Lösungswege zum Aufgabenpaket 5

zu Aufgabe 5.1)

Zur Ermittelung der Speicherhierarchie kann man folgendermaßen vorgehen:

1. Anlegen eines Arrays, das größer ist als moderne Cachespeicher
2. Messen der Systemzeit
3. wiederholtes Durchlaufen dieses Arrays vom Start bis zu einem Endindex
4. erneutes Messen der Systemzeit und Ermitteln der Anzahl der Arrayzugriffe
5. Zeitdifferenz * 10e9 / Anzahl der Arrayzugriffe liefert Zugriffszeit
6. Verdoppeln des Endindex
7. Fortsetzen bei Schritt 2.

```
#include <stdio.h>                      /* program MEMTEST.C */
#include <sys/times.h>
#include <sys/types.h>
#include <time.h>
#define CACHE_MIN (1024)                /* smallest cache */
#define CACHE_MAX (1024 * 1024)         /* largest cache */
#define SAMPLE (512)
#ifndef CLK_TCK
#define CLK_TCK 60                      /* number of clock ticks per second */
#endif

int x[CACHE_MAX];                       /* array going to stride through */

double get_seconds() {                  /* routine to read time */
  struct tms rusage;
  times(&rusage);                       /* UNIX utility: time in clock ticks */
  return (double) (rusage.tms_utime)/CLK_TCK;
}

int main() {
  int register i, index, temp;
  int steps, tsteps, csize;
  double sec0, sec;                     /* timing variables */

  for (csize = CACHE_MIN; csize <= CACHE_MAX; csize = csize * 2) {
    sec = 0.;
    sec0 = get_seconds();               /* start timer */
    for (i = 1; i <= 1e3; i++)          /* 1000 samples for better */
      for (index = 0; index < csize; index++)       /* measurements */
          x[index]++;
    sec = get_seconds() - sec0;         /* end timer */

    sec0 = get_seconds();               /* empty loop to get loop overhead */
    for (i = 1; i <= 1e3; i++)
      for (index = 0; index < csize; index++)
          temp++;
    sec = sec - (sec0 - get_seconds());

    printf ("Cachesize: %7d KByte read+write: %14.0fns\n",
              csize * sizeof(int) / 1024,
              sec * 1e9 / (1e3 * csize));
  }
  return 0;
}
```

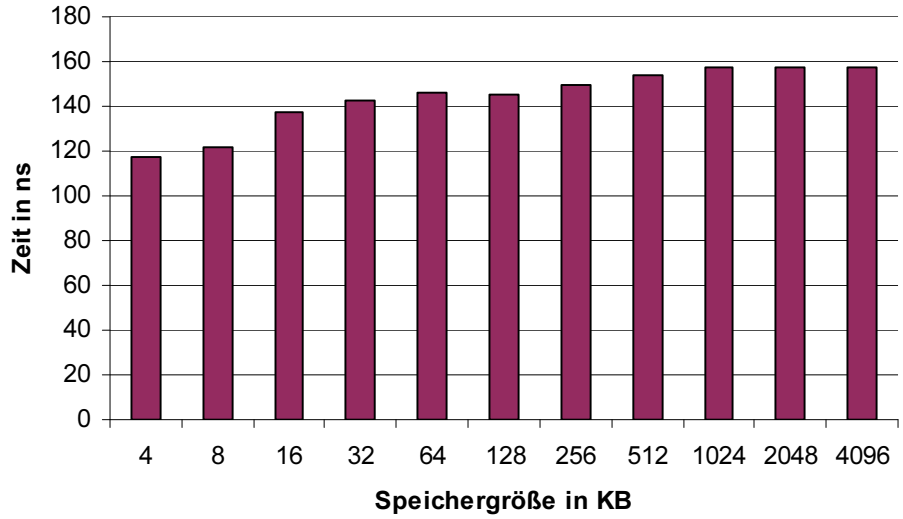
Lösungswege zum Aufgabenpaket 5

zu Aufgabe 5)

Hier die ermittelten Werte auf tipc001:

Cachesize:	4 KByte read+write:	117ns
Cachesize:	8 KByte read+write:	122ns
Cachesize:	16 KByte read+write:	137ns
Cachesize:	32 KByte read+write:	143ns
Cachesize:	64 KByte read+write:	146ns
Cachesize:	128 KByte read+write:	145ns
Cachesize:	256 KByte read+write:	150ns
Cachesize:	512 KByte read+write:	154ns
Cachesize:	1024 KByte read+write:	157ns
Cachesize:	2048 KByte read+write:	157ns
Cachesize:	4096 KByte read+write:	157ns

Daraus ergibt sich für die Speicherhierarchie:
3 levels
8 KByte L1 Cache
128 KByte L2 Cache
> 128 KByte Hauptspeicher

