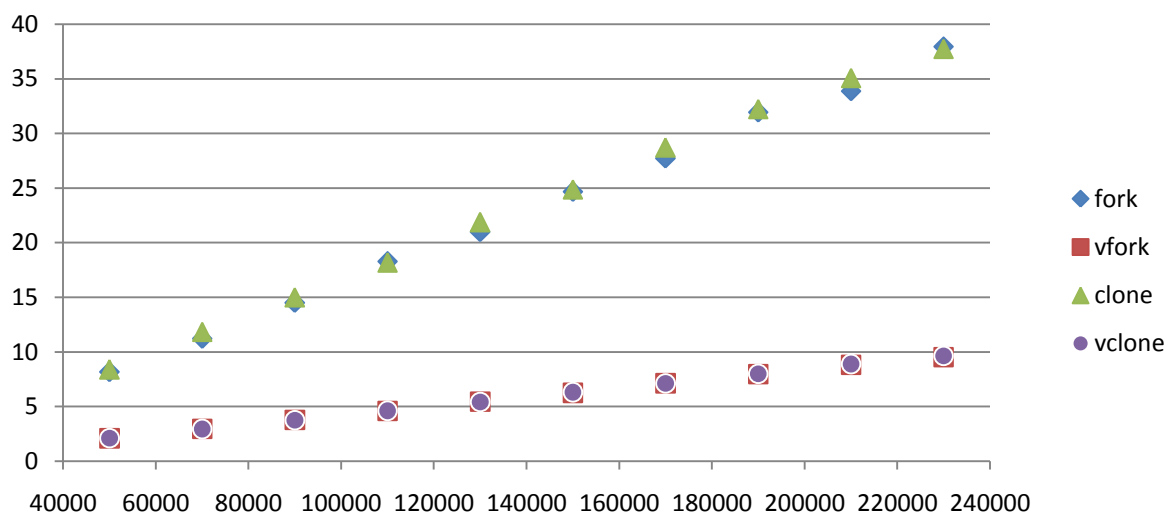
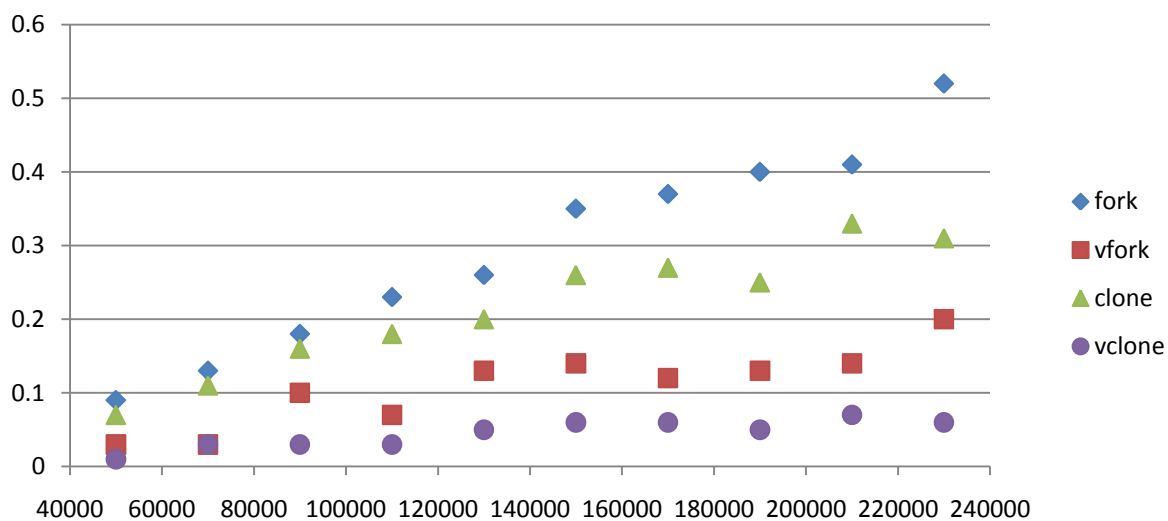


