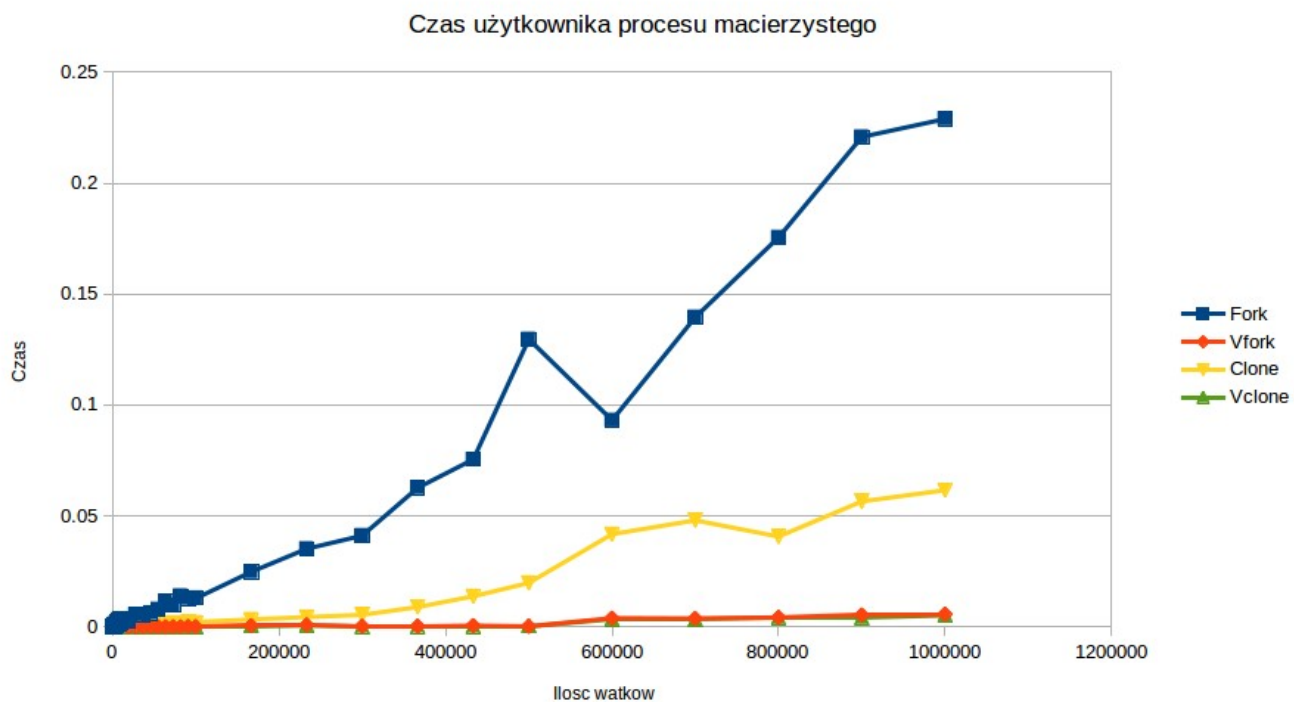
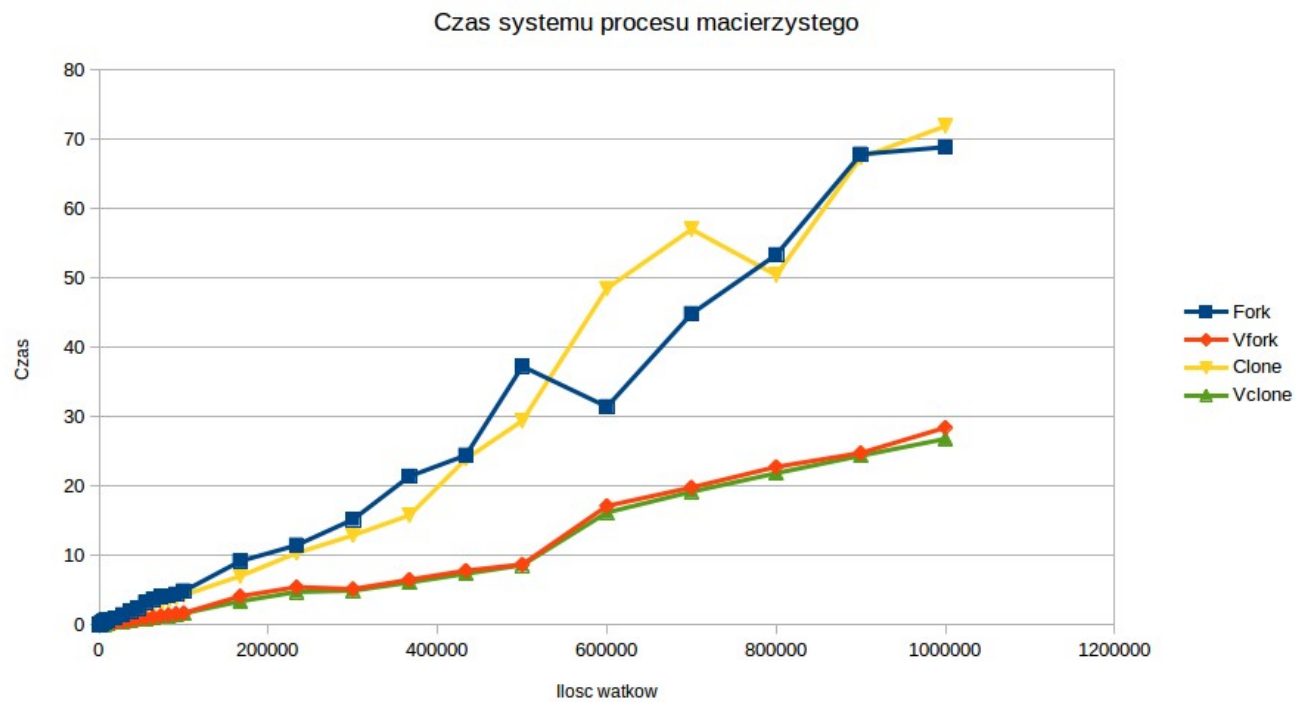
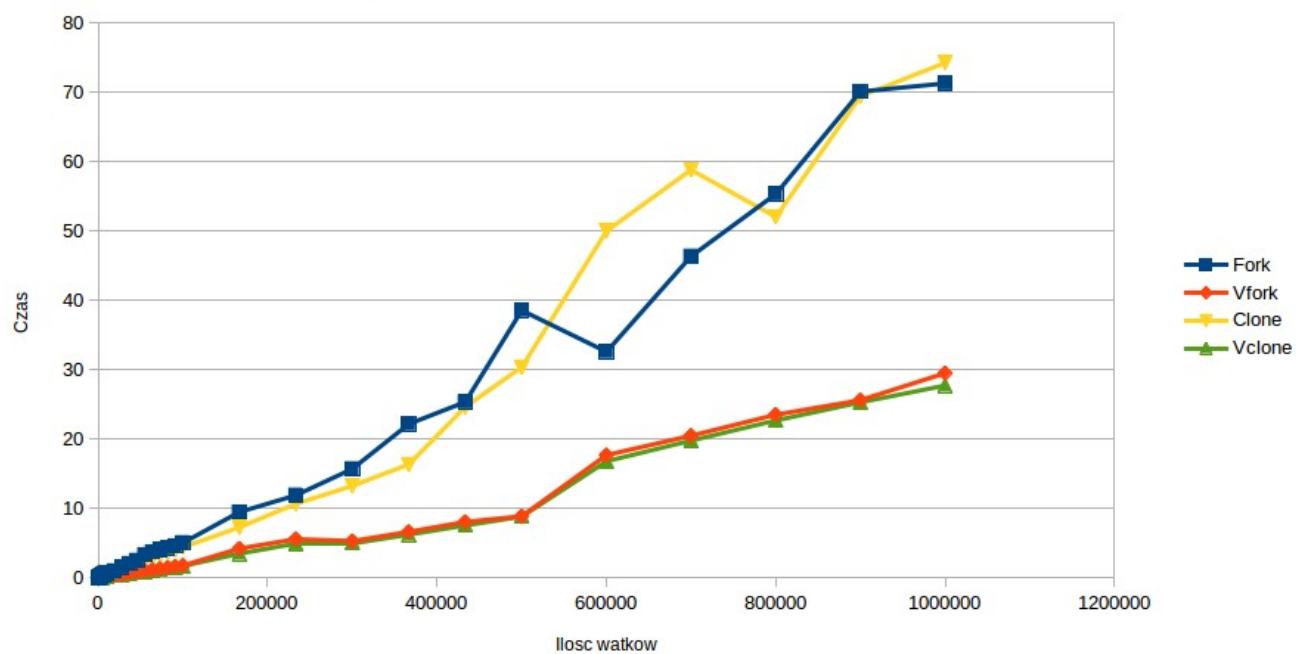


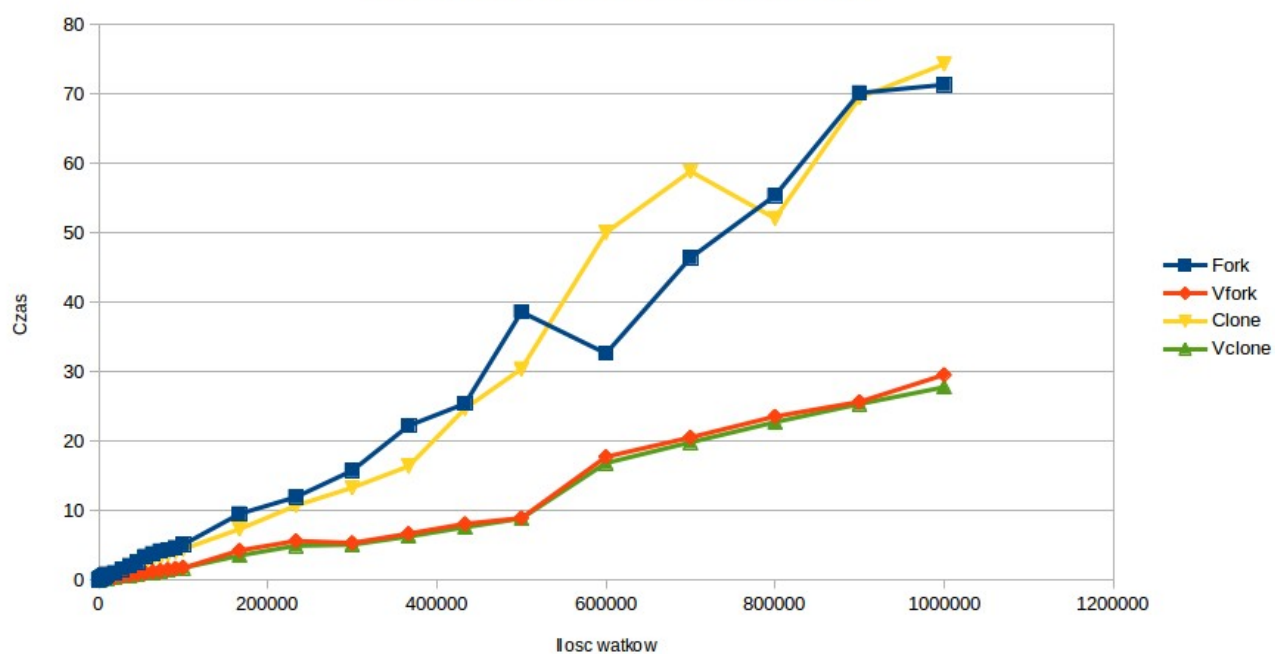
Zbigniew Królikowski, wnioski z ćwiczenia



