

## Skapa run-time bibliotek för MD407

Som del av laboration 5 ska du skapa ett programbibliotek libMD407.a som kompletterar med de hårdvaruberoende run-time funktionerna



```

/*
 * libMD407.h
 * Declaration of library functions, constants etc...
 */
#include <stdio.h>
#include <errno.h>
#include <sys/stat.h>
/* Type definitions */
typedef struct
{
    ...
} DEV_DRIVER_DESC, *PDEV_DRIVER_DESC;

/* Constants */
#define MAX_FILENO 5
/* Constant defined by linker */
extern char _heap_low;
extern char _heap_top;
extern char _bss_start;
extern char _bss_end;

/* Library defined functions */
void crt_init( void );
void crt_deinit( void );

char *_sbrk(int);
int _close(int file);
int _fstat(int file, struct stat *st);
etc ...
  
```

```

/*
 * libMD407.h
 * Typer och konstanter som används i hela programbiblioteket
 */
#include <stdio.h>
#include <errno.h>
#include <sys/stat.h>
/* Type definitions */
#include <sys/stat.h>

typedef struct
{
    char name[16];
    int (*init) (int);
    void (*deinit) (int);
    int (*fstat)(struct stat *st);
    int (*isatty)(void);
    int (*open)(const char name,int flags,int mode);
    int (*close)(void);
    int (*lseek)(int ptr, int dir);
    int (*write)(char *ptr, int len);
    int (*read)(char *ptr, int len);
} DEV_DRIVER_DESC, *PDEV_DRIVER_DESC;

/* File handle constants */
enum {STDIN=0,STDOUT,STDERR,KEYPAD,ASCIIDISPLAY};
#define MAX_FILENO ASCIIDISPLAY

/* Library defined functions */
int _isatty(int file);
  
```

```

/*
 * libMD407
 * general.c
 * includes: _crt_init, _crt_deinit, _sbrk for malloc-support.
 * template functions for stdio:
 */
#include      "libMD407.h"

extern DEV_DRIVER_DESC StdIn,StdOut,StdErr,KeyPad,AsciiDisplay;

PDEV_DRIVER_DESC device_table[MAX_FILENO+1] =
{
    &StdIn,
    &StdOut,
    &StdErr,
    &KeyPad,
    &AsciiDisplay,
};

static char *heap_end;
extern char __heap_low; /* Defined by the linker */
extern char __heap_top; /* Defined by the linker */
extern char __bss_start__; /* Defined by the linker */
extern char __bss_end__; /* Defined by the linker */

char * _sbrk(int incr) { ... }

void _crt_init() {
    char *s;
    PDEV_DRIVER_DESC fd;

    heap_end = 0;
    s = &__bss_start__;
    while( s < &__bss_end__ )
        *s++ = 0;
    s = &__heap_low;
    while( s < &__heap_top )
        *s++ = 0;
    for( int i = 0; i <= MAX_FILENO; i++ )
    {
        fd = device_table[i];
        if( fd && fd->init != 0)
            (void) fd->init( 0 );
    }
    /* NOTE: No printf buffering */
    (void)setvbuf(stdin, 0, _IONBF, 0);
    (void)setvbuf(stdout, 0, _IONBF, 0);
    (void)setvbuf(stderr, 0, _IONBF, 0);
}

void _crt_deinit() {
    PDEV_DRIVER_DESC fd;
    fflush(0); /* Will cause terminal flush... */
    for( int i = 0; i <= MAX_FILENO; i++ )
    {
        fd = device_table[i];
        if( fd && fd->deinit != 0)
            fd->deinit( 0 );
    }
}
/* Anm. parametrar till respektive device init/deinit är valbar */

/* STDIO-common functions */
int _close(int file) { return -1; }
int _open(const char *name, int flags, int mode) { return -1; }
int _fstat(int file, struct stat *st) { st->st_mode = S_IFCHR; return 0; }
int _lseek(int file, int ptr, int dir) { return 0; }

int _isatty(int file) {
    switch (file)
    {
        case STDIN: case STDOUT: case STDERR: return 1;
        default: return 0;
    }
}

int _write(int file, char *ptr, int len) { ... }
int _read(int file, char *ptr, int len) { ... }