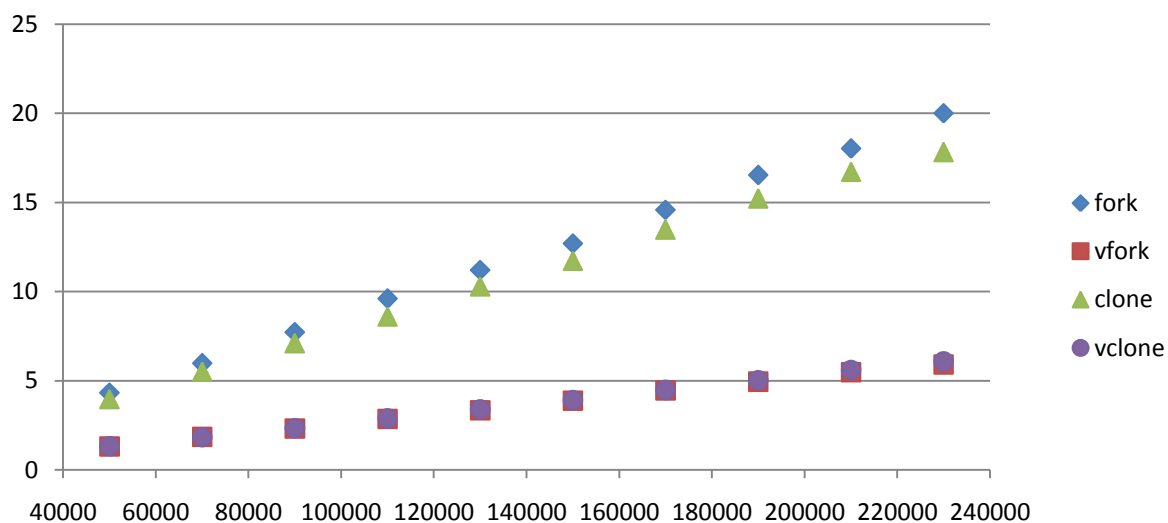
proces macierzysty - czas rzeczywisty



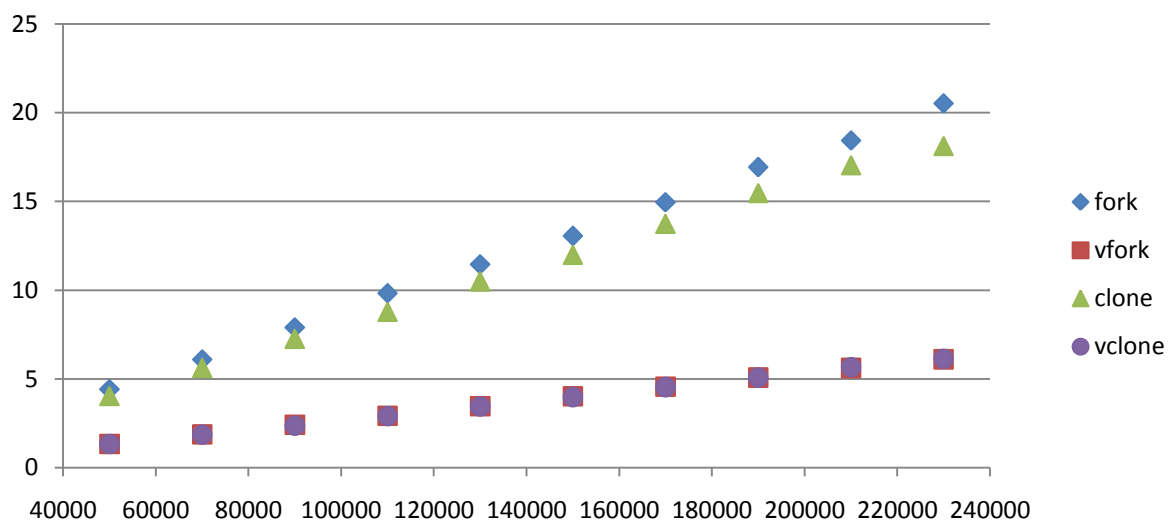
proces macierzysty - czas uzytkownika



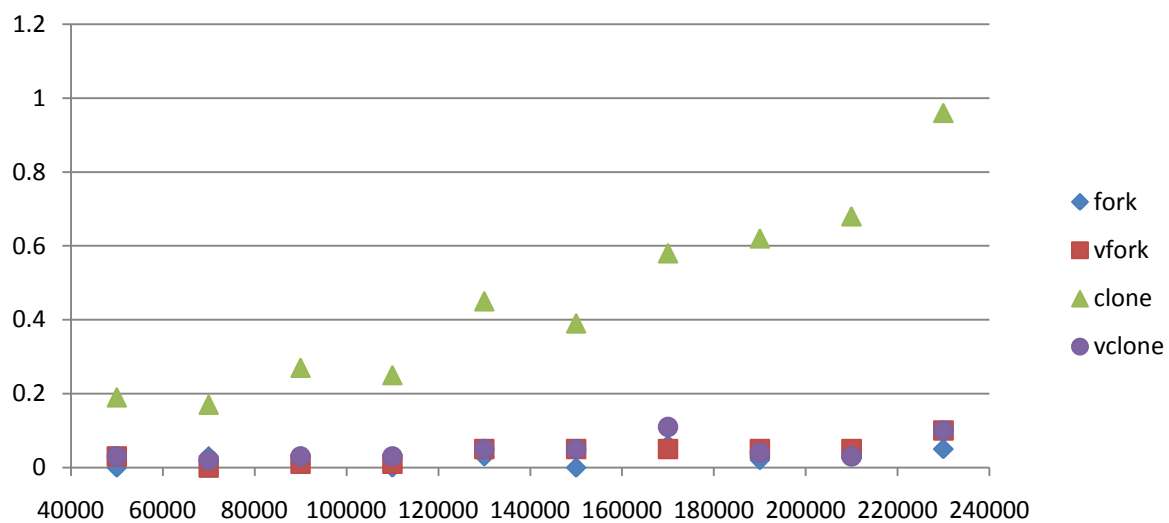
proces macierzysty - czas systemowy



