebrala cegle o masie 1
Obecny ladunek 2/30
Pracownik 3 dodal cegle o masie 3
Pracownik 2 dodal cegle o masie 2
Pracownik 1 dodal cegle o masie 1

Stworzono ciezarowke 12
Pracownik 3 dodal cegle o masie 3
Pracownik 2 dodal cegle o masie 2
Pracownik 1 dodal cegle o masie 1

Stworzono ciezarowke 13
Ciezarowka 1 odebrala cegle o masie 1
Obecny ladunek 3/30
Stworzono ciezarowke 14

```

Na początku dyspozytor przystępuje do tworzenia ciężarówek. Można zobaczyć, że pracownicy jak i dotychczas stworzone ciężarówki zaczynają pracować, mimo że dyspozytor nie stworzył jeszcze wszystkich ciężarówek. Nie przeszkadza to jednak w żadnym stopniu w synchronizacji, gdyż nowo stworzone ciężarówki po prostu ustawiają się w kolejce na semaforze

```

Liczba cegieł: 15/20
Masa cegieł: 30/30
Pracownik 2 dodał cegle o masie 2

Ciezarowka 298 odebrała cegle o masie 3
Liczba cegieł: 15/20
Masa cegieł: 27/30
Liczba cegieł: 16/20
Masa cegieł: 27/30
Obecny ładunek 9/30
Liczba cegieł: 16/20
Masa cegieł: 30/30
Pracownik 3 dodał cegle o masie 3
Ciezarowka 298 odebrała cegle o masie 1
Liczba cegieł: 15/20
Masa cegieł: 30/30
Liczba cegieł: 15/20
Masa cegieł: 29/30
Obecny ładunek 10/30
Liczba cegieł: 16/20
Masa cegieł: 30/30
Pracownik 1 dodał cegle o masie 1
Ciezarowka 298 odebrała cegle o masie 2
Liczba cegieł: 15/20
Masa cegieł: 30/30
Liczba cegieł: 15/20
Masa cegieł: 28/30
Liczba cegieł: 16/20
Masa cegieł: 28/30
Obecny ładunek 12/30
Liczba cegieł: 16/20

```

W trakcie symulacji można zobaczyć jak kolejne ciężarówki podjeżdżają do taśmy odbierać cegły.

```

Brak zostawionych cegieł: No message of desired type
Obecny ładunek 0/30
Brak zostawionych cegieł: No message of desired type
Ciezarowka 998 kończy prace
Ciezarowka 1000 podjeżdża do taśmy
Brak zostawionych cegieł: No message of desired type
Obecny ładunek 0/30
Brak zostawionych cegieł: No message of desired type
Ciezarowka 1000 kończy prace
Liczba cegieł: 0/20
Masa cegieł: 0/30
Ciezarowka 999 podjeżdża do taśmy
Liczba cegieł: 0/20
Masa cegieł: 0/30
Brak zostawionych cegieł: No message of desired type
Obecny ładunek 0/30
Brak zostawionych cegieł: No message of desired type
Ciezarowka 999 kończy prace
Liczba cegieł: 0/20
Masa cegieł: 0/30
Ciezarowka 829 podjeżdża do taśmy
Brak zostawionych cegieł: No message of desired type
Obecny ładunek 0/30
Brak zostawionych cegieł: No message of desired type
Ciezarowka 829 kończy prace
Liczba cegieł: 0/20
Masa cegieł: 0/30
Liczba cegieł: 0/20
Masa cegieł: 0/30
Kolejka komunikatów o ID 6717454 została usunięta.
Zbiór semaforów o ID 5144627 został usunięty.
Zbiór semaforów o ID 5144628 został usunięty.

```

Po wystąpieniu SIGUSR2 wszystkie ciężarówki po kolei kończą pracę.

```

● jaros.antoni.151439@torus:~/projekt$ wc -l logs_workers.txt logs_trucks.txt
182952 logs_workers.txt
182952 logs_trucks.txt
365904 total
⊗ jaros.antoni.151439@torus:~/projekt$ ipcs | grep $(whoami)
○ jaros.antoni.151439@torus:~/projekt$

```

Można również zobaczyć, że wszystkie wyprodukowane cegły zostały rozwieszone, a semafony i kolejka komunikatów zostały usunięte. Test zakończył się sukcesem

Test czwarty – bardzo duża pojemność taśmociągu i jedna ciężarówka bez sleep

Czwarty test polega na ustawieniu pojemności taśmociągu na 1000 cegieł, udźwigu na 3000 jednostek masy oraz stworzeniu tylko jednej ciężarówki.

