

## Zadanie B Profiler.

Należy rozbudować system MINIX o możliwość profilowania procesów. System ma obserwować wskazany proces zapamiętując jak dużo czasu spędził w poszczególnych częściach swojego kodu.

W każdej chwili obserwowany może być co najwyżej jeden proces. Segment kodu profilowanego procesu dzielimy na 100 równych części. Z każdą częścią zwiążemy licznik. Na początku obserwowania wszystkie liczniki są ustawione na 0. Po każdym przerwaniu zegarowym, jeśli bieżący proces jest obserwowany to zwiększamy o 1 licznik dotyczący części w której znajduje się ostatnio wykonywana instrukcja.

W implementacji można zaniedbać bezpieczeństwo i przyjąć że każdy proces może obserwować każdy inny (poza tym że w systemie może być obserwowany co najwyżej jeden w każdej chwili).

Interface dla procesu użytkownika:

- `int pstart(pid_t pid);` – rozpoczyna obserwowanie procesu o numerze `pid`. Zwraca 1 jeśli już jakiś proces jest obserwowany lub nastąpił inny błąd (np. IPC), wpp 0.
- `int pstop(unsigned int *buff);` – kończy obserwowanie bieżącego procesu. Zapisuje pod adresem `buff` tablicę liczników. Zwraca 1 w przypadku błędu (np. żaden proces nie jest obserwowany), zwraca 0 wpp.

Powyższe funkcjonalności mogą być udostępniane procesom użytkownika przez Process Manager.

### Instrukcja sumbitowania:

Rozwiązanie będzie testowane w systemie MINIX 3.1.0 (book version). Należy wysłać jedno archiwum `tar` zawierające wszystkie pliki źródłowe, które były zmieniane lub dodane. Archiwum będzie rozpakowane w katalogu `\usr\src` instrukcją `tar xf archive.tar`. Po rekompilacji obrazu systemu, rekompilacji biblioteki systemowej i restarcie profilowanie powinno działać.