Eirik Bodsberg – Exercise 1 C programming - TPK4128

1. Linked lists: Running of list_test

```
eirikbod@eirikbod-MS-7A94:~/Documents/TPK4128/Ex1$ make
gcc  linked_list.c linked_list.h list_test.c -o main
eirikbod@eirikbod-MS-7A94:~/Documents/TPK4128/Ex1$ ./main
create list
append valued 0 to 9
0
1
2
3
4
5
6
7
8
9
list sum: 45
insert value in the middle of the list
0
1
2
3
4
5
99
6
7
8
9
get inserted value: 5
0
1
2
3
4
5
99
6
7
8
9
extract inserted value: 5
0
1
2
3
4
99
6
7
8
9
remove all but the last value
9
remove the last value
delete list
eirikbod@eirikbod-MS-7A94:~/Documents/TPK4128/Ex1$
```

Running with Valgrind to check for memory leaks:

```
==9717== HEAP SUMMARY:
==9717==     in use at exit: 0 bytes in 0 blocks
==9717==   total heap usage: 13 allocs, 13 frees, 1,312 bytes allocated
==9717==
==9717== All heap blocks were freed -- no leaks are possible
==9717==
==9717== For lists of detected and suppressed errors, rerun with: -s
==9717== ERROR SUMMARY: 11 errors from 4 contexts (suppressed: 0 from 0)
eirikbod@eirikbod-MS-7A94:~/Documents/TPK4128/Ex1$
```

- No memory leaks

2. Error checking

I had no difference in running the program with/without error checks. Possible that I did it wrong. See code in git reposetory

3. Sleep and busy-wait

Execution time is the time the system kernel use to execute a process, if the task sleep or another task is run, the execution time is not incremented. Absolute time is the wall clock time, time the process use from start to end, regardless of how much time the kernel spends on this task. Pausing a process or prioritizing different processes will add to absolute time.

Running the 2 threaded process using the sleep function.

```
eirikbod@eirikbod-MS-7A94:~/Documents/TPK4128/Ex1$ time ./a.out
Message 1
Message 1
Message 2
Message 2

real    0m5,001s
user    0m0,001s
sys     0m0,000s
```

Running the 2 threaded process using a busy-and-wait approach.

```
eirikbod@eirikbod-MS-7A94:~/Documents/TPK4128/Ex1$ time ./a.out
Message 1
Message 1
Message 2
Message 2

real    0m2,957s
user    0m5,000s
sys     0m0,913s
eirikbod@eirikbod-MS-7A94:~/Documents/TPK4128/Ex1$ time ./a.out
Message 1
Message 1
Message 2
Message 2

real    0m2,999s
user    0m5,000s
sys     0m0,996s
eirikbod@eirikbod-MS-7A94:~/Documents/TPK4128/Ex1$ time ./a.out
Message 1
Message 1
Message 2
Message 2

real    0m2,895s
user    0m5,000s
sys     0m0,789s
eirikbod@eirikbod-MS-7A94:~/Documents/TPK4128/Ex1$ time ./a.out
Message 1
Message 1
Message 2
Message 2

real    0m2,913s
user    0m5,001s
sys     0m0,825s
eirikbod@eirikbod-MS-7A94:~/Documents/TPK4128/Ex1$ time ./a.out
Message 1
Message 1
Message 2
Message 2

real    0m2,945s
user    0m5,001s
sys     0m0,888s
eirikbod@eirikbod-MS-7A94:~/Documents/TPK4128/Ex1$ time ./a.out
Message 1
Message 1
Message 2
Message 2

real    0m2,935s
user    0m5,001s
sys     0m0,868s
eirikbod@eirikbod-MS-7A94:~/Documents/TPK4128/Ex1$
```

The “user time” is the amount of time on the cpu, outside the kernel. Since the busy-and-wait algorithm execute tasks until enough clock cycles is passed to reach 5s, the user time is around 5s, for every process run. Since the process may run on several cores, the absolute time is actually shorter than the user time. And this is causing the difference in absolute time in the different processes.