

# Systemy Operacyjne

## Laboratorium 7

### *Problem uczujących filozofów*

prowadzący: mgr inż. Piotr Nowak

---

## Zadanie

Celem zadania jest implementacja rozwiązania problemu uczujących filozofów. Zadanie należy zrealizować w domu, po czym zademonstrować na zajęciach.

Do uzyskania maksymalnej oceny należy poprawnie:

1. Rozwiązać problem synchronizacji wątków.
2. Rozwiązać problem zagłodzenia.
3. Rozwiązać problem zakleszczenia.
4. Wykorzystać zmienną warunkową.

## Wymagania

- Aplikacja może być napisana w **C** (biblioteka **pthread**s) lub **C++** (biblioteka **thread**).
- Aplikacja musi być skompilowana i uruchomiona w środowisku Linuxowym.
- Aplikacja ma za zadanie jedynie symulację - zero interakcji.
- Liczba filozofów ma być podawana jako argument linii komend podczas uruchamiania aplikacji. Liczba filozofów ma być nie mniejsza niż 5.
- Wątek główny (z funkcją **main()**) może zakończyć się dopiero po zakończeniu wszystkich innych wątków.

## Zasady wizualizacji

- Wizualizacja w postaci graficznej / zwykłego trybu konsolowego (linia po linii) jest niedozwolona. Należy wykorzystać bibliotekę **ncurses**.
- Wizualizacja ma obejmować stan filozofów oraz stan zasobów.
- Wizualizacja stanu wątków powinna być płynna - stan wątku na ekranie powinien być aktualizowany co najmniej 2-3 razy na sekundę. Jeśli wątek wykonuje jakąś czynność (np. filozofowanie) przez kilka sekund, to można dodać licznik pokazujący postęp czynności lub odliczający czas do jej końca lub pasek postępu.
  - Jedynym wyjątkiem od zasady jest sytuacja, gdy wątek jest zablokowany np. na funkcji **pthread\_mutex\_lock** (choć i z tej sytuacji można wybrnąć, jeśli wizualizacja jest wykonywana przez osobny, dedykowany wątek).
- Wizualizacja powinna być czytelna (pozwalająca na łatwe śledzenie, co się dzieje z wątkami/zasobami). Nie należy jej zbytnio komplikować.
- Zaleca się używanie jedynie prostych funkcji z biblioteki **ncurses** (oraz ewentualnie kół).