The Helpless Four OS

3.0

Generated by Doxygen 1.9.1

Chapter 1

Data Structure Index

1.1 Data Structures

Here are the data structures with brief descriptions:

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

Chapter 3

Data Structure Documentation

3.1 alarm Struct Reference

`#include <R4.h>`

Data Fields

- char message [20]
- int time
- struct alarm * NextPtr
- struct alarm * PrevPtr

3.1.1 Detailed Description

Definition at line 21 of file R4.h.

3.1.2 Field Documentation

3.1.2.1 message

`char message[20]`

Definition at line 24 of file R4.h.

3.1.2.2 NextPtr

`struct alarm* NextPtr`

Definition at line 27 of file R4.h.

3.1.2.3 PrevPtr

```
struct alarm* PrevPtr
```

Definition at line 28 of file R4.h.

3.1.2.4 time

```
int time
```

Definition at line 26 of file R4.h.

The documentation for this struct was generated from the following file:

- MPX-cs450/mpx_core/modules/R4.h

3.2 alList Struct Reference

```
#include <R4.h>
```

Data Fields

- int numAlarms
- struct alarm * head
- struct alarm * tail

3.2.1 Detailed Description

Definition at line 31 of file R4.h.

3.2.2 Field Documentation

3.2.2.1 head

```
struct alarm* head
```

Definition at line 33 of file R4.h.

3.2.2.2 numAlarms

```
int numAlarms
```

Definition at line 32 of file R4.h.

3.2.2.3 tail

```
struct alarm* tail
```

Definition at line 34 of file R4.h.

The documentation for this struct was generated from the following file:

- MPX-cs450/mpx_core/modules/R4.h

3.3 allocatedblocks Struct Reference

```
#include <R5.h>
```

Data Fields

- struct CMCB * head

3.3.1 Detailed Description

Definition at line 43 of file R5.h.

3.3.2 Field Documentation

3.3.2.1 head

```
struct CMCB* head
```

Definition at line 44 of file R5.h.

The documentation for this struct was generated from the following file:

- MPX-cs450/mpx_core/modules/R5.h

3.4 CMCB Struct Reference

`#include <R5.h>`

Data Fields

- int type
- void * addr
- int size
- char pcbname [15]
- struct CMCB * next
- struct CMCB * prev

3.4.1 Detailed Description

Definition at line 22 of file R5.h.

3.4.2 Field Documentation

3.4.2.1 addr

`void* addr`

Definition at line 25 of file R5.h.

3.4.2.2 next

`struct CMCB* next`

Definition at line 29 of file R5.h.

3.4.2.3 pcbname

`char pcbname[15]`

Definition at line 28 of file R5.h.

3.4.2.4 prev

```
struct CMCB* prev
```

Definition at line 30 of file R5.h.

3.4.2.5 size

```
int size
```

Definition at line 27 of file R5.h.

3.4.2.6 type

```
int type
```

Definition at line 24 of file R5.h.

The documentation for this struct was generated from the following file:

- MPX-cs450/mpx_core/modules/R5.h

3.5 context Struct Reference

```
#include <R2.h>
```

Data Fields

- u32int gs
- u32int fs
- u32int es
- u32int ds
- u32int edi
- u32int esi
- u32int ebp
- u32int esp
- u32int ebx
- u32int edx
- u32int ecx
- u32int eax
- u32int eip
- u32int cs
- u32int eflags

3.5.1 Detailed Description

Definition at line 44 of file R2.h.

3.5.2 Field Documentation

3.5.2.1 cs

```
u32int cs
```

Definition at line 47 of file R2.h.

3.5.2.2 ds

```
u32int ds
```

Definition at line 45 of file R2.h.

3.5.2.3 eax

```
u32int eax
```

Definition at line 46 of file R2.h.

3.5.2.4 ebp

```
u32int ebp
```

Definition at line 46 of file R2.h.

3.5.2.5 ebx

```
u32int ebx
```

Definition at line 46 of file R2.h.

3.5.2.6 ecx

```
u32int ecx
```

Definition at line 46 of file R2.h.

3.5.2.7 edi

```
u32int edi
```

Definition at line 46 of file R2.h.

3.5.2.8 edx

```
u32int edx
```

Definition at line 46 of file R2.h.

3.5.2.9 eflags

```
u32int eflags
```

Definition at line 47 of file R2.h.

