Artur Grudkowski 283738

SOI

Lab 3 – Semafory

1. **Koncepcja**

Projekt realizacji urzędu będzie się opierał o cztery kolejki priorytetowe z czego dwie pierwsze po prostu nie będą zwracać uwagi na priorytet. Każdej kolejce będzie przypisany mutex zapewniający synchronizację między procesami korzystającymi z danej kolejki. Ponad to, dwie pierwsze kolejki będą posiadały po semaforze pilnującym aby urzędnicy nie brali petentów z pustej kolejki, nie jest potrzebny semafor, który będzie sprawdzał czy kolejka nie jest pełna ponieważ, będę wywoływał na poprzednim semaforze funkcji sprawdzającą stan danego semafora. Analogiczna sytuacja będzie miała miejsce w następnych dwóch kolejkach, jednakże tym razem będę sprawdzał czy urzędnicy drugiej warstwy nie kierują petentów do pełnej kolejki. Poza tym, przewiduje użycie jednego semafora na potrzeby urzędnika 0, który będzie sprawdzał ilość petentów w kolejkach A i B (A – kolejka między urzędnikiem 0 a pierwszym urzędnikiem 1, analogicznie B dla drugiego urzędnika), jeśli zmienna w semaforze przekroczy wartość 20 to proces urzędnika 0 będzie wstrzymywany na tym semaforze. Podobna sytuacja będzie miała miejsce w przypadku urzędnika 2, który będzie korzystał z semafora sprawdzającego czy są jakieś elementy w kolejkach C i D ( analogicznie do nazewnictwa w poprzedniej sytuacji ).

1. **Symulalcja**

Utworze jeszcze dodatkowy mutex do obsługi terminala. Sama funkcja korzystając a z terminala będzie w wypisywać zwartość wszystkich kolejek w mocno uproszczonym 2D korzystając z tabeli znaków ASCII.

**Struktura kolejki**

struct Queue{

int \*queue;

int start, end;

};

* **\*queue** – wskaźnik na tablicę dwuwymiarową, gdzie w kolumnie będzie numer petenta oraz jego priorytet ( 0 – niski, 1 – wysoki ). Sama tabela będzie się znajdowała w pamięci współdzielonej.

1. **Pliki**

Planuję stworzyć dwie biblioteki, jedna do obsługi kolejki a drugą upraszczającą operacje na semaforach.

* queue.h – plik nagłówkowy zawierający strukturę kolejki oraz deklarację funkcji na kolejkach
* queue.c – plik zawierający definicje funkcji z pliku queue.h
* sem.h – plik nagłówkowy zawierający deklaracje funkcji wykonywanych na semaforze
* sem.c – plik zawierający definicję funkcji z pliku sem.h
* main.c – plik zawierający procesy urzędników, czyli symulujący urząd