

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	5
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	6
3.5 Dokumentacja przestrzeni nazw procesy	6
3.5.1 Opis szczegółowy	6
4 Dokumentacja klas	7
4.1 Dokumentacja klasy procesy.Proces	7
4.1.1 Opis szczegółowy	7
4.1.2 Dokumentacja konstruktora i destruktor	8
4.1.2.1 __init__()	8
4.2 Dokumentacja klasy procesor.Procesor	8
4.2.1 Opis szczegółowy	9
4.2.2 Dokumentacja konstruktora i destruktor	9
4.2.2.1 __init__()	9
Indeks	11

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	Klasa procesu	6

Rozdział 2

Indeks klas

2.1 Lista klas

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

procesy.Proces	
Klasa procesu	7
procesor.Procesor	
Klasa symulująca działanie procesora	8

Rozdział 3

Dokumentacja przestrzeni nazw

3.1 Dokumentacja przestrzeni nazw generator

Zajmuje się generowaniem oraz wczytywaniem danych do programu.

Funkcje

- def `menu` ()
Funkcja wypisuje interaktywne menu na konsolę oraz obiera dane wejściowe.
- def `generuj_procesy` (ilość)
- def `załaduj_procesy` (plik)

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.

3.2.1 Opis szczegółowy

Plik startowy main.

3.3 Dokumentacja przestrzeni nazw planista

Zawiera czyste agorytmy kolejkowania procesów.

Funkcje

- def [SJF](#) (kolejka)
Algorytm "Shortest Job First (niewywłaszczający)".
- def [RR](#) (kolejka, czas, kwant_czasu)
Algorytm "Round Robin (wywłaszczający)".

3.3.1 Opis szczegółowy

Zawiera czyste agorytmy 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

Klasa procesu.

Komponenty

- class [Proces](#)
Klasa procesu.

3.5.1 Opis szczegółowy

Klasa procesu.

Inicjalizuje się ją podając:

- PID
- czas_przybycia procesu
- czas wykonania

Rozdział 4

Dokumentacja klas

4.1 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`
- `zakończony`
- `uśpiony`

4.1.1 Opis szczegółowy

Klasa procesu.

4.1.2 Dokumentacja konstruktora i destruktora

4.1.2.1 `__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`

4.2 Dokumentacja klasy `procesor.Processor`

Klasa symulująca działanie procesora.

Metody publiczne

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

Atrybuty publiczne

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

4.2.1 Opis szczegółowy

Klasa symulująca działanie procesora.

4.2.2 Dokumentacja konstruktora i destruktora

4.2.2.1 `__init__()`

```
def procesor.Procesor.__init__ (
    self,
    lista_procesow,
    algorytm_kolejkowania )
```

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](#), [9](#)
 [procesy.Proces](#), [8](#)

[generator](#), [5](#)

[main](#), [5](#)

[planista](#), [6](#)
[procesor](#), [6](#)
[procesor.Procesor](#), [8](#)
 [__init__](#), [9](#)
[procesy](#), [6](#)
[procesy.Proces](#), [7](#)
 [__init__](#), [8](#)