

Projekt na Systemy Operacyjne, Kolejowanie Procesów
1.11.012

Wygenerowano przez Doxygen 1.8.20

1 Indeks przestrzeni nazw	1
1.1 Lista przestrzeni nazw	1
2 Indeks klas	3
2.1 Lista klas	3
3 Dokumentacja przestrzeni nazw	5
3.1 Dokumentacja przestrzeni nazw generator	5
3.1.1 Opis szczegółowy	5
3.2 Dokumentacja przestrzeni nazw main	5
3.2.1 Opis szczegółowy	6
3.3 Dokumentacja przestrzeni nazw planista	6
3.3.1 Opis szczegółowy	6
3.4 Dokumentacja przestrzeni nazw procesor	6
3.4.1 Opis szczegółowy	7
3.5 Dokumentacja przestrzeni nazw procesy	7
3.5.1 Opis szczegółowy	7
4 Dokumentacja klas	9
4.1 Dokumentacja klasy generator.czas_przybycia_konfiguracja	9
4.1.1 Opis szczegółowy	9
4.2 Dokumentacja klasy procesy.Proces	9
4.2.1 Opis szczegółowy	10
4.2.2 Dokumentacja konstruktora i destruktor	10
4.2.2.1 __init__()	10
4.3 Dokumentacja klasy procesor.Procesor	10
4.3.1 Opis szczegółowy	11
4.3.2 Dokumentacja konstruktora i destruktor	11
4.3.2.1 __init__()	11
Indeks	13

Rozdział 1

Indeks przestrzeni nazw

1.1 Lista przestrzeni nazw

Tutaj znajdują się wszystkie udokumentowane przestrzenie nazw wraz z ich krótkimi opisami:

generator	Zajmuje się generowaniem oraz wczytywaniem danych do programu	5
main	Plik startowy main	5
planista	Zawiera czyste algorytmy kolejkowania procesów	6
procesor	Zawiera klasę Procesor	6
procesy	Plik zawiera klasę procesu	7

Rozdział 2

Indeks klas

2.1 Lista klas

Tutaj znajdują się klasy, struktury, unie i interfejsy wraz z ich krótkimi opisami:

generator.czas_przybycia_konfiguracja	
Enum do wyboru czasu brzybycia STALY/LOSOWY	9
procesy.Proces	
Klasa procesu	9
procesor.Procesor	
Klasa symulująca działanie procesora	10

Rozdział 3

Dokumentacja przestrzeni nazw

3.1 Dokumentacja przestrzeni nazw generator

Zajmuje się generowaniem oraz wczytywaniem danych do programu.

Komponenty

- class `czas_przybycia_konfiguracja`
Enum do wyboru czasu przybycia STALY/LOSOWY.

Funkcje

- def `menu` ()
Funkcja wypisuje interaktywne menu na konsolę oraz obiera dane wejściowe.
- def `setSeed` (sd)
Ustawia seed na generatorze liczb losowych.
- def `generuj_procesy` (ilość, średnia_długość_wykonywania=10, odchylenie_standardowe_czasu_↔ wykonywania=5, generuj_raport=True, quiet=False, czas_przybycia_config=czas_przybycia_konfiguracja.↔ LOSOWY)
Funkcja generuje procesy o losowych czasach przybycia oraz wykonywania.
- def `załaduj_procesy` (plik)
Funkcja wczytuje procesy z pliku csv.

3.1.1 Opis szczegółowy

Zajmuje się generowaniem oraz wczytywaniem danych do programu.

3.2 Dokumentacja przestrzeni nazw main

Plik startowy main.

Funkcje

- def `main` ()
Funkcja main wywołująca menu z generatora oraz wywołująca pętlę CPU.
- def `generuj_raport` (lista_procesow, kolejnosc, algorytm)
Funkcja tworząca raport:
- def `policz_sredni_czas_oczekiwania` (lista_procesow)
Funkcja licząca średni czas oczekiwania dla listy procesów po symulacji.
- def `max_czas_reakcji` (lista_procesow)
Funkcja wyliczająca maksymalny czas reakcji dla listy procesów.
- def `porównaj_algorytmy` (kwant_czasu)
Funkcja która dokonuje porównania obu algorytmów.

3.2.1 Opis szczegółowy

Plik startowy main.

3.3 Dokumentacja przestrzeni nazw planista

Zawiera czyste algorytmy kolejkowania procesów.

Funkcje

- def `SJF` (kolejka)
Algorytm "Shortest Job First (niewyłączający)".
- def `RR` (kolejka, czas, kwant_czasu)
Algorytm "Round Robin (wyłączający)".

