allocs.h: xmicro's allocator

Scope

No super permissions needed.

Name

`allocs.h` is the allocator implementation for xmicro.

Synopsis

```c
#include <types.h>
#include <allocs.h>
void* alloc(ssize_t length);
void free(void* ptr);
```

Description

`alloc` allocates length bytes on memory and returns the location of the memory, returns `NULL` in failure. `free` frees a part of the memory if address is not `NULL` and is a malloc address and is owned by current process.

errno.h: xmicro's error helper

Scope

No super permissions needed.

Name

`errno.h` is the errno variable implementation for xmicro.

Synopsis

```c
#include <types.h>
#include <errno.h>
typedef enum {
    ENOERR = 0, // No error
    ENOIMPL,    // Not implemented
    ENSCOPE,    // Wrong scope
    ENQFULL,    // Queue full
    ENMFAIL,    // Message send failed
    ENUNKW,     // Unknown error
    ENOPID,     // No such PID
    EMPLEP,     // Message payload empty
```

```
    EMSINC      // Message size incorrect
} errno_t;
extern int (*__errno_pointer)(void);
#define errno *(_errno_pointer())
```

Description

Implements `errno_t` enum and `errno` symbol.

intercom.h: xmicro's IPC

Scope

No super permissions needed (except for an edge case in kmsg).

Name

`intercom.h` is the glue between the kernel and a thread or a task.

Synopsis

```
#include <types.h>
#include <intercom.h>
typedef struct {
    pid_t sender;
    pid_t receiver;
    void* payload;
    ssize_t size;
} msg_t;
msg_t* cmsg(pid_t receiver, void* payload, ssize_t psiz);
/* Creates a message to send to the kernel for it to send it to a process */
int chqueue();
/* Check message queue (receive) */
void clqueue();
/* Clear message queue (receive) */
int kmsg(msg_t* message);
/* Sends message to kernel */
```

Description

`intercom.h` is used to send messages to processes in the kernel. `intercom.h` must define the `msg_t` type.

cmsg: Create message packet

Name

`cmsg` creates a message to send to a PID with a payload.

Synopsis

```
#include <intercom.h>
msg_t* cmsg(pid_t receiver, void* payload, ssize_t psiz);
```

Description

cmsg creates a msg_t *packet to send to* ***`receiver`*** *with the payload* ***`payload`***. cmsg* returns `NULL` and sets `errno` if the supplied `pid_t`, `void*` or `ssize_t` argument is wrong. PID 0 sends a kernel message and requires super permissions.

kmsg: Send message packet

Name

`kmsg` sends a message crafted with `cmsg` to the kernel.

Synopsis

```
#include <intercom.h>
int kmsg(msg_t* message);
```

Description

kmsg sends `message` to the PID Message Queue. If it fails, it returns `-1` and sets `errno`. `sender` field is replaced in the *kmsg* function to keep message integrity.

types.h: xmicro's type definitions

Scope

No super permissions needed.

Name

`types.h` is the type implementation for xmicro.

Synopsis

```
#include <types.h>
typedef unsigned long long max_t;
typedef signed long long size_t;
typedef max_t ssize_t;
typedef max_t pid_t;
#define NULL ((void*) 0)
```

Description

Implements required types for xmicro. - max_t - pid_t - size_t - ssize_t