Operating Systems 2016/17
Tutorial-Assignment 4

Prof. Dr. Frank Bellosa
Dipl.-Inform. Marc Rittinghaus

Question 4.1: Understanding Assignment P3.1

a. Recap how you can round up a given integer to the next larger multiple of a predefined power of two (e.g., 16) in C with simple integer and binary arithmetic only.

b. Look at the declaration of `Block`. Why is the `padding` field required? How does it work?

```
typedef struct _Block {
    struct _Block *next;
    uint8_t padding[8 - sizeof(void*)];
    uint64_t size;
    uint8_t data[];
} Block;
```

c. What are the major steps for allocating memory as requested by the assignment? Where can you find the respective operations in the solution?

d. During allocation, a free block might be split into two blocks: one for the allocated space and one for the remaining free space. Can you think of a reason, why the assignment prefers returning the first block, although the implementation might be easier the opposite way?

e. What are the major steps for freeing memory as requested by the assignment? Where can you find the respective operations in the solution?

f. In the solution you will find many `assert` statements. What is their purpose? Why doesn't the solution use regular `if` statements instead?

Question 4.2: Understanding Assignment P3.2

a. Consider the following code from `dispatcher.c`, lines 122 to 136. Explain what components make up the assembly fragment.

```
__asm__ __volatile__ (

    "nop" // No-operation

    : [prevSp] "=m" (_threads[prevThread].currentSP)
    : [newSp] "m" (_threads[_currentThread].currentSP)
    : "cc", "memory"
);
```

b. Describe how the stack develops during the execution of the `yield` function. Visualize the stack contents for interesting points.