

Anleitung fuer die Speicherverwaltungs-Testumgebung

Das Programm, mit dem ihr eure Speicherverwaltung testen koennt, steht auf der Vorlesungswebsite zur Verfuegung. Wichtig fuer die Implementierung von `my_alloc()` und `my_free()` sind vier Dateien:

<code>my_system.h</code>	Stellt die Konstante <code>BLOCKSIZE</code> und eine Deklaration der Funktion <code>get_block_from_system()</code> zur Verfuegung
<code>my_system.c</code>	Die Implementierung von <code>get_block_from_system()</code> .
<code>my_alloc.h</code>	Enthaelt die Deklationen von <code>init_my_alloc()</code> , <code>my_alloc()</code> und <code>my_free()</code> .
<code>my_alloc.c</code>	Diese Datei muesst ihr durch eure Implementierung ersetzen. Die mitgelieferte Datei enthaelt eine extrem einfache Beispiel-Implementierung.

Ein ggf. selbst zu schreibendes Testprogramm `mytest.c` kann dann eure Implementierung mit `#include "my_alloc.h"` einbinden und die Funktionen `my_alloc()` und `my_free()` analog zu `malloc()` und `free()` verwenden.

Das mitgelieferte Makefile erstellt beim Aufruf von `make` im selben Verzeichnis das Programm `testit`, das es erlaubt, einen Testlauf durchzufuehren. Dazu muessen zwei Parameter angegeben werden:

- Startwert fuer den Zufallszahlengenerator
- Die (ungefaehre) Anzahl der `my_alloc()` und `my_free()` Operationen, die in diesem Test durchgefuehrt werden sollen.

Ueber weitere, optionale Parameter, kann das Verhaeltnis zwischen `my_alloc()` und `my_free()` Aufrufen und die Groesse der allokierten Speicherbereiche veraendert werden.

Ein Testlauf koennte so aussehen:

```
$ ./testit 1 100000
.....
...
4426640 33444 582 123454
Relative size overhead: 0.077057
Runtime per operation: 0.925151
Points for this test: 90.443989
$
```

Hier wird ein Test mit rund 100000 Operationen durchgefuehrt. Waehrend des Tests werden immer wieder Punkte ausgegeben, um den Fortschritt anzuzeigen. Die anschliessenden Ergebniszeilen haben folgende Bedeutung:

- *maxmem*, *maxcount*, *nrblocks* und die Gesamtzeit in μs
- Relative size overhead: $\frac{\text{wasted}}{\text{base}}$
- Runtime per operation: *time*
- Points for this test: Das errechnete Ergebnis