

Хар-ка/прогр.	С (проц.подход)	С++(ООП)	Python(дин.тип.)	С + NASM
<b>Время выполнения,сек</b>				
5 языков (ввод из файла)	0.00401	0.008473	0.0002075	0.000285
12 языков (ввод из файла)	0.00639	0.003756	0.0005823	0.000421
100 языков (ввод из файла)	0.01532	0.0040382	0.002376	0.000678
100 языков (программная генерация)	0.0266	0.003282	0.002521	0.000460
500 языков (программная генерация)	0.03612	0.005681	0.009842	0.001073
5000 языков (программная генерация)	0.21378	0.021975	0.55555	0.008597
10000 языков (программная генерация)	0.40532	0.047688	2.14932	0.015971
<b>Размеры файлов, Кб</b>				
Общий размер исходных текстов	21.2	15.8	9.8	12.3
Общий размер исходных текстов (на диске)	48	48	24	48
Размер исполняемого файла	44	114	9.3	57

Таким образом, даже частичное использование ассемблера в программе дает многократное преимущество в скорости ее работы. Полагаю, что это связано с тем, что ассемблер работает напрямую с регистрами процессора. Однако по объему занимаемой памяти программа с динамической типизации на python все еще остается оптимальнее других.