

19579 - PR: Programming Wireless Sensor Networks

Threads, Stacks and Messages in μ kleos

Oliver Hahm, Kaspar Schleiser

Department of Mathematics and Computer Science

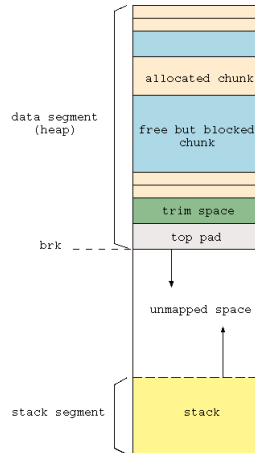
Institute of Computer Science

October 27th, 2010

1. Memory in C
2. Stacks in μ kleos
3. Multithreading & IPC in μ kleos

Allocation methods in C:

1. statically
 - Size has to be chosen at compile time, can't be freed
2. on stack
 - Stack is limited, memory only valid in the current scope
3. on heap
 - Costly, possible fragmentation, memory leaks possible



```
#include <malloc.h>

char allocated_statically[1024];

int main(void) {
    char allocated_on_stack[1024];

    for (int i = 0; i < 100; i++) {
        char also_allocated_on_stack[1024];
    }

    char *allocated_from_heap = malloc(1024);
    if (char == NULL) {
        // ERROR
    }
}
```

- No MMU
- Stack cannot be shared
- Stack cannot be resized
- Stack overflow cannot be prevented

Careful stack handling necessary!

Stackdaten	Thread 1
Freier Speicher	
Stackdaten	Thread2
Freier Speicher	
Stackdaten	Thread2
Freier Speicher	
Stackdaten	Thread2
Freier Speicher	

μ kleos Scheduler:

- priority-based
- preemptive
- “tickless”
- Simple scheduling policy:

Ein Prozess wird nur durch Interrupts oder von einem Prozess mit höherer Priorität unterbrochen.

μ kleos Threading-Funktionen

Funktionsdeklaration	Beschreibung
<code>thread_create(...)</code>	Erzeugt einen neuen Thread
<code>thread_sleep(...)</code>	Legt den laufenden Thread schlafen
<code>thread_wakeup(pid)</code>	Weckt einen Thread auf
<code>fk_yield()</code>	Ruft den scheduler auf

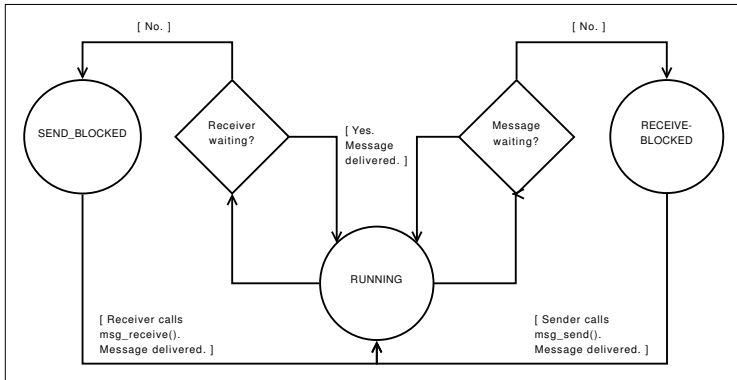
For details, check the API docs!

μ kleos Messaging-Funktionen

Funktionsdeklaration	Beschreibung
<code>int msg_send(msg* m, unsigned int target_pid, bool block);</code>	Sendet eine Nachricht
<code>int msg_receive(msg* m);</code>	Empfängt eine Nachricht
<code>int msg_send_receive(msg* m, msg* reply, int target_pid);</code>	Sendet eine Nachricht und wartet auf Antwort
<code>int msg_reply(msg* m, msg* reply);</code>	Antwortet

For details, check the API docs!

μ kleos Messaging Zustandsdiagramm



```
int main(void)
{
    msg m;
    int pid = thread_create(KERNEL_CONF_STACKSIZE_MAIN, PRIORITY_MAIN-1,
                           CREATE_WOUT_YIELD, second_thread, "pong");
    m.content.value = 1;
    while(1) {
        msg_send(&m, pid, true);
    }
}

void second_thread(void) {
    msg m;

    while(1) {
        msg_receive(&m);
        printf("2nd: got msg from %u with content %i\n", m.sender_pid,
              m.content.value);
    }
}
```

Questions?