```

#define CONVEYOR_MAX_NUMBER 1000 // Maksymalna ilosc cegieł na taśmie
#define CONVEYOR_MAX_LOAD 3000   // Maksymalny udźwig taśmy
#define NUMBER_OF_TRUCKS 1       // Ilość ciężarówek

```

```

Masa cegieł: 2000/3000

Ciężarówka 1 odebrała cegle o masie 2                               Pracownik 2 dodał cegle o masie 2
Liczba cegieł: 999/1000
Masa cegieł: 1998/3000
Liczba cegieł: 1000/1000
Masa cegieł: 1999/3000

Obecny ładunek 11/30
Ciężarówka 1 odebrała cegle o masie 1                               Pracownik 1 dodał cegle o masie 1
Liczba cegieł: 999/1000
Masa cegieł: 1998/3000
Liczba cegieł: 1000/1000
Masa cegieł: 2001/3000

Obecny ładunek 12/30
Ciężarówka 1 odebrała cegle o masie 3                               Pracownik 3 dodał cegle o masie 3
Liczba cegieł: 999/1000
Masa cegieł: 1998/3000
Liczba cegieł: 1000/1000
Masa cegieł: 2000/3000

Obecny ładunek 15/30
Ciężarówka 1 odebrała cegle o masie 2                               Pracownik 2 dodał cegle o masie 2
Liczba cegieł: 999/1000
Masa cegieł: 2000/3000
Liczba cegieł: 1000/1000
Masa cegieł: 1999/3000

Obecny ładunek 17/30
Ciężarówka 1 odebrała cegle o masie 1                               Pracownik 1 dodał cegle o masie 1
Liczba cegieł: 999/1000

```

Podczas symulacji nie widać żadnych odchyśleń. Taśma jest cały czas prawie maksymalnie zapelniona co jest spodziewanym zachowaniem, gdyż ciężarówka jest tylko jedna a pracowników jest trzech.

```

jaros.antoni.151439@torus:~/projekt$ ./sigusr.sh 2
Wysłano sygnał SIGUSR2 do procesu 'dyspozytor' (PID: 730306
764569).
jaros.antoni.151439@torus:~/projekt$ wc -l logs_workers.txt logs_trucks.txt
40381 logs_workers.txt
40381 logs_trucks.txt
80762 total
jaros.antoni.151439@torus:~/projekt$ ipcs | grep $(whoami)
jaros.antoni.151439@torus:~/projekt$

```

Weryfikacja czy wszystkie wyprodukowane cegły zostały rozwiezione.

Wszystko zadziało poprawnie, więc test jest zakończony sukcesem.

Test piąty – jedna ciężarówka z bardzo małą pojemnością z sleep

W piątym teście funkcje sleep zostały przywrócone, natomiast zostanie stworzona tylko jedna ciężarówka o pojemności 3.

```

#define CONVEYOR_MAX_NUMBER 20 // Maksymalna ilość cegieł na taśmie
#define CONVEYOR_MAX_LOAD 30 // Maksymalny udźwig taśmy
#define NUMBER_OF_TRUCKS 1 // Ilość ciężarówek

#define TRUCK_MAX_LOAD 3 // Maksymalna ładowność ciężarówki
#define TRUCK_GONE_TIME 30 // Czas rozwożenia cegieł przez ciężarówkę
#define TRUCK_LOADING_TIME 2 // Czas potrzebny do załadowania cegły na ciężarówkę

```

```

jaros.antoni.151439@torus:~/projekt$ ./dyspozytor
Stworzono pracownika 1
Stworzono pracownika 2
Stworzono pracownika 3
Stworzono ciężarówkę 1

Liczba cegieł: 0/20
Masa cegieł: 0/30

Ciężarówka 1 podjeżdża do taśmy
Brak zostawionych cegieł: No message of desired type
Obecny ładunek 0/3