3.5.2.10 eip

```
u32int eip
```

Definition at line 47 of file R2.h.

3.5.2.11 es

```
u32int es
```

Definition at line 45 of file R2.h.

3.5.2.12 esi

```
u32int esi
```

Definition at line 46 of file R2.h.

3.5.2.13 esp

```
u32int esp
```

Definition at line 46 of file R2.h.

3.5.2.14 fs

```
u32int fs
```

Definition at line 45 of file R2.h.

3.5.2.15 gs

```
u32int gs
```

Definition at line 45 of file R2.h.

The documentation for this struct was generated from the following file:

- MPX-cs450/mpx_core/modules/R2.h

3.6 date_time Struct Reference

```
#include <system.h>
```

Data Fields

- int sec
- int min
- int hour
- int day_w
- int day_m
- int day_y
- int mon
- int year

3.6.1 Detailed Description

Definition at line 30 of file system.h.

3.6.2 Field Documentation

3.6.2.1 day_m

```
int day_m
```

Definition at line 35 of file system.h.

3.6.2.2 day_w

```
int day_w
```

Definition at line 34 of file system.h.

3.6.2.3 day_y

```
int day_y
```

Definition at line 36 of file system.h.

3.6.2.4 hour

```
int hour
```

Definition at line 33 of file system.h.

3.6.2.5 min

```
int min
```

Definition at line 32 of file system.h.

3.6.2.6 mon

```
int mon
```

Definition at line 37 of file system.h.

3.6.2.7 sec

```
int sec
```

Definition at line 31 of file system.h.

3.6.2.8 year

```
int year
```

Definition at line 38 of file system.h.

The documentation for this struct was generated from the following file:

- MPX-cs450/mpx_core/include/system.h

3.7 footer Struct Reference

```
#include <heap.h>
```

Data Fields

- header head

3.7.1 Detailed Description

Definition at line 16 of file heap.h.

3.7.2 Field Documentation

3.7.2.1 head

```
header head
```

Definition at line 17 of file heap.h.

The documentation for this struct was generated from the following file:

- MPX-cs450/mpx_core/include/mem/heap.h

3.8 freeblocks Struct Reference

```
#include <R5.h>
```

Data Fields

- struct CMCB * head

3.8.1 Detailed Description

Definition at line 47 of file R5.h.

3.8.2 Field Documentation

3.8.2.1 head

```
struct CMCB* head
```

Definition at line 48 of file R5.h.

The documentation for this struct was generated from the following file:

- MPX-cs450/mpx_core/modules/R5.h

3.9 gdt_descriptor_struct Struct Reference

```
#include <tables.h>
```

Data Fields

- u16int limit
- u32int base

3.9.1 Detailed Description

Definition at line 23 of file tables.h.

3.9.2 Field Documentation

3.9.2.1 base

```
u32int base
```

Definition at line 26 of file tables.h.

3.9.2.2 limit

```
u16int limit
```

Definition at line 25 of file tables.h.

The documentation for this struct was generated from the following file:

- MPX-cs450/mpx_core/include/core/tables.h

3.10 gdt_entry_struct Struct Reference

```
#include <tables.h>
```

Data Fields

- u16int limit_low
- u16int base_low
- u8int base_mid
- u8int access
- u8int flags
- u8int base_high

3.10.1 Detailed Description

Definition at line 30 of file tables.h.

3.10.2 Field Documentation

3.10.2.1 access

```
u8int access
```

Definition at line 35 of file tables.h.

3.10.2.2 base_high

```
u8int base_high
```

Definition at line 37 of file tables.h.

3.10.2.3 base_low

```
u16int base_low
```

Definition at line 33 of file tables.h.

3.10.2.4 base_mid

```
u8int base_mid
```

Definition at line 34 of file tables.h.

3.10.2.5 flags

```
u8int flags
```

Definition at line 36 of file tables.h.

3.10.2.6 limit_low

```
u16int limit_low
```

Definition at line 32 of file tables.h.

The documentation for this struct was generated from the following file:

- MPX-cs450/mpx_core/include/core/tables.h

3.11 header Struct Reference

```
#include <heap.h>
```

Data Fields

- int size
- int index_id

3.11.1 Detailed Description

Definition at line 11 of file heap.h.

3.11.2 Field Documentation

3.11.2.1 index_id

```
int index_id
```

Definition at line 13 of file heap.h.

