

Übung Nr. 5:

Erstellen Sie ein kurzes, plattform-unabhängiges C-Programm („Win32-Konsolenanwendung“), das sich in die Reihe ggf. bereits gestarteter Kopien einordnet, sich nach 20 sec aus dieser Reihe abmeldet und sich nach weiteren 20 sec selbsttätig beendet. (Nachbau von Memo.exe)

- Unter <http://homepages.thm.de/christ/> finden Sie weitere Hinweise zum Umgang mit VC++.
- Benutzen Sie bei der Programmierung ausschließlich Anweisungen aus dem Sprachumfang von C!
- Zur Simulation des gemeinsamen Speicherbereichs können Sie eine formatierte Datei verwenden. Sie können sich dabei am Programmbeispiel für die Zeitmessung bei R/W-Operationen orientieren. (MemoTest)
- Bauen Sie Ihr Programm so aus, daß es die benötigte Datei im aktuellen Verzeichnis einrichtet, (erst:) wenn es die erwartete Verzeichnisstruktur („../dat/“) nicht vorfindet. (Damit wird es in jedem Ordner mit Schreiberlaubnis ausführbar.)

Für Hedonist/inn/en (d.h. freiwillig): wahlweise vorzeitige Ausführung des jeweiligen Schrittes durch <CR> und Beendigung durch <ESC> über die Funktion `_kbhit()`.

```
/*          MemoTest.c (gekuerzt)          */
#include <conio.h>
#include <stdio.h>
#include <time.h>
#define WDHLG 10000

int main(void)
{ int j1=0, j2=WDHLG;
  FILE *memo;
  clock_t tj1, tj2;
  printf ("Aris Christidis:\n\r");
  printf ("Formatierte Datei (fscanf/fprintf):\n\r");
  _getch(); tj1 = clock();
  if ((memo=fopen("../dat/Memo.txt","r+"))==NULL &&
      (memo=fopen("../dat/Memo.txt","w+"))==NULL)
    return(-1);
  for (j1=0; j1<WDHLG; j1++)
  { fprintf(memo,"%10d", j1); fflush(memo); rewind(memo);
    fscanf(memo,"%10d", &j2); rewind(memo);
  }
  tj2 = clock(); fclose (memo);
  printf ("Zeit fuer %d RW-Operationen: %.2lf msec\n\n\r",\
          j1, (double)(tj2-tj1)*1000/CLOCKS_PER_SEC);
  /*...*/ _getch(); return(0);
}
```