

Task Analizator

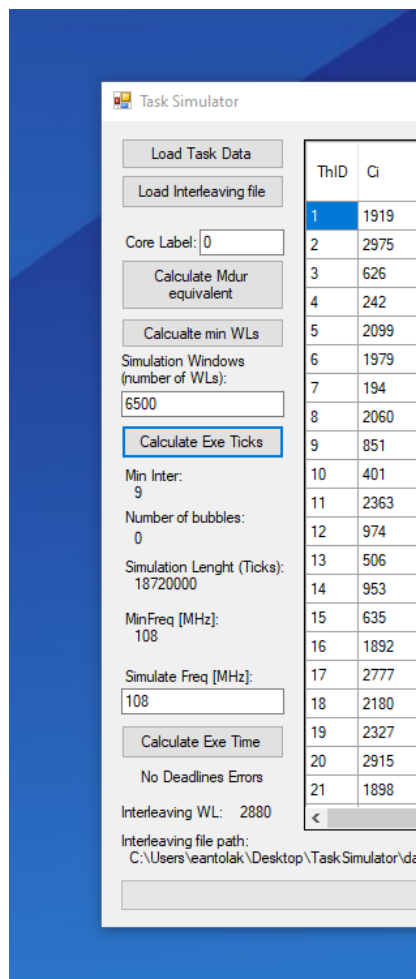
Formaty plików:

.td.txt – plik wejściowy z pierwotnymi danymi zadań, dane są separowane średnikiem, wg poniższego klucza: //ThID;Ci;Mi;Di;Mdur;SH;MH

.il.txt – plik zawierający kolejność przetwarzanych zadań. Dane są w nim zapisywane linia po linii. Pierwsza linia zawiera informacje o długości tego ciągu, każda następna to identyfikator przeplatanego zadania.

.ta.txt – plik zawierający dane zadań po procesie podziału zadań (zadania są podzielone na rdzenie). Każdy plik dotyczy tylko jednego rdzenia, jest on potrzebny do etapu szeregowania. Analizator nie wykorzystuje tych plików.

Obsługa:



Dwa pierwsze przyciski służą do wczytania plików zadań i przeplotu (.td.txt i .ta.txt).

„Caluclate Mdur equivalent” przelicza, ile potrzeba standardowych instrukcji (minimalnie i maksymalnie) do wykonania jednej instrukcji wymiany danych, każdego z analizowanych zadań.

„Caluclate min WLS” jest przyciskiem opcjonalnym, jego wciśnięcie powoduje obliczenie minimalnej długości symulacji (tak, aby każde zadanie wykonało się przynajmniej dwa razy) i wpisanie jej do pola niżej. Wartość tą można również wpisać ręcznie.

„Calculate Exe Ticks” powoduje obliczenie liczby taktów zegara potrzebnych do wykonania każdego zadania. Obliczana jest również minimalna częstotliwość, przy której wszystkie zadania zostaną wykonane przed swoim czasem deadline (MinFreq). Minimalny przeplot tego samego zadania znaleziony w pliku .il.txt (Min Inter). Liczba taktów w których nie są wykonywane żadne instrukcje (Number of bubbles) oraz długość symulacji wyrażona w cyklach zegara (Simulation Lenght).

„Calculate Exe Time” oblicza czas wykonania zadań na podstawie częstotliwości podanej w „Simulate Freq”. Jeżeli wszystkie zadania zostaną wykonane przed ich czasem deadline, pod przyciskiem pojawi się napis: „No Deadline Errors”.

