## Program 1A

**Program po wywołaniu bez argumentów kończy się, wyświetlając jedynie komunikat, że został wywołany bez argumentów.**

W przypadku wywołania z liczbą argumentów większą niż 1, tworzy taką liczbę procesów potomnych, ile było argumentów, a każdy z procesów wyświetla na ekranie jeden argument.

***Gdy pierwszym argumentem jest "c", wykorzystana zostanie funkcja *clone(…)*, jeżeli jest to inny znak, używany jest *fork()*.**

## Program 1B

**Program po uruchomieniu nie kończy się, ale nie robi nic ;) Pozwala jednak na uruchomienie tylko jednej swojej kopii. W przypadku próby uruchomienia kolejnej wersji, nowo-uruchamiana instancja powinna wyświetlić komunikat, że proces jest już uruchomiony i się zakończyć.**

Jeżeli w zmiennych systemowych znajduję się zmienna **SO2=NEW**, "stara" instacja (pracująca wcześniej) powinna zostać zakończona, a pracuje tylko ta ostatnio uruchomiona.

Rozwiązanie możliwe jest na kilka sposobów, można wykorzystać tzw. blokadę plikową, semafor nazwany, gniazdo sieciowe itd.

---

## Program 3A

**Zachowanie programu jest takie jak w wersji *IX/POSIX, z pominięciem opcji wywołanie "c", ale został napisany z wykorzystaniem wyłącznie WinAPI (*windows.h*).**

Wskazówka: *CreateProcess(…)*

## Program 3B

**Zachowanie programu jest takie jak w wersji *IX/POSIX, z pominięciem opcji wywołanie "c", ale został napisany z wykorzystaniem wyłącznie WinAPI (*windows.h*).**

*Wskazówka: *CreateMutex (…)* lub *EnumProcesses(…)* lub *CreateEvent(…)*

---

### Po wykonaniu zadań:

- [ ] Umiem wywołać nowy proces, zarówno będacy własną kopią jak i zewnętrzny program
- [ ] Rozumiem listę argumentów przekazywanych podczas uruchamiania procesu oraz umiem odczytać zmienne środowiskowe, a za pomocą argumentów jak i zmiennych środowiskowych sterować przebiegiem programu
- [ ] Znam podstawowe pojęcia pozwalające na poruszanie się po dokumentacji API używanego systemu