

W wyniku wykonania poniższych zadań powinien powstać **program**, którego kody źródłowe powinny zostać przesłane na adres antyplagiatu (informacja w ramce na końcu zadania).

Naszym zadaniem będzie napisanie programu, który pozwoli na przetestowanie obsługi przychodzących sygnałów.

Program uruchamiamy z dwoma parametrami liczbowymi:

- maksymalną długością życia procesów potomnych (sekundy) oraz
- przerwę pomiędzy ich tworzeniem (sekundy).

Program główny w pętli, w stałych odstępach czasu (podanych jako argument wywołania), tworzy nowy proces potomny. Proces potomny losuje liczbę z zakresu do podanego jako argument maksimum. Następnie wykonuje dowolne obliczenia (np. liczymy kolejne wartości silni) przez okres równy wylosowanej wartości (posłużyć się funkcją **alarm**). Proces potomny powinien obsłużyć sygnał **SIGALRM** i zakończyć się, zwracając jako kod powrotu wylosowaną wcześniej liczbę.

Posłużenie się sygnałami czasu rzeczywistego (**sigqueue**) pozwoli na ich kolejkowanie po stronie rodzica, gdyby jednocześnie kilka procesów potomnych kończyło działanie w tym samym czasie.

Każdy proces potomny po uruchomieniu wyświetla informację o sobie w jednej kolumnie (pid, wylosowana wartość, czas utworzenia). Główny program aktywnie kontroluje zakończenie procesów potomnych: po odebraniu informacji o zakończeniu potomka wyświetla w drugiej kolumnie otrzymane informacje (pid zakończonego procesu, kod zakończenia, czas zakończenia). Czekanie ma być zaimplementowane bez użycia funkcji z rodziny **wait** - należy wykorzystać fakt, że proces macierzysty otrzymuje sygnał **SIGCHLD** po zakończeniu potomka.

**Podpowiedź:** Należy użyć funkcji **sigaction**, zwracając szczególną uwagę na flagę **SA\_SIGINFO** i budowę struktury **siginfo\_t**.

Program działa w pętli tak długo, aż otrzyma sygnał **SIGINT**. Powinien wtedy zaczekać na zakończenie wszystkich uruchomionych do tej pory potomków, nie tworząc już nowych i zakończyć swoje działanie. Zwróć uwagę, żeby **SIGINT** wysyłane za pomocą kombinacji klawiszy **Ctrl-C** nie przerwało działania procesów potomnych (sygnał kierowany jest w takiej sytuacji do grupy procesów).

Przykładowy wynik działania programu (procesy potomne tworzone w 10 sekundowych odstępach, maksymalna długość działania procesu potomnego 20 sekund).

```
./testSig -w 10 -m 20
```

```
[ 1051 ] [ 3 ] [ Fri Mar 19 19:30:11 2021 ]
          [ 1051 ] [ 3 ] [ Fri Mar 19 19:30:14 2021 ]
[ 1052 ] [ 12 ] [ Fri Mar 19 19:30:21 2021 ]
[ 1053 ] [ 16 ] [ Fri Mar 19 19:30:31 2021 ]
^C
          [ 1052 ] [ 3 ] [ Fri Mar 19 19:30:34 2021 ]
          [ 1053 ] [ 3 ] [ Fri Mar 19 19:30:47 2021 ]
```