3.11.2.2 size

```
int size
```

Definition at line 12 of file heap.h.

The documentation for this struct was generated from the following file:

- MPX-cs450/mpx_core/include/mem/heap.h

3.12 heap Struct Reference

`#include <heap.h>`

Data Fields

- index_table index
- u32int base
- u32int max_size
- u32int min_size
- char size [1000]

3.12.1 Detailed Description

Definition at line 33 of file heap.h.

3.12.2 Field Documentation

3.12.2.1 base

`u32int base`

Definition at line 35 of file heap.h.

3.12.2.2 index

`index_table index`

Definition at line 34 of file heap.h.

3.12.2.3 max_size

`u32int max_size`

Definition at line 36 of file heap.h.

3.12.2.4 min_size

```
u32int min_size
```

Definition at line 37 of file heap.h.

3.12.2.5 size

```
char size[1000]
```

Definition at line 20 of file R5.h.

The documentation for this struct was generated from the following files:

- MPX-cs450/mpx_core/include/mem/heap.h
- MPX-cs450/mpx_core/modules/R5.h

3.13 history Struct Reference

```
#include <serial.h>
```

Data Fields

- char * buffer
- struct history * next
- struct history * prev

3.13.1 Detailed Description

Definition at line 49 of file serial.h.

3.13.2 Field Documentation

3.13.2.1 buffer

```
char* buffer
```

Definition at line 51 of file serial.h.

3.13.2.2 next

```
struct history* next
```

Definition at line 52 of file serial.h.

3.13.2.3 prev

```
struct history* prev
```

Definition at line 53 of file serial.h.

The documentation for this struct was generated from the following file:

- MPX-cs450/mpx_core/include/core/serial.h

3.14 idt_entry_struct Struct Reference

```
#include <tables.h>
```

Data Fields

- u16int base_low
- u16int sselect
- u8int zero
- u8int flags
- u16int base_high

3.14.1 Detailed Description

Definition at line 6 of file tables.h.

3.14.2 Field Documentation

3.14.2.1 base_high

```
u16int base_high
```

Definition at line 12 of file tables.h.

3.14.2.2 base_low

```
u16int base_low
```

Definition at line 8 of file tables.h.

3.14.2.3 flags

```
u8int flags
```

Definition at line 11 of file tables.h.

3.14.2.4 sselect

```
u16int sselect
```

Definition at line 9 of file tables.h.

3.14.2.5 zero

```
u8int zero
```

Definition at line 10 of file tables.h.

The documentation for this struct was generated from the following file:

- MPX-cs450/mpx_core/include/core/tables.h

3.15 idt_struct Struct Reference

```
#include <tables.h>
```

Data Fields

- u16int limit
- u32int base

3.15.1 Detailed Description

Definition at line 16 of file tables.h.

3.15.2 Field Documentation

3.15.2.1 base

```
u32int base
```

Definition at line 19 of file tables.h.

3.15.2.2 limit

```
u16int limit
```

Definition at line 18 of file tables.h.

The documentation for this struct was generated from the following file:

- MPX-cs450/mpx_core/include/core/tables.h

3.16 index_entry Struct Reference

```
#include <heap.h>
```

Data Fields

- int size
- int empty
- u32int block

3.16.1 Detailed Description

Definition at line 20 of file heap.h.

3.16.2 Field Documentation

3.16.2.1 block

```
u32int block
```

Definition at line 23 of file heap.h.

3.16.2.2 empty

```
int empty
```

Definition at line 22 of file heap.h.

3.16.2.3 size

```
int size
```

Definition at line 21 of file heap.h.

The documentation for this struct was generated from the following file:

- MPX-cs450/mpx_core/include/mem/heap.h

3.17 index_table Struct Reference

```
#include <heap.h>
```

Data Fields

- index_entry table [TABLE_SIZE]
- int id

3.17.1 Detailed Description

Definition at line 27 of file heap.h.

3.17.2 Field Documentation

3.17.2.1 id

```
int id
```

Definition at line 29 of file heap.h.

3.17.2.2 table

```
index_entry table[TABLE_SIZE]
```

Definition at line 28 of file heap.h.