```

```

/*
 *      libMD407
 *  usart_driver.c
 *  USART driver
 *  Drivers for STDIN, STDOUT and STDERR
 */
#include     "libMD407.h"

/* Define functions here */
static int usart_init( int initval );
static void usart_deinit( int deinitval);
static int usart_write(char *ptr, int len);
static int usart_read(char *ptr, int len);

DEV_DRIVER_DESC StdIn =
{
    {"stdin"},           /* device name */
    usart_init,          /* open function */
    usart_deinit,        /* close function */
    0,                  /* read timeout */
    _isatty,             /* isatty flag */
    0,                  /* read buffer size */
    0,                  /* write buffer size */
    0,                  /* read buffer */
    usart_read           /* read function */
};

DEV_DRIVER_DESC StdOut =
{
    {"stdout"},           /* device name */
    usart_init,          /* open function */
    usart_deinit,        /* close function */
    0,                  /* read timeout */
    _isatty,             /* isatty flag */
    0,                  /* read buffer size */
    0,                  /* write buffer size */
    0,                  /* read buffer */
    usart_write          /* write function */
};

DEV_DRIVER_DESC StdErr =
{
    {"stderr"},           /* device name */
    usart_init,          /* open function */
    usart_deinit,        /* close function */
    0,                  /* read timeout */
    _isatty,             /* isatty flag */
    0,                  /* read buffer size */
    0,                  /* write buffer size */
    0,                  /* read buffer */
    usart_write          /* write function */
};

/* USART types and constants definitions */
.....
/* USART implementation */

static int usart_init( int initval ) { ... }
static void usart_deinit( int deinitval){ ... }
static int usart_write(char *ptr, int len){ ... }
static int usart_read(char *ptr, int len){ ... }

```

```

/*
 *      libMD407
 *  keypad_driver.c
 *  Keypad connected to PD8-PD15
 *  Driver for KEYPAD
 */
#include      "libMD407.h"

static int keypad_init( int initval );
static void keypad_deinit( int deinitval);
static int keypad_read(char *ptr, int len);

DEV_DRIVER_DESC KeyPad =
{
    {"Keypad"},           // Name of driver
    keypad_init,          // Function to initialize
    keypad_deinit,        // Function to deinitialize
    0,                   // Number of pins
    0,                   // Pin 1
    0,                   // Pin 2
    0,                   // Pin 3
    0,                   // Pin 4
    0,                   // Pin 5
    0,                   // Pin 6
    0,                   // Pin 7
    0,                   // Pin 8
    0,                   // Pin 9
    0,                   // Pin 10
    0,                   // Pin 11
    0,                   // Pin 12
    0,                   // Pin 13
    0,                   // Pin 14
    0,                   // Pin 15
    keypad_read          // Function to read
};

/* KEYPAD types and constants definitions */
...

/* Define functions here */
static int keypad_init( int initval ){ ... }
static void keypad_deinit( int deinitval){ ... }
static int keypad_read(char *ptr, int len){ ... }

```

```

/*
 *      libMD407
 *  asciidisplay_driver.c
 *  Display connected to PE
 *  Driver for ASCIIDISPLAY
 */
#include      "libMD407.h"

static int asciidisplay_init( int initval );
static void asciidisplay_deinit( int deinitval);
static int asciidisplay_write(char *ptr, int len);

DEV_DRIVER_DESC AsciiDisplay =
{
    {"AsciiDisplay"},           // Name of driver
    asciidisplay_init,          // Function to initialize
    asciidisplay_deinit,        // Function to deinitialize
    0,                           // Number of pins
    0,                           // Pin 1
    0,                           // Pin 2
    0,                           // Pin 3
    0,                           // Pin 4
    0,                           // Pin 5
    0,                           // Pin 6
    0,                           // Pin 7
    0,                           // Pin 8
    0,                           // Pin 9
    0,                           // Pin 10
    0,                           // Pin 11
    0,                           // Pin 12
    0,                           // Pin 13
    0,                           // Pin 14
    0,                           // Pin 15
    asciidisplay_write,         // Function to write
    0
};

/* ASCIIDISPLAY types and constants definitions */
...

/* Define functions here */
static int asciidisplay_init( int initval ){ ... }
static void asciidisplay_deinit( int deinitval){ ... }
static int asciidisplay_write(char *ptr, int len){ ... }

```