

# Sprawozdanie – Systemy Operacyjne

**Grupowa wykonawcza:** Prowadzący: dr inż. Marcin Koźniewski

Filip Rybiński

Dominik Gudalewski

**Temat projektu:**

**Data: 21.06.2022r.**

Temat 1 – Wąski most

**Treść zadania:**

Z miasta A do miasta B prowadzi droga, na której znajduje się wąski most umożliwiający tylko ruch jednokierunkowy. Most jest również dość słaby, także może po nim przejeżdżać tylko jeden samochód na raz. Napisać program w którym N samochodów (wątków) będzie nieustannie przejeżdżało z miasta do miasta, pokonując po drodze most (N przekazywane jako argument linii poleceń). "Miasto" jest to funkcja, którą wątki mogą wykonywać niezależnie od siebie przez krótki, losowy czas (maks. kilka sekund). Zsynchronizuj dostęp wątków do mostu:

- nie wykorzystując zmiennych warunkowych (tylko mutexy/semafony) [17 p]
- wykorzystując zmienne warunkowe (condition variables) [17 p]

Aby móc obserwować działanie programu, każdemu samochodowi przydziel numer. Program powinien wypisywać komunikaty według poniższego przykładu:

**A-5 10>>> [>> 4 >>] <<< 4 6-B**

Oznacza to, że po stronie miasta A jest 15 samochodów z czego 10 czeka w kolejce przed mostem, przez most przejeżdża samochód z numerem 4 z miasta A do B, po stronie miasta B jest 10 samochodów, z czego 4 oczekują w kolejce przed mostem. Po uruchomieniu programu z parametrem -debug należy wypisywać całą zawartość kolejek po obu stronach mostu, nie tylko ilość samochodów. Komunikat należy wypisywać w momencie, kiedy w programie zmieni się którakolwiek z tych wartości.

**Opis poszczególnych modułów programu:**

**Zmienne globalne:**

```
int In_A=0,Que_A=0,In_B=0,Que_B=0; // POMOCCNICZE //
int N=10; // ILOSC SAMOCHODOW //
int debug=0; // pokazywanie kolejki //
int clear=0; // ładniejsze wypisywanie na konsole.//
int mode=0; // 0- bez zmiennych warunkowych // mode=1 zmienien warunkowe
int ticket=0; // zmienna ticket dla samochodow//
pthread_mutex_t bridge = PTHREAD_MUTEX_INITIALIZER; // INIT //
pthread_cond_t ticket_var=PTHREAD_COND_INITIALIZER; // INIT //
int *carStatus; // STATUS SAMOCHODU W JAKIM STANIE SIE ZNAJDUJE //
int *carDirection; // KIERUNEK W KTORYM JEDZIE SAMOCHOD //
```

## Opis:

Program działający bez zmiennych warunkowych tworzy  $n$  wątków które przejeżdżają z miasta do miasta, po dojechaniu ich do nowego miasta watek odczekuje chwile po czym przechodzi znowu do kolejki i zmienia kierunek w którym chce jechać na nowy. Np. Jechał z miasta A do B dojeżdża chwile jest w mieście B po czym przechodzi do kolejki w mieście B i ustawia swój kierunek na Miasto A W trybie debug program działa tam samo poza tym że wyświetla wątki które są w kolejce w mieście A i B. W trybie clear program czyści okno terminala, aby wszystko wyglądało lepiej :/ średnio wyszło W trybie mode program używa dodatkowo zmiennej warunkowej która oczekuje na otrzymanie biletu przez wątek który aktualnie chce przejechać przez most. Po otrzymaniu biletu wątek działa jak w podstawowej wersji programu.