The documentation for this struct was generated from the following file:

- MPX-cs450/mpx_core/include/mem/heap.h

3.18 page_dir Struct Reference

```
#include <paging.h>
```

Data Fields

- page_table * tables [1024]
- u32int tables_phys [1024]

3.18.1 Detailed Description

Definition at line 34 of file paging.h.

3.18.2 Field Documentation

3.18.2.1 tables

```
page_table* tables[1024]
```

Definition at line 35 of file paging.h.

3.18.2.2 tables_phys

```
u32int tables_phys[1024]
```

Definition at line 36 of file paging.h.

The documentation for this struct was generated from the following file:

- MPX-cs450/mpx_core/include/mem/paging.h

3.19 page_entry Struct Reference

`#include <paging.h>`

Data Fields

- u32int present: 1
- u32int writeable: 1
- u32int usermode: 1
- u32int accessed: 1
- u32int dirty: 1
- u32int reserved: 7
- u32int frameaddr: 20

3.19.1 Detailed Description

Definition at line 12 of file paging.h.

3.19.2 Field Documentation

3.19.2.1 accessed

`u32int accessed`

Definition at line 16 of file paging.h.

3.19.2.2 dirty

`u32int dirty`

Definition at line 17 of file paging.h.

3.19.2.3 frameaddr

`u32int frameaddr`

Definition at line 19 of file paging.h.

3.19.2.4 present

```
u32int present
```

Definition at line 13 of file paging.h.

3.19.2.5 reserved

```
u32int reserved
```

Definition at line 18 of file paging.h.

3.19.2.6 usermode

```
u32int usermode
```

Definition at line 15 of file paging.h.

3.19.2.7 writeable

```
u32int writeable
```

Definition at line 14 of file paging.h.

The documentation for this struct was generated from the following file:

- MPX-cs450/mpx_core/include/mem/paging.h

3.20 page_table Struct Reference

```
#include <paging.h>
```

Data Fields

- page_entry pages [1024]

3.20.1 Detailed Description

Definition at line 26 of file paging.h.

3.20.2 Field Documentation

3.20.2.1 pages

```
page_entry pages[1024]
```

Definition at line 27 of file paging.h.

The documentation for this struct was generated from the following file:

- MPX-cs450/mpx_core/include/mem/paging.h

3.21 param Struct Reference

```
#include <mpx_supt.h>
```

Data Fields

- int op_code
- int device_id
- char * buffer_ptr
- int * count_ptr

3.21.1 Detailed Description

Definition at line 31 of file mpx_supt.h.

3.21.2 Field Documentation

3.21.2.1 buffer_ptr

```
char* buffer_ptr
```

Definition at line 34 of file mpx_supt.h.

3.21.2.2 count_ptr

```
int* count_ptr
```

Definition at line 35 of file mpx_supt.h.

3.21.2.3 device_id

```
int device_id
```

Definition at line 33 of file mpx_supt.h.

3.21.2.4 op_code

```
int op_code
```

Definition at line 32 of file mpx_supt.h.

The documentation for this struct was generated from the following file:

- MPX-cs450/mpx_core/modules/mpx_supt.h

3.22 pcb Struct Reference

```
#include <R2.h>
```

Data Fields

- char name [10]
- int class
- int priority
- int state
- int suspended
- unsigned char stack [1024]
- unsigned char * top
- struct pcb * NextPtr
- struct pcb * PrevPtr

3.22.1 Detailed Description

Definition at line 19 of file R2.h.

3.22.2 Field Documentation

3.22.2.1 class

```
int class
```

Definition at line 23 of file R2.h.

3.22.2.2 name

```
char name[10]
```

Definition at line 21 of file R2.h.

3.22.2.3 NextPtr

```
struct pcb* NextPtr
```

Definition at line 34 of file R2.h.

3.22.2.4 PrevPtr

```
struct pcb* PrevPtr
```

Definition at line 35 of file R2.h.

3.22.2.5 priority

```
int priority
```

Definition at line 25 of file R2.h.

3.22.2.6 stack

```
unsigned char stack[1024]
```

Definition at line 31 of file R2.h.

3.22.2.7 state

```
int state
```

Definition at line 27 of file R2.h.

3.22.2.8 suspended

```
int suspended
```

Definition at line 29 of file R2.h.

3.22.2.9 top

```
unsigned char* top
```

Definition at line 32 of file R2.h.

