Naszym zadaniem będzie napisanie programu, który pozwoli na łamanie wybranego hasła przechowywanego w **/etc/shadow** metodą słownikową. Zakładamy, że będziemy w tym celu mieli dostęp do tego pliku lub jego wypreparowany fragment.

1. W pierwszym kroku należy zapoznać się z problemem generowania skrótów haseł systemowych za pomocą funkcji **crypt** i **crypt_r (3)**. Zwrócić uwagę na to, czy w naszym systemie obydwie funkcje wspierają kryptograficzną funkcję skrótu **SHA-512** (wersja biblioteki standardowej od **glibc 2.7**).

   W ramach przygotowania do pracy napisz program generujący skróty haseł do formy przechowywanej w **/etc/shadow** (kryptograficzna funkcja skrótu **SHA-512** z domieszką). Program powinien pobierać jako argumenty hasło i domieszkę (salt), a zwracać skrót sformatowany do postaci przechowywanej w **/etc/shadow**. Przykładowo dla hasła **dees** i domieszki **5MfvmFOaDU** powinniśmy otrzymać skrót:

   CVt7jU9wJRYz3K98EklAJqp8RMG5NvReUSVK7ctVvc2VOnYVrvyTfXaIgHn2xQS78foEJZBq2oCIqwfdNp.2V1

   który trafiłby do **/etc/shadow** w formie:

   $6$5MfvmFOaDU$CVt7jU9wJRYz3K98EklAJqp8RMG5NvReUSVK7ctVvc2VOnYVrvyTfXaIgHn2xQS78foEJZBq2oCIqwfdNp.2V1

2. Drugi program ma służyć do łamania hasła metodą słownikową. Jako argumenty wywołania podajemy:
   - sformatowany skrót hasła, które chcemy złamać (np. wypreparowany z **/etc/shadow** bądź otrzymany w wyniku działania pierwszego programu),
   - plik tekstowy zawierający słownik z hasłami (do testów udostępniono plik **lab07.medium.txt** z rzeczywistymi hasłami pochodzącymi głównie z wycieków),
   - liczbę wątków, które będą współpracowały przy obliczeniach.

   Program kończy swoje działanie albo po dopasowaniu hasła (wyświetlając je na ekranie) albo, w przypadku braku dopasowania, po przejściu całego słownika (odpowiednia informacja również powinna pojawić się na ekranie).

   Program powinien sprawdzić aktualnie dostępną w systemie ilość procesorów (funkcja **sysconf (3)**) i nie tworzyć większej od niej ilości wątków.

   Program powinien w procentach pokazywać zaawansowanie obliczeń (ilość sprawdzonych haseł ze słownika), uwzględniając wyniki wszystkich współpracujących wątków.

Zwrócić uwagę na problem sekcji krytycznej w przypadku wykorzystywania przez wątki wspólnych zmiennych i zastanowić się nad optymalną częstotliwością ich aktualizacji.

Nie wykonuj odczytu haseł z pliku pojedynczymi bajtami lub liniami (zbyt czasochłonne). Rozważyć mapowanie pliku do pamięci (**mmap**) lub odczyty wielokrotnościami bloków dyskowych.

W przypadku braku argumentu określającego ilość wątków program nie ma docelowo łamać hasła a jedynie ustalić optymalną ilość wątków, które mogą zostać użyte. W tym celu program będzie wykonywał obliczenia dla pewnej ilości haseł ze słownika (np. 1000) i dla różnej ilości wątków, mierząc przy tym dokładny czas obliczeń. Otrzymane wyniki powinny zostać wyświetlone na ekranie.

Na antyplagiat wysyłamy jedynie drugi program, zmieniając przed wysłaniem jego nazwę na: ***numer_indeksu.ps.lab07.main.c*** *(czyli np. 66666.ps.lab07.main.c).*

Kod źródłowy (1 plik) po oddaniu prowadzącemu zajęcia laboratoryjne muszą zostać jako załączniki przesłane na adres **pss1@zut.edu.pl** (wysyłamy jeden mail z trzema załącznikami):

- plik z kodem źródłowym musi mieć nazwę zgodną 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 LAB07**, gdzie 999X to numer grupy laboratoryjnej (np. PS IS1 321 LAB07),
- 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 LAB07
    // 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.