## Poprawne użycie programu:

- Składnia programu (bez debuga, bez zmiennych warunkowych, bez clear)= ./program -n [Ilość samochodów]
- Składnia programu (bez zmiennych warunkowych, bez clear)= ./program -n [Ilość samochodów] -d
- Składnia programu (bez zmiennych warunkowych)= ./program -n [Ilość samochodów] -d -c
- Składnia programu = ./program -n [Ilość samochodów] -d -c -m

## Opis parametrów:

- -n : Ilość przejeżdżających samochodów // ilość wątków
- -d : wyświetlanie zawartości kolejek
- -m : działanie na zmiennych warunkowych / bez biletu samochód nie przejedzie ani nie zmieni swojego stanu
- -c : clear czyści konsolę po każdej komunikacji / trochę ładniej wygląda :)

## Opisy kluczowych funkcji w kodzie:

```
void *Drive(void* arg){
    int val= *((int*)arg); //ID SAMOCHODU
    while(1){
        //LOCKUJE
        if (pthread_mutex_lock(&bridge) != 0){ // lockuje watek
            printf("Error");
        }
        if(mode==1){ // tryb na zmiennych to czeka na otrzymanie biletu ! ze stacji
            while(ticket==0){
                printf("No ticket to drive\n"); //nie ma biletu czeka na bilet
                pthread_cond_wait(&ticket_var, &bridge);
            }
            ticket=0; //otrzymał bilet i go używa na przejechanie jeśli jest w kolejce //jesli nie jest to go traci po prostu
            printf("Now can drive \n");
        }
        printer(val); // wypisuje calosc
        if (pthread_mutex_unlock(&bridge) != 0){ //unlockuje watek
            printf("Error");
        }
        //UNLOCKUJE
        //CZEKAJ
        int tmp; // zmienna pomocnicza
        tmp=rand()%500 +1500;
        usleep(tmp* 1000); // losowy czas czekania
        //JESLI KTORYS Z WATKOW ZNAJDOMAL SIE W MIESCIE ALE NIE UCZESTNICZYŁ W KOLEJCE TO DODAJE GO DO KOLEJKI I WYBIERA KIERUNEK
        //DODAJ DO KOLEJKI SAMOCHOD KTORY NIE BYŁ W KOLEJCE
        if(carDirection[val-1]==NONE){ //jesli samochod dojechal do miasta wybranego wcześniej i nie ma nowego kierunku to ustawia B LUB A
            if(carStatus[val-1]==CITY_A){ // JESLI W MIESCIE A
                carDirection[val-1]=CITY_B; // TO AKTUALNY KIERUNEK TO B
                carStatus[val-1]=IN_QUEUE_A; // DODAJE ZNOWU DO KOLEJKI
                add_to_list(&Que_AA,val); // DODAJE DO KOLEJKI W A
            }else{
                carDirection[val-1]=CITY_A; // JESLI W MIESCIE B TO AKTUALNY KIERUNEK A
                carStatus[val-1]=IN_QUEUE_B; // DODAJE DO KOLEJKI B
                add_to_list(&Que_BB,val); // DODAJE DO KOLEJKI B
            }
        }
    }
    pthread_exit(0);
}
```

