Łamanie haseł

Jak łatwo i przyjemnie można zrównoleglić kod - choć nie zawsze ma to sens.

Cel laboratorium

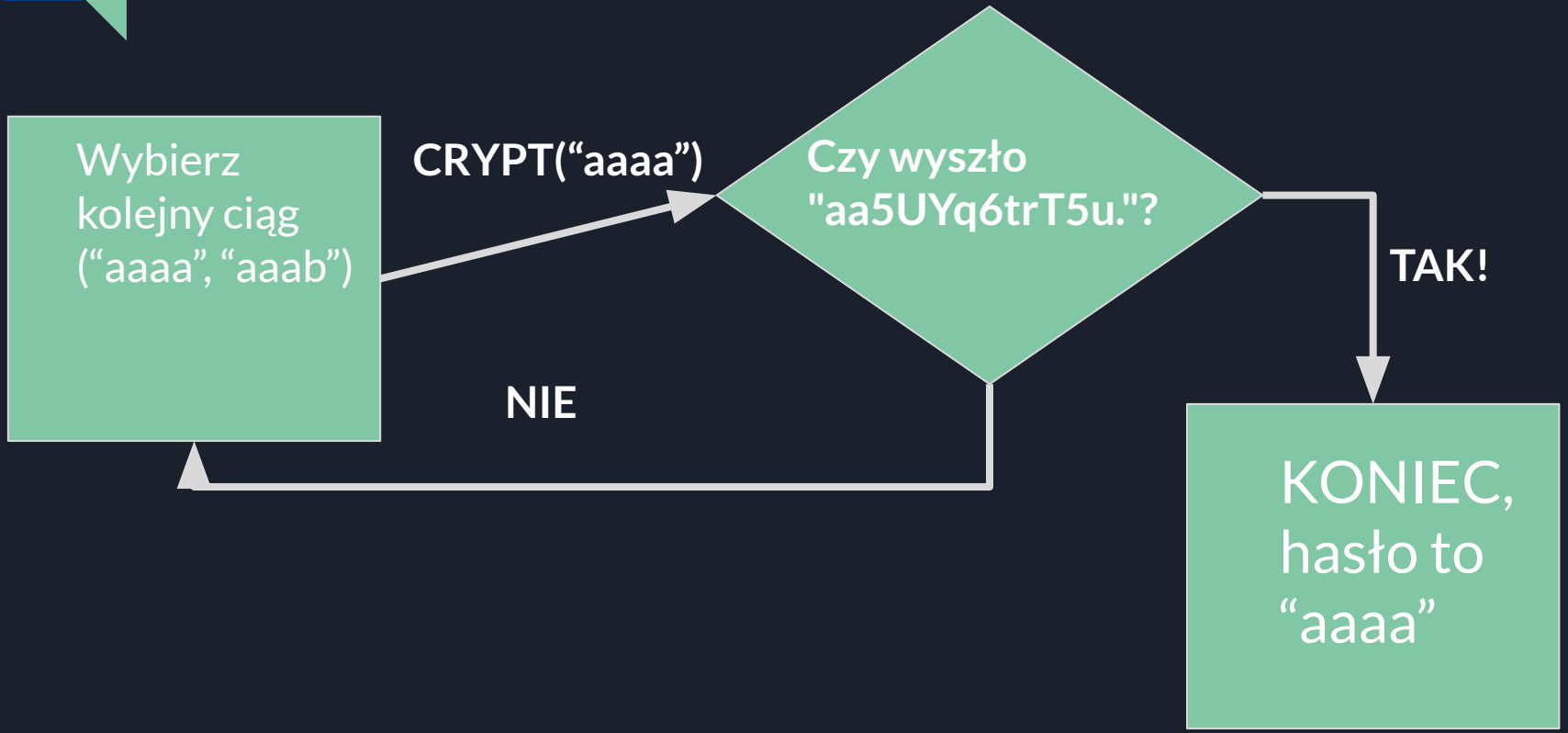
Będziemy łamać hasła metodą brutalnej siły, aby pokazać, jak łatwo jest zrównoleglić kod - oraz, że nie zawsze ma to sens.

UWAGA: zwykle hasła wybiera się tak, by były trudne do złamania (by zawierały znaki specjalne, cyfry itd), oraz wybiera się nowoczesną metodę szyfrowania. My mamy tylko półtorej godziny zajęć, więc wybierzemy hasła proste i prostą metodę..

Ogólna idea

- Hasła: czteroliterowe, same małe litery (np. mana).
- kodujemy funkcją crypt
- Dane hasła zakodowane, np "aa5UYq6trT5u."

Ogólna idea


Wybierz kolejny ciąg (“aaaa”, “aaab”)
CRYPT(“aaaa”)
Czy wyszło "aa5UYq6trT5u."?
TAK!
NIE
KONIEC, hasło to “aaaa”

Zadanie do wykonania

- Wykonać wersję sekwencyjną (**BEZ MPI!**) dla haseł cztero-, pięcio- i sześcioliterowych. Kompilujemy `gcc -lcrypt`, mierzymy czasy przy pomocy `time ./crack`
- Rozproszyć program (MPI!), dla tych samych haseł. Kompilujemy `mpicc -lcrypt`, mierzymy czasy `time mpirun -np <liczba_procesów> ./crack`

UWAGA!

- Łańcuchy porównujemy przy pomocy strcmp
- Drugi argument funkcji crypt to tzw *salt*, nie modyfikować
- W wersji rozproszonej jeden proces może odgadnąć hasło, a inne jeszcze mogą pracować
- Nie zwalniać pamięci dla rezultatu funkcji crypt
- Zmienna cmp zawsze ma rozmiar o jeden większy niż rozmiar hasła