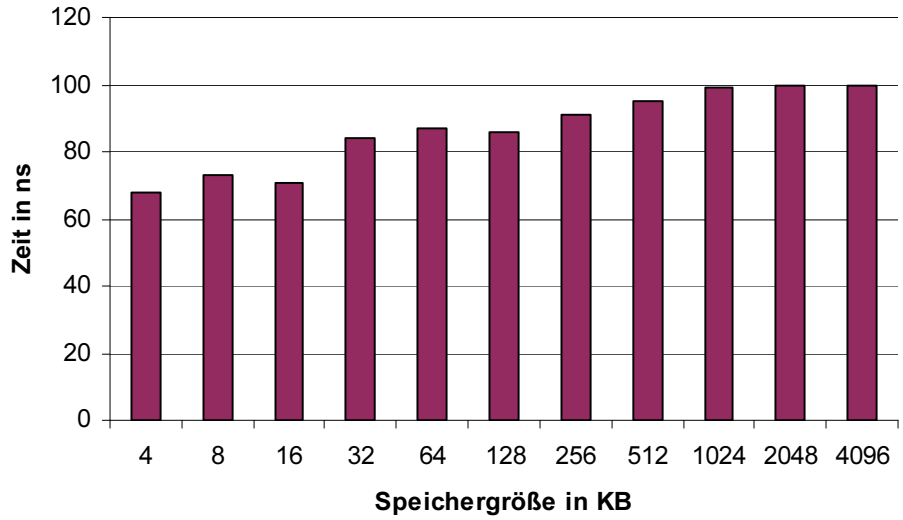


Lösungswege zum Aufgabenpaket 5

Hier die ermittelten Werte auf tipc010:

Cachesize:	4 KByte read+write:	68ns
Cachesize:	8 KByte read+write:	73ns
Cachesize:	16 KByte read+write:	71ns
Cachesize:	32 KByte read+write:	84ns
Cachesize:	64 KByte read+write:	87ns
Cachesize:	128 KByte read+write:	86ns
Cachesize:	256 KByte read+write:	91ns
Cachesize:	512 KByte read+write:	95ns
Cachesize:	1024 KByte read+write:	99ns
Cachesize:	2048 KByte read+write:	100ns
Cachesize:	4096 KByte read+write:	100ns

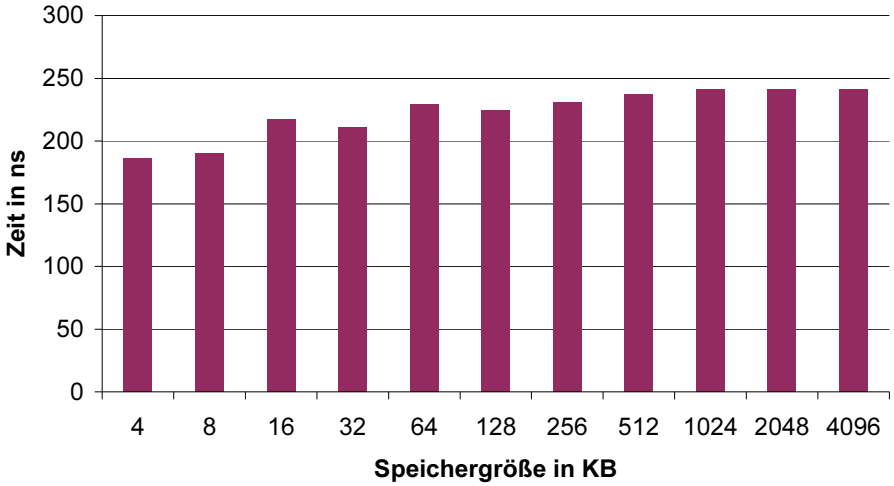
Daraus ergibt sich für die Speicherhierarchie:
3 levels
16 KByte L1 Cache
256 KByte L2 Cache
> 256 KByte Hauptspeicher



Hier die ermittelten Werte auf tipc006:

Cachesize:	4 KByte read+write:	186ns
Cachesize:	8 KByte read+write:	190ns
Cachesize:	16 KByte read+write:	217ns
Cachesize:	32 KByte read+write:	211ns
Cachesize:	64 KByte read+write:	229ns
Cachesize:	128 KByte read+write:	224ns
Cachesize:	256 KByte read+write:	231ns
Cachesize:	512 KByte read+write:	237ns
Cachesize:	1024 KByte read+write:	241ns
Cachesize:	2048 KByte read+write:	241ns
Cachesize:	4096 KByte read+write:	241ns

Daraus ergibt sich für die Speicherhierarchie:
3 levels
8 KByte L1 Cache
128 KByte L2 Cache
> 128 KByte Hauptspeicher



Hier die ermittelten Werte auf tipc001:

Cachesize:	4 KByte read+write:	29ns
Cachesize:	8 KByte read+write:	24ns
Cachesize:	16 KByte read+write:	24ns
Cachesize:	32 KByte read+write:	28ns
Cachesize:	64 KByte read+write:	27ns
Cachesize:	128 KByte read+write:	27ns
Cachesize:	256 KByte read+write:	31ns
Cachesize:	512 KByte read+write:	39ns
Cachesize:	1024 KByte read+write:	52ns
Cachesize:	2048 KByte read+write:	52ns
Cachesize:	4096 KByte read+write:	52ns

Daraus ergibt sich für die Speicherhierarchie:
3 levels
32 KByte L1 Cache
256 KByte L2 Cache
> 256 KByte Hauptspeicher