***Działanie programu:***

```

rybka@rybka-VivoBook-ASUSLaptop-X421IA-M433IA:~/Pulpit/projektso2$ ./program -n 10
KOLEJKI I POCZATEK PROGRAMU
A-3 [ 7 ] >> [ ] << [ 0 ] 0-B
A-3 [ 4 5 6 7 8 9 10 ] >> [ ] << [ ] 0-B
A-3 7>>> [ ] <<<0 0-B---W MIESCIE ALE NIE UCZESTNICZY W KOLEJCE // CZYLI NIE JEDZIE
A-3 7>>> [ ] <<<0 0-B---W MIESCIE ALE NIE UCZESTNICZY W KOLEJCE // CZYLI NIE JEDZIE
A-3 7>>> [ ] <<<0 0-B---W MIESCIE ALE NIE UCZESTNICZY W KOLEJCE // CZYLI NIE JEDZIE
A-2 7>>> [>> 4 >>] <<<0 0-B--JEDZIE
A-1 8>>> [ ] <<<0 1-B---DOJECHAL DO MIASTA
A-0 8>>> [>> 5 >>] <<<0 1-B--JEDZIE
A-0 8>>> [ ] <<<0 2-B---DOJECHAL DO MIASTA
A-0 7>>> [>> 7 >>] <<<1 1-B--JEDZIE
A-0 7>>> [ ] <<<1 2-B---DOJECHAL DO MIASTA
A-0 6>>> [>> 6 >>] <<<2 1-B--JEDZIE
A-0 6>>> [ ] <<<2 2-B---DOJECHAL DO MIASTA
A-0 5>>> [>> 8 >>] <<<3 1-B--JEDZIE
A-0 5>>> [ ] <<<3 2-B---DOJECHAL DO MIASTA
A-0 4>>> [>> 9 >>] <<<4 1-B--JEDZIE
A-0 4>>> [ ] <<<4 2-B---DOJECHAL DO MIASTA
A-0 3>>> [>> 10 >>] <<<5 1-B--JEDZIE
A-0 3>>> [ ] <<<5 2-B---DOJECHAL DO MIASTA
A-0 2>>> [>> 2 >>] <<<6 1-B--JEDZIE
A-0 2>>> [ ] <<<6 2-B---DOJECHAL DO MIASTA
A-0 1>>> [>> 1 >>] <<<7 1-B--JEDZIE
A-0 1>>> [ ] <<<7 2-B---DOJECHAL DO MIASTA
A-0 0>>> [>> 3 >>] <<<8 1-B--JEDZIE
A-0 0>>> [ ] <<<8 2-B---DOJECHAL DO MIASTA
A-0 0>>> [<< 4 <<] <<<8 1-B--JEDZIE
A-1 0>>> [ ] <<<8 1-B---DOJECHAL DO MIASTA
A-1 0>>> [<< 5 <<] <<<8 0-B--JEDZIE

```

```

rybka@rybka-VivoBook-ASUSLaptop-X421IA-M433IA: ~/Pulpit/projektso2
A-2 [ 5 6 7 8 9 10 1 ]>>> [>> 4 >>] <<<[ ] 0-B--JEDZIE

```

```

rybka@rybka-VivoBook-ASUSLaptop-X421IA-M433IA:~/Pulpit/projektso2$ ./program -n 10 -d -m
KOLEJKI I POCZATEK PROGRAMU
A-3 [ 7 ] >> [ ] << [ 0 ] 0-B
A-3 [ 4 5 6 7 8 9 10 ] >> [ ] << [ ] 0-B
No ticket to drive
No ticket to drive
No ticket to drive
Now can drive
A-3 [ 4 5 6 7 8 9 10 ]>>> [ ] <<<[ ] 0-B---W MIESCIE ALE NIE UCZESTNICZY W KOLEJCE // CZYLI NIE JEDZIE
Now can drive
A-3 [ 5 6 7 8 9 10 ]>>> [>> 4 >>] <<<[ ] 0-B--JEDZIE
A-3 [ 5 6 7 8 9 10 ]>>> [ ] <<<[ ] 1-B---DOJECHAL DO MIASTA
No ticket to drive
No ticket to drive
No ticket to drive
No ticket to drive
No ticket to drive
No ticket to drive
No ticket to drive
No ticket to drive
Now can drive
A-2 [ 5 6 7 8 9 10 3 ]>>> [ ] <<<[ ] 1-B---W MIESCIE ALE NIE UCZESTNICZY W KOLEJCE // CZYLI NIE JEDZIE
No ticket to drive
Now can drive
A-2 [ 5 6 7 8 9 3 ]>>> [>> 10 >>] <<<[ ] 1-B--JEDZIE
A-2 [ 5 6 7 8 9 3 ]>>> [ ] <<<[ 4 ] 1-B---DOJECHAL DO MIASTA
No ticket to drive
No ticket to drive
No ticket to drive
No ticket to drive
No ticket to drive
No ticket to drive
Now can drive
A-1 [ 5 6 7 9 3 1 ]>>> [>> 8 >>] <<<[ 4 ] 1-B--JEDZIE
A-1 [ 5 6 7 9 3 1 ]>>> [ ] <<<[ 4 ] 2-B---DOJECHAL DO MIASTA

```