The documentation for this struct was generated from the following file:

- MPX-cs450/mpx_core/modules/R2.h

3.23 Queue Struct Reference

```
#include <R2.h>
```

Data Fields

- int numProcesses
- struct pcb * head
- struct pcb * tail

3.23.1 Detailed Description

Definition at line 38 of file R2.h.

3.23.2 Field Documentation

3.23.2.1 head

```
struct pcb* head
```

Definition at line 40 of file R2.h.

3.23.2.2 numProcesses

```
int numProcesses
```

Definition at line 39 of file R2.h.

3.23.2.3 tail

```
struct pcb* tail
```

Definition at line 41 of file R2.h.

The documentation for this struct was generated from the following file:

- MPX-cs450/mpx_core/modules/R2.h

Chapter 4

File Documentation

4.1 MPX-cs450/mpx_core/include/core/asm.h File Reference

```
#include <system.h>
#include <tables.h>
```

4.2 MPX-cs450/mpx_core/include/core/interrupts.h File Reference

Functions

- void init_irq (void)
- void init_pic (void)

4.2.1 Function Documentation

4.2.1.1 init_irq()

```
void init_irq (
            void  )
```

Definition at line 67 of file interrupts.c.

4.2.1.2 init_pic()

```
void init_pic (
            void  )
```

Definition at line 108 of file interrupts.c.

4.3 MPX-cs450/mpx_core/include/core/io.h File Reference

Macros

- #define outb(port, data) asm volatile ("outb %%al,%%dx" : : "a" (data), "d" (port))
- #define inb(port)

4.3.1 Macro Definition Documentation

4.3.1.1 inb

```
#define inb(
              port )
```

Value:

```
({                          \
unsigned char r;                    \
asm volatile ("inb %%dx,%%al": "=a" (r): "d" (port)); \
r;                                  \
})
```

Definition at line 15 of file io.h.

4.3.1.2 outb

```
#define outb(
              port,
              data )  asm volatile ("outb %%al,%%dx" :  :  "a" (data), "d" (port))
```

Definition at line 8 of file io.h.

4.4 MPX-cs450/mpx_core/include/core/serial.h File Reference

Data Structures

- struct history

Macros

- #define COM1 0x3f8
- #define COM2 0x2f8
- #define COM3 0x3e8
- #define COM4 0x2e8

Functions

- int init_serial (int device)
- int serial_println (const char *msg)
- int serial_print (const char *msg)
- int set_serial_out (int device)
- int set_serial_in (int device)
- int * polling (char *buffer, int *count)
- void append (char *s, char c)
- void insertAtEnd (struct history **current, char *buffer)

4.4.1 Macro Definition Documentation

4.4.1.1 COM1

```
#define COM1 0x3f8
```

Definition at line 4 of file serial.h.

4.4.1.2 COM2

```
#define COM2 0x2f8
```

Definition at line 5 of file serial.h.

4.4.1.3 COM3

```
#define COM3 0x3e8
```

Definition at line 6 of file serial.h.

4.4.1.4 COM4

```
#define COM4 0x2e8
```

Definition at line 7 of file serial.h.

4.4.2 Function Documentation

4.4.2.1 append()

```
void append (
            char * s,
            char c )
```

Definition at line 272 of file serial.c.

4.4.2.2 init_serial()

```
int init_serial (
            int device )
```

Definition at line 26 of file serial.c.

4.4.2.3 insertAtEnd()

```
void insertAtEnd (
            struct history ** current,
            char * buffer )
```

Definition at line 281 of file serial.c.

4.4.2.4 polling()

```
int* polling (
            char * buffer,
            int * count )
```

DELETE AND BACKSPACE checks if special = delete and the buffer isn't at the end of the line OR checks if letter = backspace sequence and the char to be deleted isnt null and location is greater than 0

Definition at line 94 of file serial.c.

4.4.2.5 serial_print()

```
int serial_print (
            const char * msg )
```

Definition at line 59 of file serial.c.

4.4.2.6 serial_println()

```
int serial_println (
            const char * msg )
```

Definition at line 44 of file serial.c.

4.4.2.7 set_serial_in()

```
int set_serial_in (
            int device )
```

Definition at line 87 of file serial.c.

4.4.2.8 set_serial_out()

```
int set_serial_out (
            int device )
```

Definition at line 75 of file serial.c.