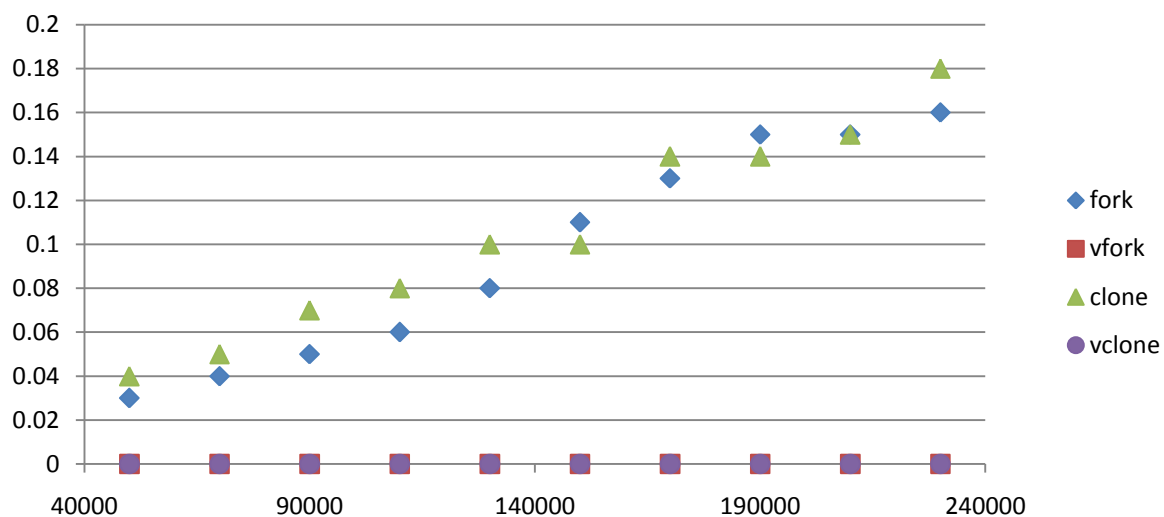
proces macierzysty - user+system



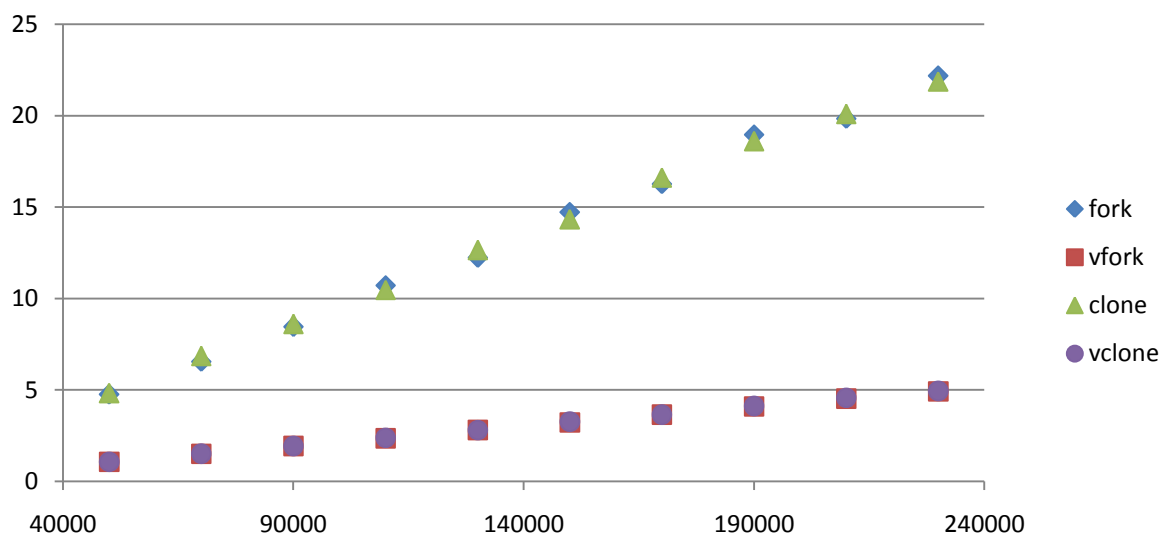
proces potomny - czas rzeczywisty



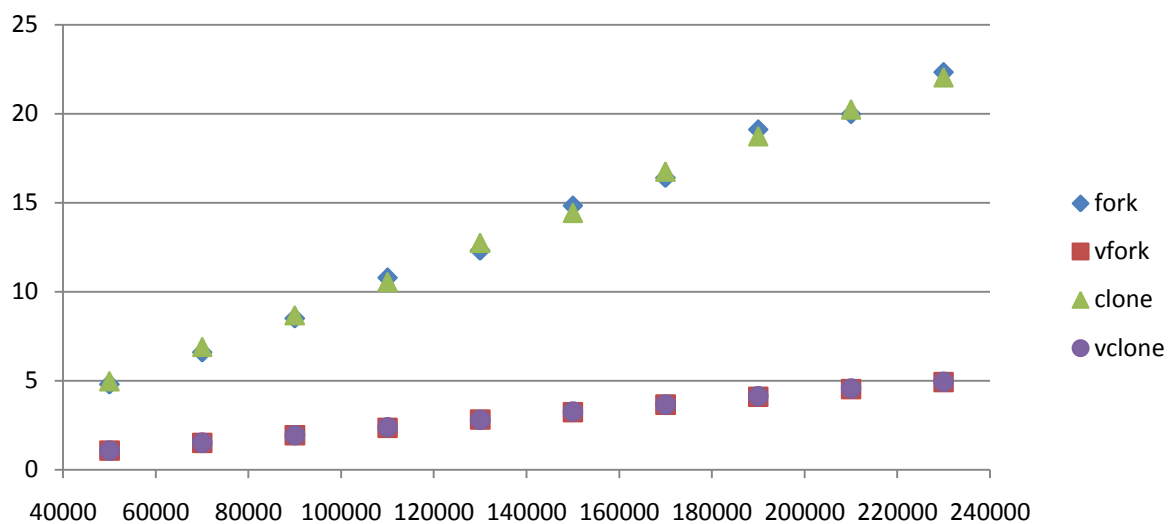
proces potomny - czas uzytkownika