Liczba cegieł: 1/20
Masa cegieł: 1/30

Pracownik 1 dodał cegle o masie 1

Liczba cegieł: 0/20
Masa cegieł: 1/30

Ciężarówka 1 odebrała cegle o masie 1

Liczba cegieł: 0/20
Masa cegieł: 0/30

Liczba cegieł: 1/20
Masa cegieł: 2/30

Pracownik 2 dodał cegle o masie 2

Liczba cegieł: 2/20
Masa cegieł: 5/30

Pracownik 3 dodał cegle o masie 3

Obecny ładunek 1/3
Ciężarówka 1 odebrała cegle o masie 2

Liczba cegieł: 1/20
Masa cegieł: 5/30

Liczba cegieł: 1/20
Masa cegieł: 3/30

Liczba cegieł: 2/20
Masa cegieł: 6/30

```

Symulacja rozpoczyna się tak jak zwykle, od stworzenia trzech pracowników i tym razem jednej ciężarówki. Wszystkie procesy od razu przystępują do pracy

```
Obecny ładunek 3/3  
Ciężarówka 1 rozwozi cegły...  
  
Liczba cegieł: 7/20  
Masa cegieł: 16/30  
  
Liczba cegieł: 8/20  
Masa cegieł: 16/30  
  
Liczba cegieł: 8/20  
Masa cegieł: 18/30  
  
Liczba cegieł: 9/20  
Masa cegieł: 19/30  
  
Pracownik 1 dodał cegle o masie 1  
  
Pracownik 3 dodał cegle o masie 3  
  
Pracownik 2 dodał cegle o masie 2
```

Ciężarówka przez swoją małą pojemność musi praktycznie do razu rozwiązać cegły.

```
Ciężarówka 1 rozwozi cegły...  
  
Liczba cegieł: 7/20  
Masa cegieł: 16/30  
  
Liczba cegieł: 8/20  
Masa cegieł: 16/30  
  
Liczba cegieł: 8/20  
Masa cegieł: 17/30  
  
Liczba cegieł: 9/20  
Masa cegieł: 17/30  
  
Liczba cegieł: 9/20  
Masa cegieł: 19/30  
  
Liczba cegieł: 10/20  
Masa cegieł: 19/30  
  
Liczba cegieł: 10/20  
Masa cegieł: 22/30  
  
Liczba cegieł: 11/20  
Masa cegieł: 25/30  
  
Liczba cegieł: 12/20  
Masa cegieł: 28/30  
  
Liczba cegieł: 13/20  
Masa cegieł: 29/30  
  
Liczba cegieł: 14/20  
Masa cegieł: 30/30  
  
Pracownik 3 dodał cegle o masie 3  
  
Pracownik 1 dodał cegle o masie 1  
  
Pracownik 2 dodał cegle o masie 2  
  
Pracownik 3 dodał cegle o masie 3  
  
Pracownik 3 dodał cegle o masie 3  
  
Pracownik 3 dodał cegle o masie 3  
  
Pracownik 1 dodał cegle o masie 1
```

Podczas rozwożenia cegieł przez ciężarówkę pracownicy kontynuują układanie cegieł na taśmie, która szybko się zapętnia. Wtedy w symulacji nie dzieje się nic, ponieważ pracownicy nie mogą dokładać nowych cegieł, a ciężarówka jeszcze nie wróciła. Jest to spodziewane i pożądane zachowanie.


```

Obecny ładunek 3/3
Ciezarowka 1 rozwozi cegly...
Zatrzymywanie pracy

Liczb cegiel: 15/20
Masa cegiel: 30/30

Liczb cegiel: 15/20
Masa cegiel: 30/30

Liczb cegiel: 15/20
Masa cegiel: 30/30

Ciezarowka 1 podjezdza do tasmy

Pracownik 3 konczy prace
Pracownik 2 konczy prace
Pracownik 1 konczy prace

```

Po zatrzymaniu pracy ciężarówka kontynuuje rozwożenie cegieł, co zajmuje trochę czasu ze względu na jej małą pojemność oraz 30-sekundowy czas rozwożenia cegieł.

