

**Cel:**

- atrybuty wątków PTHREADS.

**Kroki:**

1. Utworzenie katalogu roboczego (np. lab\_3)
2. Uruchomienie na dziesięciu wątkach funkcji wyświetlającej identyfikator wątku.

**Przykładowy wydruk:**

watek glowny: tworzenie watku potomnego nr 1  
    watek glowny: wyslanie sygnalu zabicia watku  
    watek glowny: watek potomny NIE zostal zabity - blad  
watek glowny: tworzenie watku potomnego nr 2  
    watek glowny: odlaczenie watku potomnego  
    watek glowny: wyslanie sygnalu zabicia watku odlaczonego  
watek glowny: tworzenie odlaczonego watku potomnego nr 3  
    watek glowny: koniec pracy, watek odlaczony pracuje dalej  
    watek potomny: uniemozliwione zabicie  
    watek potomny: umozliwienie zabicia  
    watek potomny: zmiana wartosci zmiennej wspolnej

3. Przeanalizuj i uzupełnij załączony [program](#) (watki.c) tak aby przetestować możliwości zarządzania wątkami

watek glowny: tworzenie watku potomnego nr 1  
    watek potomny: uniemozliwione zabicie  
    watek glowny: wyslanie sygnalu zabicia watku  
    watek glowny: watek potomny NIE zostal zabity - blad  
watek glowny: tworzenie watku potomnego nr 2  
    watek glowny: odlaczenie watku potomnego  
    watek glowny: wyslanie sygnalu zabicia watku odlaczonego  
watek glowny: tworzenie odlaczonego watku potomnego nr 3  
    watek glowny: koniec pracy, watek odlaczony pracuje dalej  
    watek potomny: uniemozliwione zabicie  
    watek potomny: umozliwienie zabicia  
    watek potomny: umozliwienie zabicia  
    watek potomny: zmiana wartosci zmiennej wspolnej

=====

watek glowny: tworzenie watku potomnego nr 1  
    watek potomny: uniemozliwione zabicie  
    watek glowny: wyslanie sygnalu zabicia watku  
    watek potomny: umozliwienie zabicia  
    watek glowny: watek potomny zostal zabity  
watek glowny: tworzenie watku potomnego nr 2  
    watek glowny: odlaczenie watku potomnego  
    watek glowny: wyslanie sygnalu zabicia watku odlaczonego  
watek glowny: tworzenie odlaczonego watku potomnego nr 3  
    watek glowny: koniec pracy, watek odlaczony pracuje dalej  
    watek potomny: uniemozliwione zabicie  
    watek potomny: umozliwienie zabicia  
    watek potomny: zmiana wartosci zmiennej wspolnej

4. Na podstawie poprzedniego punktu stworzenie programu uruchamiającego funkcję `czasozajmowacz()` lub podobną na kilku wątkach i przetestowanie sterowania wielkością stosu (**`pthread_attr_setstacksize`**) oraz przydziałem procesora (**`pthread_setaffinity_np`**) i zapoznanie się z funkcjami sterowania priorytetem (**`pthread_setschedparam`**).
5. Podglądnięcie przydziału procesora przy pomocy programu `htop`.
6. Na podstawie przetestowanego programu stworzenie nowego kodu, w którym uruchomione zostanie 10 wątków z losowymi wartościami atrybutów, a następnie parametry te zostaną
  - a). wypisane w procedurze wątku (procedura „`pthread_getattr_np`” lub podobne)
  - b). zdiagnozowane przez wątek główny na podstawie zachowania utworzonych wątków (opis zachowania wątków dla różnych wartości atrybutów można znaleźć na stronach podręcznika, np. „`man pthread_create`” i inne)