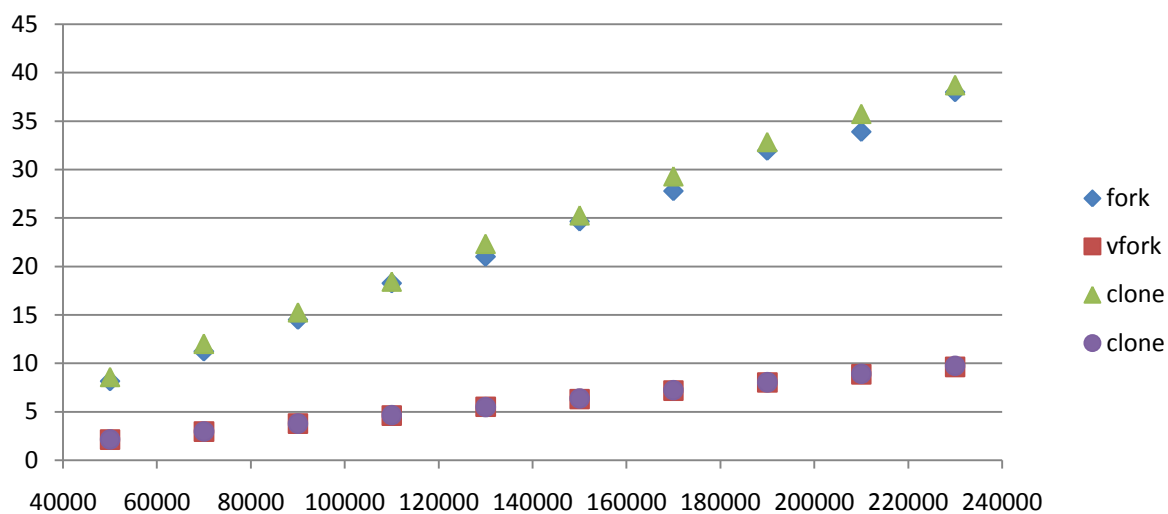
proces potomny - czas systemowy



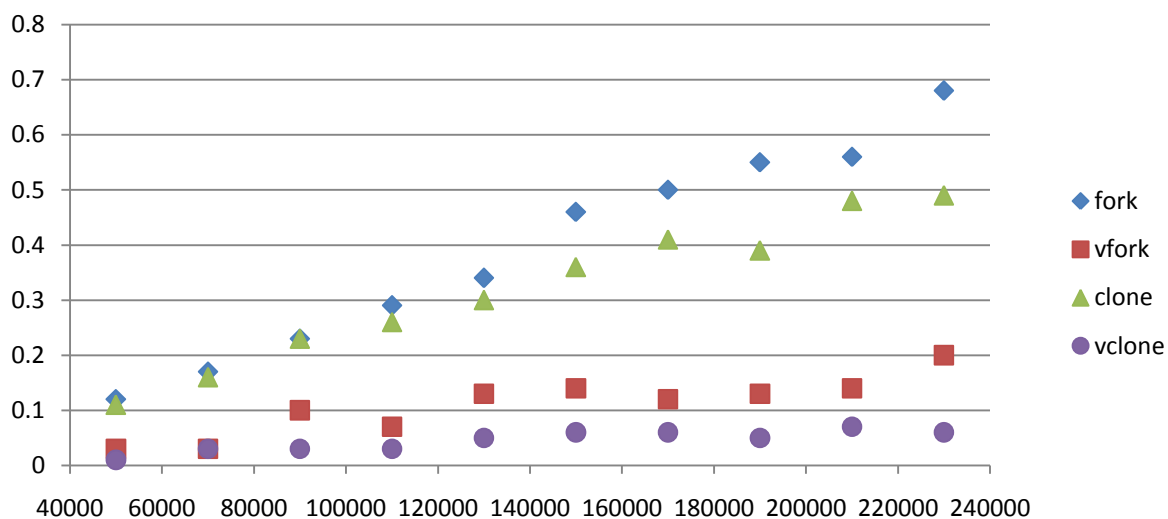
proces potomny - user+system



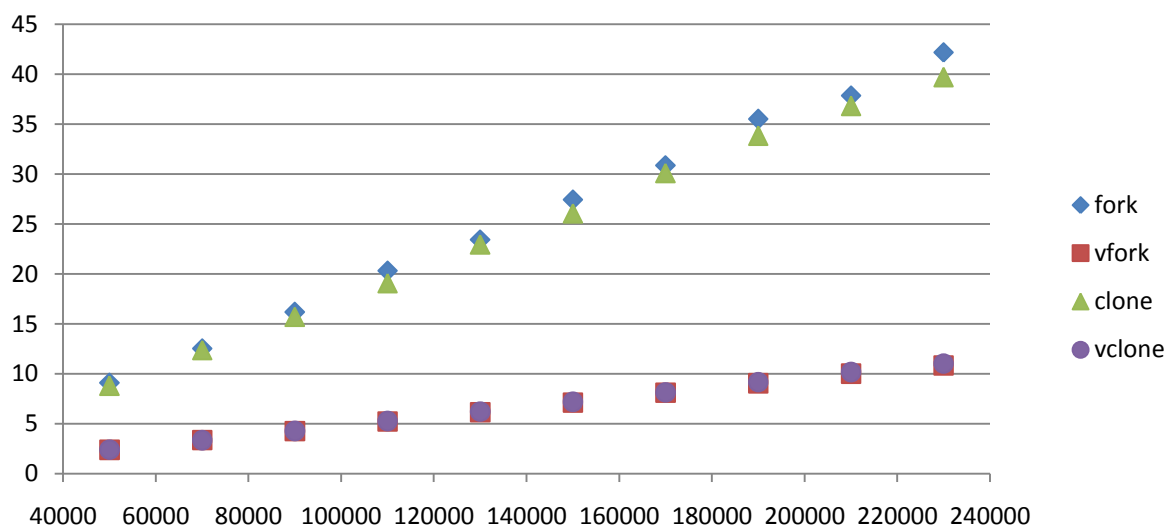
macierzysty+potomny - czas systemowy



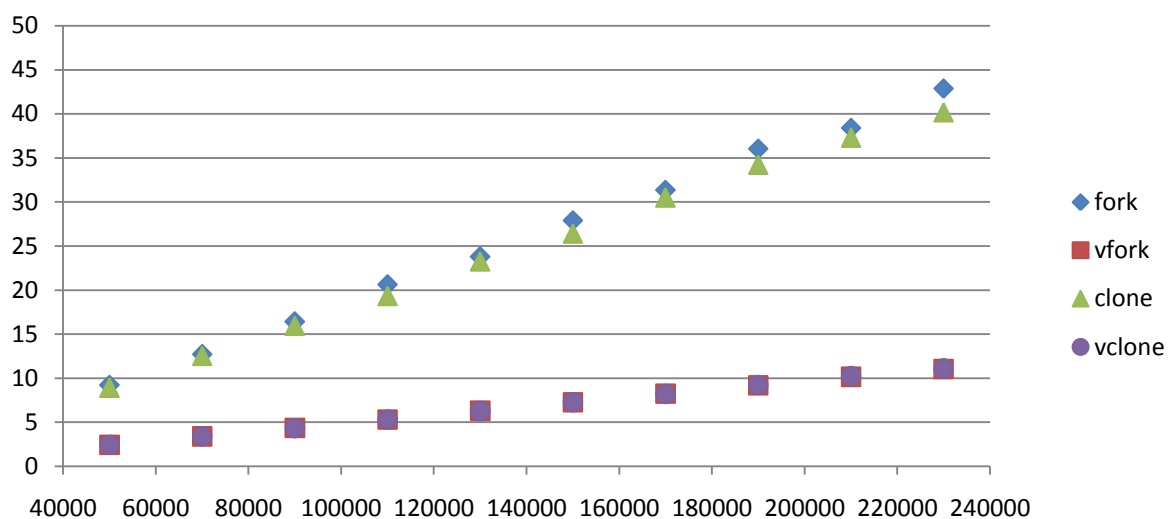
macierzysty+potomny - czas uzytkownika



macierzysty+potomny - czas systemowy



macierzysty+potomny - user+system



Wnioski:

1. `vfork()` jest znacznie szybszy od `fork()` ze względu na użycie Copy-On-Write przy `fork()`
2. Nie ma istotnej różnicy pomiędzy `clone()` a `fork()` z odpowiednią flagą, i pomiędzy `clone()` i `vfork()` z odpowiednimi flagami
3. Czasy rosną liniowo wraz ze zwiększaniem ilości procesów
4. `vfork()` jest prawie 4 razy szybszy od `fork()`