3.3.1 Opis szczegółowy

Zawiera czyste algorytmy kolejkowania procesów.

3.4 Dokumentacja przestrzeni nazw procesor

Zawiera klasę `Procesor`.

Komponenty

- class `Procesor`
Klasa symulująca działanie procesora.

3.4.1 Opis szczegółowy

Zawiera klasę `Procesor`.

3.5 Dokumentacja przestrzeni nazw procesy

Plik zawiera klasę procesu.

Komponenty

- class `Proces`
Klasa procesu.

3.5.1 Opis szczegółowy

Plik zawiera klasę procesu.

Inicjalizuje się ją podając:

- PID
- `czas_przybycia` procesu
- `czas wykonania`

Rozdział 4

Dokumentacja klas

4.1 Dokumentacja klasy generator.czas_przybycia_konfiguracja

Enum do wyboru czasu przybycia STALY/LOSOWY.

Statyczne atrybuty publiczne

- int **LOSOWY** = 0
- int **STALY** = 1

4.1.1 Opis szczegółowy

Enum do wyboru czasu przybycia STALY/LOSOWY.

Dokumentacja dla tej klasy została wygenerowana z pliku:

- generator.py

4.2 Dokumentacja klasy procesy.Proces

Klasa procesu.

Metody publiczne

- def `__init__` (self, PID, czas_przybycia, [czas_wykonywania](#))
Funkcja inicjalizująca [Proces](#), należy podać:
- def `uśpij` (self)
Funkcja która przenosi proces w tryb uśpienia.
- def `wybudź` (self)
Funkcja która wybudza proces.
- def `czas_wykonywania` (self)
Funkcja która zwraca wyliczony czas wykonywania.
- def `tick` (self)
Funkcja wykonuje "krok" działania procesu.
- def `info` (self)
Funkcja wypisuje informacje o procesie na STDOUT.

Atrybuty publiczne

- `PID`
- `czas_przybycia`
- `całkowity_czas_wykonywania`
- `pozostały_czas_wykonywania`
- `czas_oczekiwania`
- `czas_reakcji`
- `był_wywłaszczony`
- `zakończony`
- `uśpiony`

4.2.1 Opis szczegółowy

Klasa procesu.

4.2.2 Dokumentacja konstruktora i destruktora

4.2.2.1 `__init__()`

```
def procesy.Proces.__init__ (
    self,
    PID,
    czas_przybycia,
    czas_wykonywania )
```

Funkcja inicjalizująca `Proces`, należy podać:

- `PID`
- `czas_przybycia` procesu
- `czas wykonania`

Dokumentacja dla tej klasy została wygenerowana z pliku:

- `procesy.py`

4.3 Dokumentacja klasy `procesor.Processor`

Klasa symulująca działanie procesora.

Metody publiczne

- def `__init__` (self, lista_procesow, algorytm_kolejkowania, kwant_czasu=4, debug=False)
Funkcja inicjalizująca procesor.
- def `kolejkowanie` (self)
Funkcja dokonuje sortowania kolejki wybranym algorytmem.
- def `kolejka_aktualizacja` (self)
Funkcja dokonuje aktualizacji kolejki (dodaje nowo uruchomione procesy) oraz wywołuje kolejkowanie.
- def `pętla` (self)
Główna pętla procesora.

Atrybuty publiczne

- `czas`
- `kolejka`
- `lista_procesow`
- `zakonczone_procesy`
- `algorytm_kolejkowania`
- `aktualny_proces`
- `kolejnosc`
- `kwant_czasu`
- `debug`

4.3.1 Opis szczegółowy

Klasa symulująca działanie procesora.

4.3.2 Dokumentacja konstruktora i destruktora

4.3.2.1 `__init__()`

```
def procesor.Procesor.__init__ (
    self,
    lista_procesow,
    algorytm_kolejkowania,
    kwant_czasu = 4,
    debug = False )
```

Funkcja inicjalizująca procesor.

Jako argumenty bierze listę procesów i wybrany algorytm kolejkowania jako string, który może przyjmować wartości "SJF" lub "RR"

Dokumentacja dla tej klasy została wygenerowana z pliku:

- `procesor.py`

Indeks

`__init__`

 procesor.Procesor, [11](#)

 procesy.Proces, [10](#)

generator, [5](#)

generator.czas_przybycia_konfiguracja, [9](#)

main, [5](#)

planista, [6](#)

procesor, [6](#)

procesor.Procesor, [10](#)

 __init__, [11](#)

procesy, [7](#)

procesy.Proces, [9](#)

 __init__, [10](#)