

Conclusiones Lab 4 Proa

Ignacio Gomis Lli

Juan Pablo Uriol Balbin

Con la función Memcost1 que utiliza los constructores por defecto obtenemos los tiempos totales, a misma cantidad de operaciones:

CostMem1: 0.484843 s

CostMem2: 11.3025 s

CostMem3: 12.2358 s

CostMem4: 26.7433 s

Como podemos observar, realizar los cambios en la memoria estática es lo mejor en tiempo, y que los unique pointer tienen un rendimiento levemente peor que los punteros simples, aunque dan una mayor protección de los datos. Finalmente, podemos observar que los shared pointers tienen una gran carga de tiempo comparado con los anteriores, debido a la implementación para llevar el conteo de objetos.

Con la función Memcost2 en primer lugar sobrescribimos el operador new y delete con la operación general, consiguiendo un tiempo en CostMem2 de aproximadamente 9 segundos (representa una mejora de 1/6 con respecto a Memcost1)

A continuación sobrescribimos con el método visto en clase de emplear el pool y el allocation map, pero ahí el tiempo se vuelve excesivamente lento, cayendo a un tiempo de 86 segundos (una empeora del 780%)