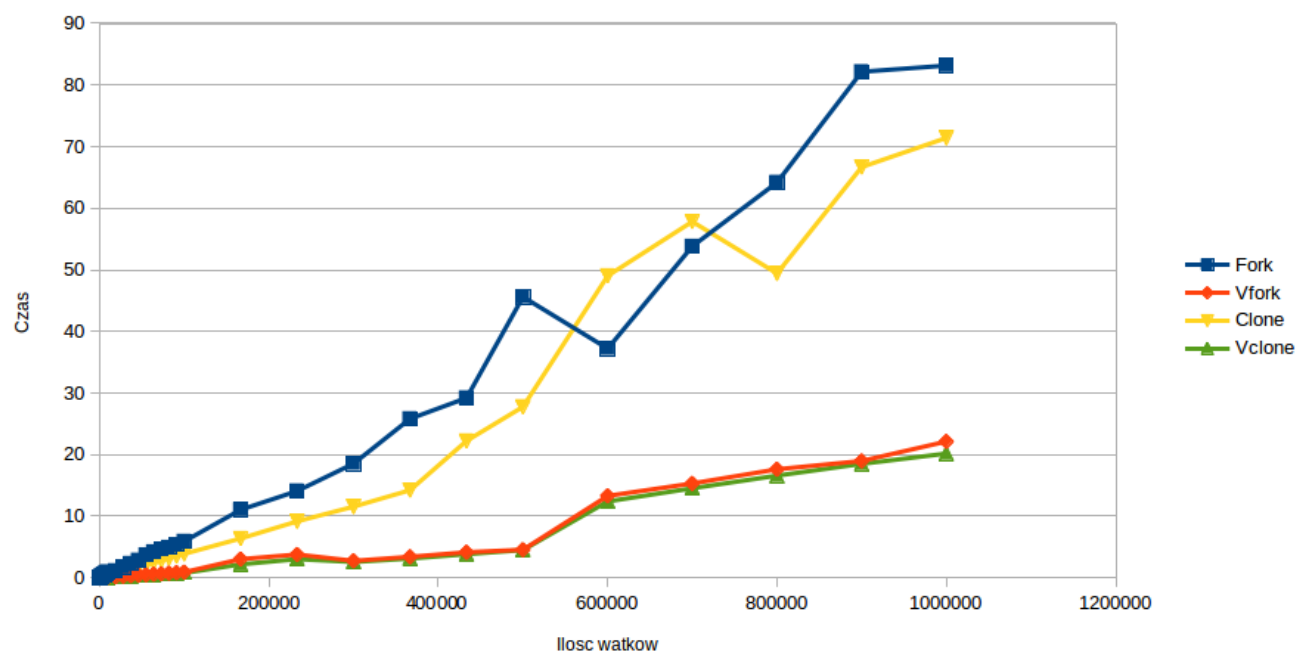
Czas systemu + czas użytkownika dla systemu macierzystego



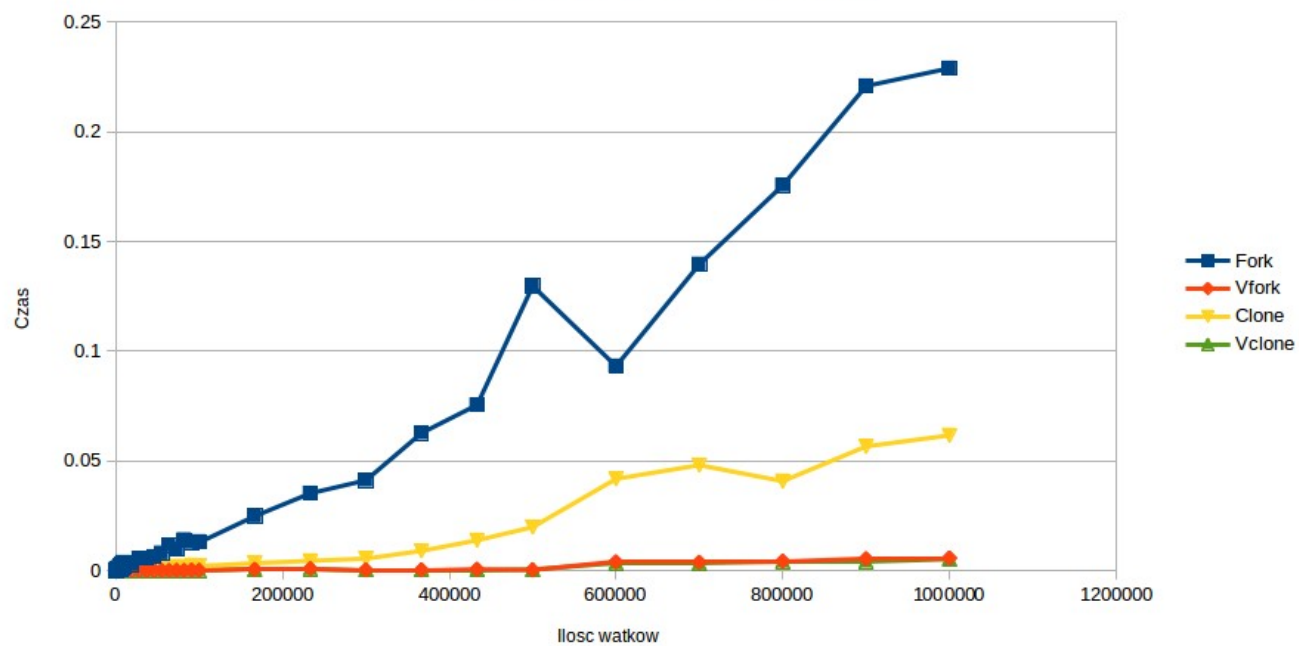
Czas rzeczywisty dla procesu macierzystego



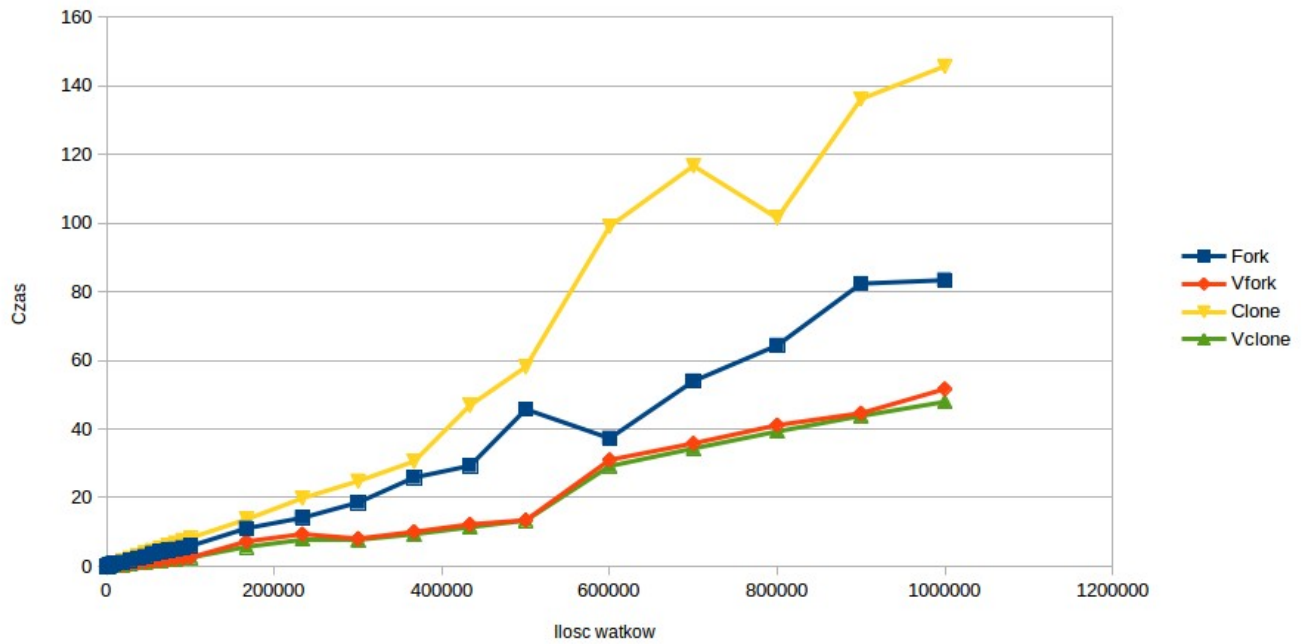