Generalnie najlepiej tych przycisków używać w następującej kolejności: (Load Task -> Load Interleaving file -> Caluclate Mdur equivalent -> Caluclate min WLS (opcjonalnie) -> Calculate Exe Ticks -> Calculate Exe Time)

W przypadku systemu z kilkoma rdzeniami, tylko raz wczytujemy dane zadań (.td.txt), a następnie dodajemy wszystkie kolejne rdzenie w następujący sposób:

Wczytanie pliku .il.txt pierwszego rdzenia -> uzupełnienie etykiety Core Label (etykietą tą będzie można rozróżniać poszczególne rdzenie w tabeli, jest to krok opcjonalny) -> Caluclate Mdur

equivalent -> Calculate min WLS (opcjonalnie) -> Calculate Exe Ticks -> Calculate Exe Time -> wczytanie pliku .il.txt następnego rdzenia -> uzupełnienie etykiety (opcjonalnie) -> Calculate Mdur equivalent -> Calculate min WLS (opcjonalnie) -> Calculate Exe Ticks -> itd. Aż do załadowania wszystkich rdzeni.

Obsługa z linii poleceń:

Pod komendą „help”, można znaleźć krótkie wyjaśnienie wszystkich używanych komend. Komendy mogą być podawane w dowolnej kolejności. Komendy te rozdzielane są spacją. Komendą guioff można wyłączyć włączanie się trybu okienkowego aplikacji.

```
C:\Users\eantolak\Desktop\TaskSimulator> TaskSimulator.exe help
help - show help
interpath= - path to interleaving file (.il.txt)
taskpath= - path to task data file (.td.txt)
wlnum= Simulation Windows (numbers of WLS), if not specified, the minimum WLS value will be calculated automatically
showbubble - show number of bubbles in interleaving file
showminfreq - show the minimum frequency that meets the deadlines times in [MHz]
showmininter - show the minimal interling in interleaving file (MinIndistance)
showresult - show if all tasks were completed before the deadline. This command has snes only when typing SimFreq manually
(result: meetdeadline=1 or meetdeadline=0)
showwls - show Simulation Windows (numbers of WLS)
showsimlenght - show simulaion lenght in Ticks
guioff - do not open GUI
simfreq= - simulation frequency, if this parameter is not specified, the calculated minimum frequency will be used
```

Przykładowo komendą:

```
TaskSimulator taskpath=C:\Users\eantolak\Desktop\TaskSimulator\dane_doktorat\zadania.td.txt
interpath=C:\Users\eantolak\Desktop\TaskSimulator\dane_doktorat\minres_c0.il.txt showminfreq
guioff
```

Można obliczyć minimalną częstotliwość, przy której wszystkie zadania wykonają się na czas (wlnum zostaje policzone automatycznie)

```
C:\Users\eantolak\Desktop\TaskSimulator>TaskSimulator taskpath=C:\Users\eantolak\Desktop\TaskSimulator\dane_doktorat\zadani
a.td.txt interpath=C:\Users\eantolak\Desktop\TaskSimulator\dane_doktorat\minres_c0.il.txt showminfreq guioff
MinFreq=108
```

Przykładowo komendą:

```
TaskSimulator taskpath=C:\Users\eantolak\Desktop\TaskSimulator\dane_doktorat\zadania.td.txt
interpath=C:\Users\eantolak\Desktop\TaskSimulator\dane_doktorat\minres_c0.il.txt simfreq=80
showresult showmininter guioff
```

Można sprawdzić, czy wszystkie zadania zostaną wykonane na czas przy prędkości zegara równej 80MHz, dodatkowo komenda ta wypisuje również minimalny przepływ znaleziony w pliku il.txt.

```
C:\Users\eantolak\Desktop\TaskSimulator>TaskSimulator taskpath=C:\Users\eantolak\Desktop\TaskSimulator\dane_doktorat\zadani
a.td.txt interpath=C:\Users\eantolak\Desktop\TaskSimulator\dane_doktorat\minres_c0.il.txt simfreq=80 showresult showmininte
r guioff
MinInter=9 meetdeadline=0
```

Najmniejszy znaleziony przepływ w pliku .il.txt to 9. Meetdeadline=0 oznacza, że zadania nie wykonały się na czas. W takim wypadku można powtórzyć tą komendę bez fragmentu guioff i dokładnie sprawdzić w interfejsie co do tego doprowadziło.

W systemach wielordzeniowych należy sprawdzać wszystkie rdzenie osobno.

Do tego pliku dołączam również katalog „dane_doktorat” z przykładami z doktoratu.