4.5 MPX-cs450/mpx_core/include/core/tables.h File Reference

```
#include "system.h"
```

Data Structures

- struct idt_entry_struct
- struct idt_struct
- struct gdt_descriptor_struct
- struct gdt_entry_struct

Functions

- struct idt_entry_struct __attribute__ ((packed)) idt_entry
- void idt_set_gate (u8int idx, u32int base, u16int sel, u8int flags)
- void gdt_init_entry (int idx, u32int base, u32int limit, u8int access, u8int flags)
- void init_idt ()
- void init_gdt ()

Variables

- u16int base_low
- u16int sselect
- u8int zero
- u8int flags
- u16int base_high
- u16int limit
- u32int base
- u16int limit_low
- u8int base_mid
- u8int access

4.5.1 Function Documentation

4.5.1.1 __attribute__()

```
struct gdt_entry_struct __attribute__ (
             (packed)  )
```

4.5.1.2 gdt_init_entry()

```
void gdt_init_entry (
             int idx,
             u32int base,
             u32int limit,
             u8int access,
             u8int flags )
```

Definition at line 57 of file tables.c.

4.5.1.3 idt_set_gate()

```
void idt_set_gate (
             u8int idx,
             u32int base,
             u16int sel,
             u8int flags )
```

Definition at line 27 of file tables.c.

4.5.1.4 init_gdt()

```
void init_gdt ( )
```

Definition at line 75 of file tables.c.

4.5.1.5 init_idt()

```
void init_idt ( )
```

Definition at line 43 of file tables.c.

4.5.2 Variable Documentation

4.5.2.1 access

```
u8int access
```

Definition at line 3 of file tables.h.

4.5.2.2 base

```
u32int base
```

Definition at line 1 of file tables.h.

4.5.2.3 base_high

```
u8int base_high
```

Definition at line 4 of file tables.h.

4.5.2.4 base_low

```
u16int base_low
```

Definition at line 0 of file tables.h.

4.5.2.5 base_mid

```
u8int base_mid
```

Definition at line 2 of file tables.h.

4.5.2.6 flags

```
u8int flags
```

Definition at line 3 of file tables.h.

4.5.2.7 limit

```
u16int limit
```

Definition at line 0 of file tables.h.

4.5.2.8 limit_low

```
u16int limit_low
```

Definition at line 0 of file tables.h.

4.5.2.9 sselect

```
u16int sselect
```

Definition at line 1 of file tables.h.

4.5.2.10 zero

```
u8int zero
```

Definition at line 2 of file tables.h.

4.6 MPX-cs450/mpx_core/include/mem/heap.h File Reference

Data Structures

- struct header
- struct footer
- struct index_entry
- struct index_table
- struct heap

Macros

- #define TABLE_SIZE 0x1000
- #define KHEAP_BASE 0xD000000
- #define KHEAP_MIN 0x10000
- #define KHEAP_SIZE 0x1000000

Functions

- u32int _kmalloc (u32int size, int align, u32int *phys_addr)
- u32int kmalloc (u32int size)
- u32int kfree ()
- void init_kheap ()
- u32int alloc (u32int size, heap *hp, int align)
- heap * make_heap (u32int base, u32int max, u32int min)

4.6.1 Macro Definition Documentation

4.6.1.1 KHEAP_BASE

```
#define KHEAP_BASE 0xD000000
```

Definition at line 6 of file heap.h.

4.6.1.2 KHEAP_MIN

```
#define KHEAP_MIN 0x10000
```

Definition at line 7 of file heap.h.

4.6.1.3 KHEAP_SIZE

```
#define KHEAP_SIZE 0x1000000
```

Definition at line 8 of file heap.h.

4.6.1.4 TABLE_SIZE

```
#define TABLE_SIZE 0x1000
```

Definition at line 5 of file heap.h.

4.6.2 Function Documentation

4.6.2.1 _kmalloc()

```
u32int _kmalloc (
             u32int size,
             int align,
             u32int * phys_addr )
```

Definition at line 24 of file heap.c.

4.6.2.2 alloc()

```
u32int alloc (
             u32int size,
             heap * hp,
             int align )
```

Definition at line 57 of file heap.c.

4.6.2.3 init_kheap()

```
void init_kheap ( )
```

4.6.2.4 kfree()

```
u32int kfree ( )
```

4.6.2.5 kmalloc()

```
u32int kmalloc (
            u32int size )
```

Definition at line 52 of file heap.c.