Czas systemu procesów potomnych



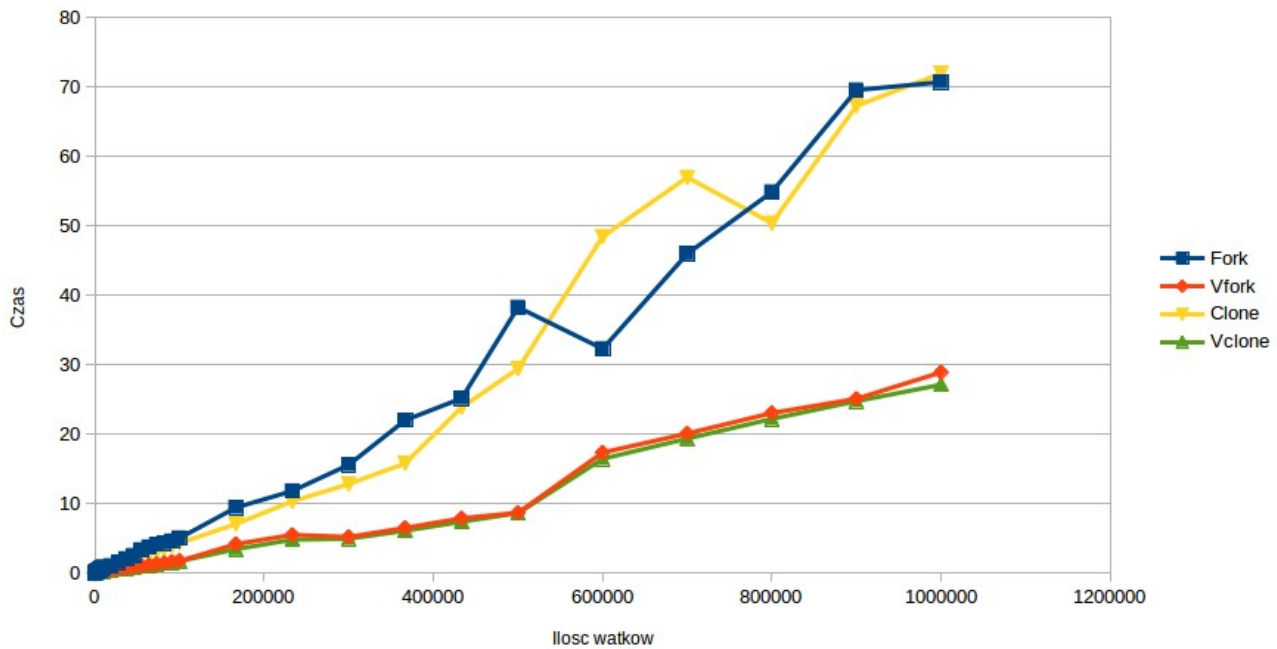
Czas użytkownika procesów potomnych



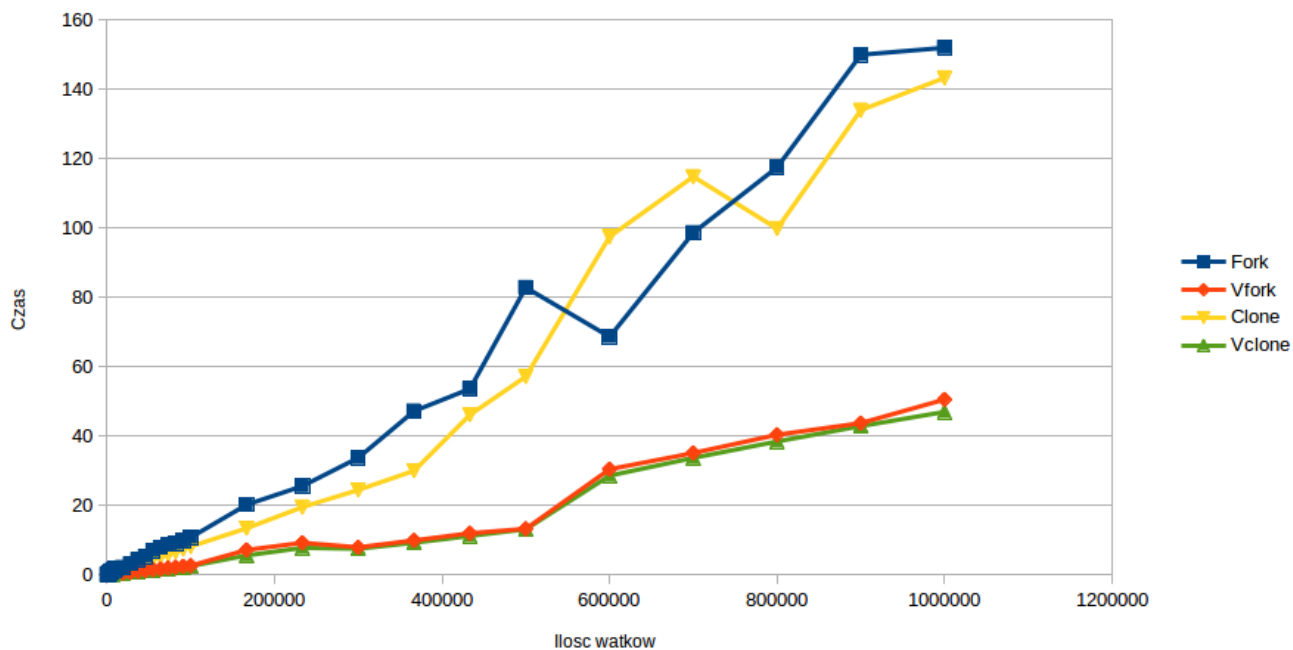
Czas systemu + użytkownika dla procesów potomnych



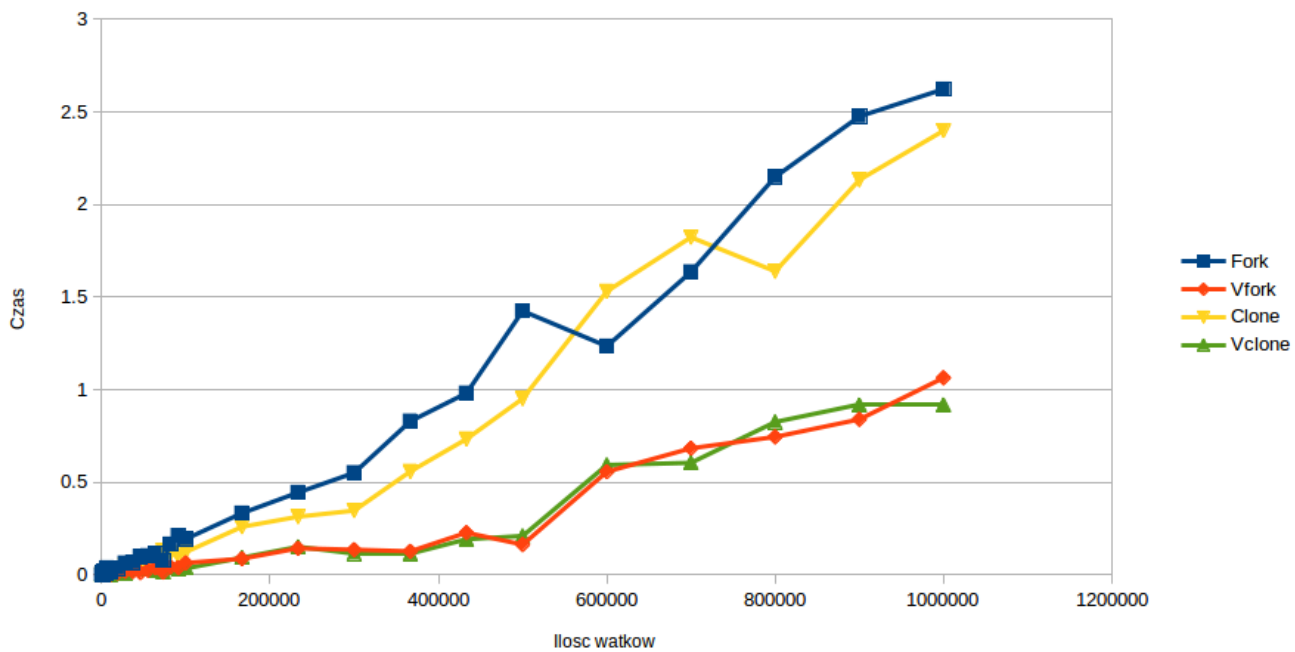