```

Ciezarowka 1 podjezdza do tasmy
Ciezarowka 1 odebrala zostawiona cegle o masie 3
Obecny ładunek 3/3
Ciezarowka 1 zwrocila cegle o masie 3
Obecny ładunek 3/3
Ciezarowka 1 rozwozi cegly...
Ciezarowka 1 podjezdza do tasmy
Ciezarowka 1 odebrala zostawiona cegle o masie 3
Obecny ładunek 3/3
Ciezarowka 1 zwrocila cegle o masie 3
Obecny ładunek 3/3
Ciezarowka 1 rozwozi cegly...
Ciezarowka 1 podjezdza do tasmy
Ciezarowka 1 odebrala zostawiona cegle o masie 3
Obecny ładunek 3/3
Ciezarowka 1 zwrocila cegle o masie 1
Obecny ładunek 3/3
Ciezarowka 1 rozwozi cegly...
Ciezarowka 1 podjezdza do tasmy
Ciezarowka 1 odebrala zostawiona cegle o masie 1
Obecny ładunek 1/3
Ciezarowka 1 odebrala cegle o masie 1
Obecny ładunek 2/3
Ciezarowka 1 odebrala cegle o masie 1
Obecny ładunek 3/3
Ciezarowka 1 zwrocila cegle o masie 2
Obecny ładunek 3/3
Ciezarowka 1 rozwozi cegly...
Ciezarowka 1 podjezdza do tasmy
Ciezarowka 1 odebrala zostawiona cegle o masie 2
Obecny ładunek 2/3
Brak zostawionych cegiel: No message of desired type
Ciezarowka 1 rozwozi cegly...
Ciezarowka 1 konczy prace

Liczb cegiel: 0/20
Masa cegiel: 0/30

Kolejka komunikatów o ID 6783031 zostala usunieta.
Zbior semaforow o ID 5308462 zostal usuniety.
Zbior semaforow o ID 5308463 zostal usuniety.
jaros.antoni.151439@torus:~/projekt$ |

```

Po rozwiezeniu wszystkich cegieł symulacja się kończy

```
● jaros.antoni.151439@torus:~/projekt$ wc -l logs_workers.txt logs_trucks.txt
  17 logs_workers.txt
  17 logs_trucks.txt
  34 total
⊗ jaros.antoni.151439@torus:~/projekt$ ipcs | grep $(whoami)
○ jaros.antoni.151439@torus:~/projekt$
```

Wszystkie wyprodukowane cegły zostały rozwiezione, a zbiory semaforów i kolejka komunikatów zostały usnięte. Test zakończył się sukcesem.

4. Linki do istotnych fragmentów kodu

Tworzenie procesów (funkcje `fork()`, `exec()`)

https://github.com/Yume24/Systemy-operacyjne-projekt/blob/8172cc09d9e75a29c36888f8f6a08576fa67281b/dyspozytor_utils.c#L64-L92

https://github.com/Yume24/Systemy-operacyjne-projekt/blob/8172cc09d9e75a29c36888f8f6a08576fa67281b/dyspozytor_utils.c#L35-L61

Obsługa sygnałów (handlers, `signal()`, `kill()`)

<https://github.com/Yume24/Systemy-operacyjne-projekt/blob/8172cc09d9e75a29c36888f8f6a08576fa67281b/dyspozytor.c#L7-L44>

<https://github.com/Yume24/Systemy-operacyjne-projekt/blob/8172cc09d9e75a29c36888f8f6a08576fa67281b/dyspozytor.c#L68-L69>

Obsługa plików

https://github.com/Yume24/Systemy-operacyjne-projekt/blob/8172cc09d9e75a29c36888f8f6a08576fa67281b/pracownik_ciezarowka_utils.c#L192-L219

Tworzenie i obsługa semaforów

<https://github.com/Yume24/Systemy-operacyjne-projekt/blob/8172cc09d9e75a29c36888f8f6a08576fa67281b/utils.c#L19-L42>

https://github.com/Yume24/Systemy-operacyjne-projekt/blob/8172cc09d9e75a29c36888f8f6a08576fa67281b/pracownik_ciezarowka_utils.c#L143-L190

Tworzenie i obsługa kolejki komunikatów

<https://github.com/Yume24/Systemy-operacyjne-projekt/blob/8172cc09d9e75a29c36888f8f6a08576fa67281b/utils.c#L4-L16>

https://github.com/Yume24/Systemy-operacyjne-projekt/blob/8172cc09d9e75a29c36888f8f6a08576fa67281b/pracownik_ciezarowka_utils.c#L9-L35

https://github.com/Yume24/Systemy-operacyjne-projekt/blob/8172cc09d9e75a29c36888f8f6a08576fa67281b/pracownik_ciezarowka_utils.c#L38-L129