

Wydział Informatyki Politechniki Białostockiej Przedmiot: Systemy Operacyjne	Data oddania: 12.05.2023r.
Projekt nr 1 Temat: Minicron Zespół: 1. Bartosz Karolczak 2. Paweł Niewierowski	Prowadzący: mgr inż. Daniel Reska

Dokumentacja projektu “minicron”

1. Sposób uruchamiania projektu:

Aby uruchomić projekt, należy użyć polecenia: **./minicron <taskfile> <outfile>**.

Plik **<taskfile>** zawiera zadania do wykonania przez demona zapisane w formacie:
<hour>:<minute>:<command>:<mode>

hour - godzina wykonania zadania

minutes - minuta wykonania zadania

command - komenda lub potok komend do wykonania przez program

mode - tryb zapisu wyniku zadania

Plik **<outfile>** zawiera wynik każdego wykonanego zadania wraz z informacją jakie zadanie zostało wykonane. Treść wyniku zadania różni się w zależności od zaznaczonego trybu zapisu. Wyróżniamy 3 tryby:

- 0 - do pliku **<outfile>** wpisujemy treść, jaką polecenie wypisało na standardowe wyjście (stdout)
- 1 - do pliku **<outfile>** wpisujemy treść, jaką polecenie wypisało na wyjście błędów (stderr).
- 2 - do pliku **<outfile>** wpisujemy treść, jaką polecenie wypisało na standardowe wyjście i wyjście błędów.

2. Sposób testowania projektu:

1. Skompiluj projekt, używając pliku Makefile i polecenia **make**.
2. Przygotuj plik z zadaniami (**taskfile**) oraz plik wyjściowy (**outfile**) - upewnij się, że mają różne nazwy.
3. Uruchom program, podając odpowiednie nazwy plików jako argumenty.
4. Sprawdź wyniki w pliku wyjściowym lub logach systemowych.

3. Lista zaimplementowanych funkcjonalności:

- **Wczytywanie zadań z pliku taskfile:** po inicjalizacji, demon wczytuje wszystkie zadania z pliku **taskfile** w swojej pamięci.
- **Posortowanie wczytanych zadań:** po wczytaniu zadań, demon sortuje listę zadań według czasu wykonania, jeśli godzina wykonania danego zadania jest wcześniejsza niż aktualna godzina, zadanie trafia na koniec listy.
- **Wywołanie zadań o określonym czasie:** demon odczytuje czas pierwszego zadania, oblicza czas na który ma zasnąć i zasypia. Po obudzeniu uruchamia proces potomny wykonujący komendę, a samemu usuwa zadanie z listy i jeżeli są kolejne komendy, ponawia działanie. Jeśli w liście nie ma kolejnych zadań, demon kończy pracę
- **Wykonanie komendy / potoku komend:** demon odczytuje wszystkie komendy z wywołanego zadania i wykonuje je w oddzielnych procesach, łącząc je ze sobą potokami aby umożliwić przepływ informacji między nimi.
- **Zapisywanie wyników do pliku outfile:** po zakończeniu wykonywania zadania demon zapisuje jego wynik w pliku **outfile** w zależności od podanego wcześniej trybu zapisu.
- **Obsługa sygnałów: SIGINT, SIGUSR1, SIGUSR2:**
 - **SIGINT:** Demon natychmiast kończy pracę
 - **SIGUSR1:** Demon porzuca niewykonane zadania i ponownie ładuje zadania z pliku **taskfile**
 - **SIGUSR2:** Demon wypisuje do logów systemowych listę pozostałych zadań do wykonania.
- **Logowanie informacji do pliku syslog:** przed wywołaniem każdego zadania demon wpisuje do logów systemowych informację o uruchomieniu danego zadania, po ukończeniu wykonywanego zadania demon wpisuje do logów systemowych informację o kodzie wyjścia danego zadania.

4. Opisy wybranych funkcji:

a) **addTask(char* taskString)**

Funkcja przyjmuje wskaźnik na ciąg znaków reprezentujący zadanie. Dodaje nowe zadanie do listy zadań typu **tasklist**, tworząc nową strukturę typu **task** dla zadania i ustawiając odpowiednie wskaźniki.

Parametry:

- **taskString:** wskaźnik na ciąg znaków reprezentujący zadanie.

Wartość zwracana:

- **Brak**

Przykład użycia:

```
char* taskString = "12:30:ls -l:0";
addTask(taskString);
```

b) readTaskFile(char* fileName)

Funkcja przyjmuje wskaźnik na ciąg znaków reprezentujący nazwę pliku **taskfile**. Otwiera go, odczytuje po kolei każde zadanie zapisane w tym pliku i dodaje je do pamięci za pomocą funkcji **addTask**.

Parametry:

- **fileName**: wskaźnik na ciąg znaków reprezentujący nazwę pliku do odczytania

Wartość zwracana:

- **Brak**

Przykład użycia:

```
char* taskfileName = "taskfile.txt";  
readTaskFile(taskfileName);
```