

Podobnie jak w ćwiczeniu poprzednim, głównym celem będzie napisanie programu (a dokładniej dwóch programów), które pozwolą na łamanie metodą słownikową przechwyconego hasła w formacie akceptowanym przez **crypt**. Zmieniamy jednak model obliczeń. Tym razem współpracować będzie ze sobą wiele procesów utworzonych z dwóch różnych programów. D komunikacji między procesami należy wykorzystać (do wyboru w wersji **System V** lub **POSIX**):

- kolejki komunikatów,
- pamięć współdzieloną,
- semaforey.

1. Pierwszy program (**Manager**) zarządza obliczeniami, czyli:

- przygotowuje dane, czyli m.in. odczytuje plik z hasłami albo mapuje go do pamięci współdzielonej (posługujemy się plikiem z poprzedniego ćwiczenia),
- tworzy kolejkę komunikatów,
- udostępnia współdzieloną pamięć oraz ewentualne semaforey,
- dzieli zadanie obliczeniowe na niezależne fragmenty (podzadania),
- przygotowuje i kontroluje struktury potrzebne do prowadzenia obliczeń i zbierające dane o ich postępie.

Manager uruchamiamy jest z następującymi argumentami:

1. nazwą pliku słownika haseł,
2. skrótem hasła do przetwarzania,
3. liczbą podzadań, na które zostanie podzielone całe zadanie obliczeniowe.

Po uruchomieniu Manager wyświetla na ekranie klucz (IPC V) lub nazwę (POSIX) nowoutworzonej kolejki komunikatów i umieszcza w niej komunikaty opisujące kolejne podzadania (należy przewidzieć i obsłużyć przypadek, w którym w kolejce chwilowo zabraknie miejsca na komunikaty). Komunikaty powinny zawierać **komplet** informacji potrzebnych do wykonania obliczeń związanych z konkretnym podzadaniem (odpowiednie klucze lub nazwy obiektów IPC, offsety pamięci współdzielonej, liczbę elementów do przetworzenia, itp.).

Manager kontroluje stopień zaawansowania obliczeń (śledzi, ile podzadań zostało zakończonych, ile jest pobranych i wykonywanych, ile czeka na pobranie).

Po zakończeniu obliczeń (co nastąpi, gdy zakończono wszystkie podzadania albo w trakcie wykonania jednego z nich dopasowano hasło) Manager:

- wyświetla wynik,
- zapobiega dalszym obliczeniom w procesach obliczeniowych, które wykonują kod drugiego programu,
- usuwa niepotrzebne już obiekty potrzebne do komunikacji (Uaga! Należy zadbać o usunięcie obiektów IPC także wtedy, gdy program jest przerywany przed zakończeniem obliczeń (obsłużyć przynajmniej SIGINT).

2. Drugi z programów (**Worker**) może zostać uruchomiony wielokrotnie i pracować równolegle w wielu procesach. Worker wykonuje obliczenia związane z przydzielonymi przez *Managera* podzadaniami. W tym celu Worker uruchamiany jest z następującymi opcjami:

1. kluczem (ewentualnie nazwą) kolejki komunikatów,
2. liczbą podzadań, jakie Worker deklaruje się wykonać.

Worker wykonuje następujące czynności:

- kolejno pobiera dla siebie podzadania z kolejki komunikatów i pobiera ich tyle, ile podano drugim argumentem,
- wykonuje obliczenia i aktualizuje dane we wspólnej pamięci,
- zakładamy, że do momentu odczytania komunikatu Worker nie zna łamanego skrótu, przydzielonych haseł ani żadnych danych potrzebnych do wykonania obliczeń.

Worker powinien prawidłowo obsłużyć sytuację, gdy zostanie przerwany w trakcie działania. W takim wypadku niewykonane podzadania powinny wrócić do kolejki.

Uwaga! Worker pracuje jednowątkowo i nie powołuje do życia żadnych nowych procesów/wątków!

Na antyplagiat wysyłamy kody dwóch programów, zmieniając przed wysłaniem ich nazwy na:
numer_indeksu.ps.lab08.manager.c (czyli np. 66666.ps.lab08.manager.c),
numer_indeksu.ps.lab08.worker.c (czyli np. 66666.ps.lab08.worker.c).

Kody źródłowe (2 pliki) po oddaniu prowadzącemu zajęcia laboratoryjne muszą zostać jako załączniki przesłane na adres pss1@zut.edu.pl (wysyłamy jeden mail z trzema załącznikami):

- pliki z kodem źródłowym muszą mieć nazwy zgodne ze wzorcem podanym w treści zadania,
- mail musi zostać wysłany z poczty uczelnianej (domena **zut.edu.pl**),
- temat maila musi mieć postać: **PS IS1 999X LAB08**, gdzie 999X to numer grupy laboratoryjnej (np. PS IS1 321 LAB08),
- w pierwszych trzech liniach kodu źródłowego w komentarzach (każda linia komentowana osobno) musi znaleźć się:
 - informacja identyczna z zamieszczoną w temacie maila,
 - imię i nazwisko osoby wysyłającej maila,
 - adres e-mail, z którego wysłano wiadomość,np.:

```
// PS IS1 321 LAB08
// Jan Nowak
// nj66666@zut.edu.pl
```

- e-mail nie może zawierać żadnej treści (tylko załączniki).

Dostarczone kody programów będą analizowane pod kątem wykrywania plagiatów. Niewysłanie wiadomości, wysłanie jej w formie niezgodnej z powyższymi wymaganiami lub wysłanie pliku, który nie będzie się kompilował i uruchamiał, będzie traktowane jako brak programu i skutkowało otrzymaniem za niego oceny niedostatecznej.