Czas rzeczywisty dla procesów potomnych



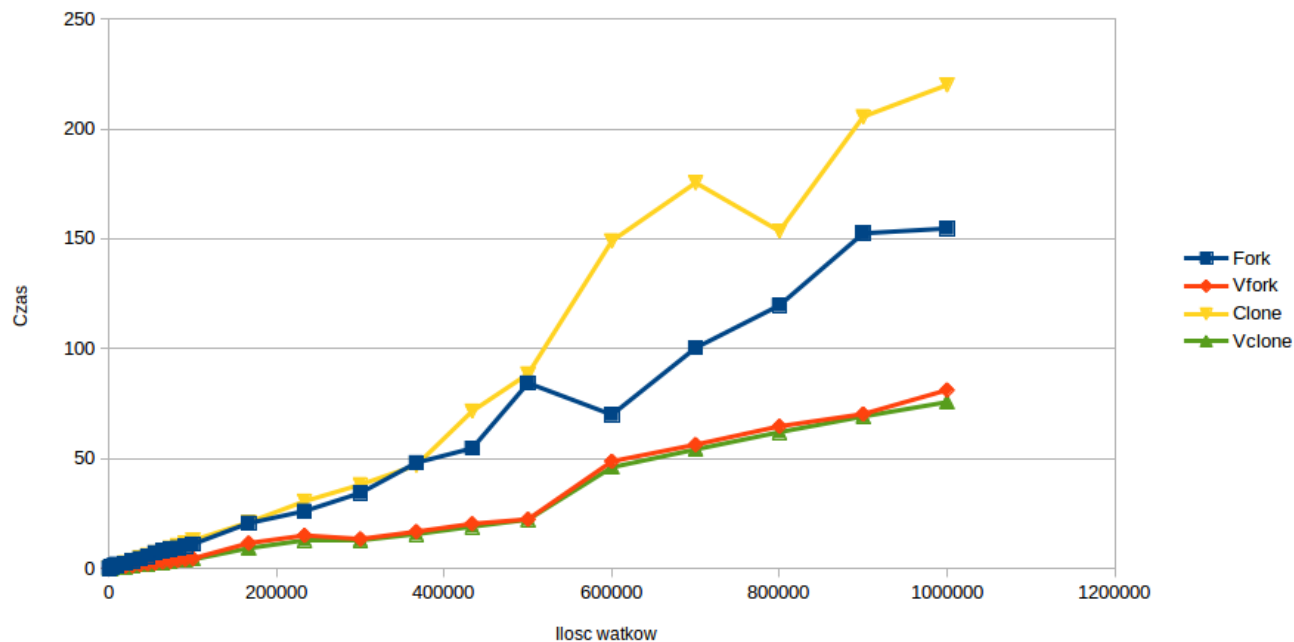
Czas systemu procesu macierzystego + procesów potomnych



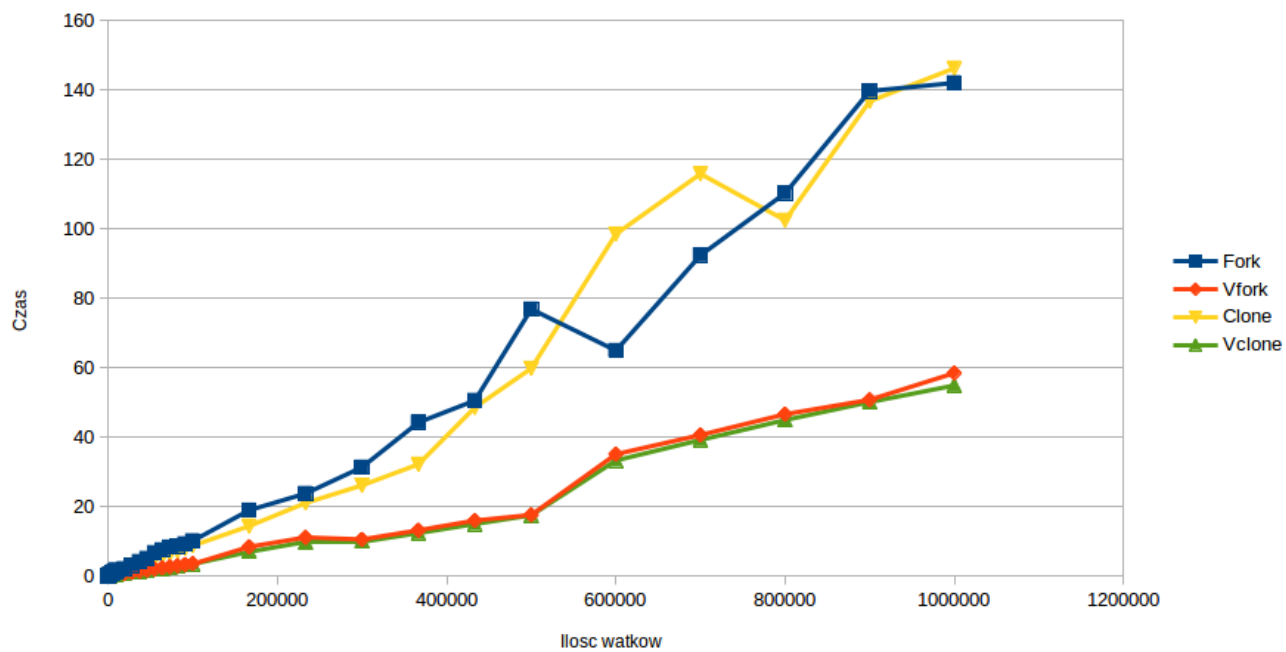
Czas użytkownika procesu macierzystego + procesów potomnych



Czas systemu + użytkownika dla procesu macierzystego oraz procesów potomnych



Czas rzeczywisty dla procesu macierzystego oraz procesów potomnych



Wnioski:

1. Zgodnie z oczekiwaniami czasy dla `fork` oraz `clone` są dłuższe od czasów dla `vfork` oraz `vclone`. Wynika to z fundamentalnej różnicy pomiędzy tymi dwoma funkcjami. Te pierwsze kopiują wszystkie zasoby procesu takie jak pamięć, przestrzeń adresowa, otwarte pliki co jest wykonywane w czasie systemu.
2. Czasy procesu macierzystego są większe na funkcji Unix'owych niż w przypadku funkcji zaimplementowanych w systemie Linux. Wynika to z ich zależności programowej. Funkcje `fork` i `vfork` wykorzystują funkcję `clone`. Da się o tym przekonać wykorzystując funkcję `strace`.
3. Funkcje wykorzystują procesor, głównie w trybie systemu, co oznacza, że czynności wykonywane w ich ramie wymagają tego uprawnienia.