

Projekt 3

Zadanie 3.8 Symulator linii produkcyjnej układów scalonych

Kamil Kaczkowski

PROI 2018/2019

1. Założenia

W programie użyłem wyimaginowanej jednostki czasu, dzięki której operacje wykonują się co 5 jednostek. Główne elementy w programie to dwa wektory, wektor SpisStanowisk oraz SpisPodlozy, które przetrzymują odpowiednio wszystkie Stanowiska oraz wszystkie Podłoża, które powstały w trakcie pracy programu. Pierwszy element wektora SpisPodlozy, to Podłoże wyimaginowane o ID = -1, nie jest ono wykorzystywane w żadnym momencie w programie, podobnie jak inne elementy tego typu(wyimaginowane). Jest to jedynie wektor pomocny przy zabezpieczeniu przed przedwczesnym zwolnieniem całego wektora.

2. Podział na klasy

Klasa Stanowisko:

POLA:

String nazwa – symbolizuje nazwę danego stanowiska

int CzasTrwania – określa ile czasu potrzebuje stanowisko, by wykonać prace

int CzasAktualnyProces – określa ile czasu upłynęło od rozpoczęcia procesu.

vector <Podloze> oczekujace – wektor podłoży oczekujących na obsługę na danym podłożu, pierwszy element wektora jest wyimaginowanym podłożem z ID = -1, zapobiega to usunięciu wektora w całości.

vector <Podloze> Zakonczone- wektor podłoży, które były już na tym stanowisku i nad którymi została wykonana praca. Podobnie jak wyżej pierwszy element to wyimaginowany wektor

bool available – określa czy dane stanowisko w tym momencie pracuje, czy stoi nieużywane

METODY:

dodaj_do_kolejki(vector <Podloze> &SpisPodlozy) – Funkcja dodająca do wektora oczekujace

pobierz_z_kolejki() – funkcja pobierająca podłoże do obsługi

koniec_procesu(vector <Podloze> &SpisPodlozy) – funkcja kończąca pracę nad danym podłożem gdy minie czas

przechwytywanie_czasu() – Funkcja odpowiadająca za odliczanie czasu do końca pracy nad danym podłożem

FUNKCJE ZAPRZYJAZNIONE:

ostream & operator<<(std::ostream &wyjscie, const Stanowisko &s)- przeciążenie operatora wyjścia dla klasy

Klasa Podłoże:

POLA:

vector <string> DoPrzejscia – wektor określający przez jakie stanowiska musi przejść podłoże w produkcji. Pierwszy wektor jest wyimaginowany.

vector <string> Zakonczone- wektor przechowujący dane na temat przebytych stanowisk podczas produkcji. Pierwszy wektor jest wyimaginowany.

bool CzyUzywane- określa czy dane podłoże, jest aktualnie używane.

bool Wprodukcji – określa czy dane podłoże trafiło do produkcji.

int ID – Numeruje podłoża, by można było je zidentyfikować.

METODY:

Podloze(int identyfikator) – konstruktor klasy, tworzy podłoże o id podanym przez program

Przenies_do_zakonczonych() – Po ukończeniu etapu, przenosi go do wektora zakończonych etapów

dodaj_do_przebiegu(vector <Stanowisko> &SpisStanowisk)- Dodaje do przebiegu kolejne stanowisko. Można to zrobić jedynie gdy podłoże nie trafiło jeszcze na linię produkcyjną.

FUNKCJE ZAPRZYJAZNIONE:

Wyswietl_podloza_ID(vector <Podloze> SpisPodlozy) – Wyświetla jedynie ID wszystkich podłoży

dodaj_do_spisu_podlozy(int identyfikator) -Dodaje podłoże do wektora przechowującego wszystkie podłoża

ostream & operator<<(std::ostream &wyjscie, const Podloze &s) – przeciążenie operatora wyjścia dla klasy

3. Interfejs

Program działa w pętli, za każdym razem ukazując nam MENU z 10 opcjami do wyboru, którymi sterujemy wpisując cyfry z klawiatury.

```
-----MENU-----
1. Utworz nowe podloze
2. Dodaj krok produkcyjny dla podloza
3. Zapisz do pliku historie podlozy
4. Zapisz do pliku historie urzadzen
5. Wyszwietl podloza i ich kroki produkcyjne
6. Wyszwietl wybrane podloze i kroki jego produkcji
7. Przewin czas o 5 jednostek
0. Zakoncz dzialanie programu
Twój wybór: 1
-----MENU-----
```

4. Schemat działania

Program działa w pętli, którą steruje użytkownik poprzez wprowadzanie cyfr w menu wyboru. Tworzy on Podłoża o dowolnym przebiegu, z ograniczeniem co do stanowiska początkowego oraz końcowego, do momentu podania stanowisk na taśmę produkcyjną. Podanie stanowisk na taśmę produkcyjną, nie uniemożliwia tworzenia nowych stanowisk i edytowania tych nowych. Program dodaje stanowiska do kolejki, a następnie pobiera pierwsze stanowisko do pracy nad nim. Po wykonaniu pracy, używa odpowiednich metod klasy, by przenieść stanowisko do odpowiednich wektorów klasy. Następnie pobiera kolejne podłoże, zakolejkowując jednocześnie podłoże, które ukończyło dane stanowisko, do innego stanowiska. Program działa w tym trybie, tyle ile użytkownik chce, w każdym momencie może dołożyć nowe podłoże.