4.6.2.6 make_heap()

```
heap* make_heap (
            u32int base,
            u32int max,
            u32int min )
```

Definition at line 71 of file heap.c.

4.7 MPX-cs450/mpx_core/include/mem/paging.h File Reference

```
#include <system.h>
```

Data Structures

- struct page_entry
- struct page_table
- struct page_dir

Macros

- #define PAGE_SIZE 0x1000

Functions

- void set_bit (u32int addr)
- void clear_bit (u32int addr)
- u32int get_bit (u32int addr)
- u32int first_free ()
- void init_paging ()
- void load_page_dir (page_dir *new_page_dir)
- page_entry * get_page (u32int addr, page_dir *dir, int make_table)
- void new_frame (page_entry *page)

4.7.1 Macro Definition Documentation

4.7.1.1 PAGE_SIZE

```
#define PAGE_SIZE 0x1000
```

Definition at line 6 of file paging.h.

4.7.2 Function Documentation

4.7.2.1 clear_bit()

```
void clear_bit (
            u32int addr )
```

Definition at line 44 of file paging.c.

4.7.2.2 first_free()

```
u32int first_free ( )
```

4.7.2.3 get_bit()

```
u32int get_bit (
            u32int addr )
```

Definition at line 56 of file paging.c.

4.7.2.4 get_page()

```
page_entry* get_page (
            u32int addr,
            page_dir * dir,
            int make_table )
```

Definition at line 85 of file paging.c.

4.7.2.5 init_paging()

```
void init_paging ( )
```

Definition at line 111 of file paging.c.

4.7.2.6 load_page_dir()

```
void load_page_dir (
            page_dir * new_page_dir )
```

Definition at line 158 of file paging.c.

4.7.2.7 new_frame()

```
void new_frame (
            page_entry * page )
```

Definition at line 173 of file paging.c.

4.7.2.8 set_bit()

```
void set_bit (
            u32int addr )
```

Definition at line 32 of file paging.c.

4.8 MPX-cs450/mpx_core/include/string.h File Reference

```
#include <system.h>
```

Functions

- int isspace (const char *c)
- void * memset (void *s, int c, size_t n)
- char * strcpy (char *s1, const char *s2)
- char * strcat (char *s1, const char *s2)
- int strlen (const char *s)
- int strcmp (const char *s1, const char *s2)
- char * strtok (char *s1, const char *s2)
- int atoi (const char *s)

4.8.1 Function Documentation

4.8.1.1 atoi()

```
int atoi (
            const char * s )
```

Definition at line 48 of file string.c.

4.8.1.2 isspace()

```
int isspace (
            const char * c )
```

Definition at line 119 of file string.c.

4.8.1.3 memset()

```
void* memset (
            void * s,
            int c,
            size_t n )
```

Definition at line 137 of file string.c.

4.8.1.4 strcat()

```
char* strcat (
            char * s1,
            const char * s2 )
```

Definition at line 106 of file string.c.

4.8.1.5 strcmp()

```
int strcmp (
            const char * s1,
            const char * s2 )
```

Definition at line 79 of file string.c.

4.8.1.6 strcpy()

```
char* strcpy (
            char * s1,
            const char * s2 )
```

Definition at line 36 of file string.c.

4.8.1.7 strlen()

```
int strlen (
            const char * s )
```

Definition at line 24 of file string.c.

4.8.1.8 strtok()

```
char* strtok (
            char * s1,
            const char * s2 )
```

Definition at line 151 of file string.c.

4.9 MPX-cs450/mpx_core/include/system.h File Reference

Data Structures

- struct date_time

Macros

- #define NULL 0
- #define no_warn(p) if (p) while (1) break
- #define asm __asm__
- #define volatile __volatile__
- #define sti() asm volatile ("sti"::)
- #define cli() asm volatile ("cli"::)
- #define nop() asm volatile ("nop"::)
- #define hlt() asm volatile ("hlt"::)
- #define iret() asm volatile ("iret"::)
- #define GDT_CS_ID 0x01
- #define GDT_DS_ID 0x02

Typedefs

- typedef unsigned int size_t
- typedef unsigned char u8int
- typedef unsigned short u16int
- typedef unsigned long u32int

Functions

- void klogv (const char *msg)
- void kpanic (const char *msg)

4.9.1 Macro Definition Documentation

4.9.1.1 asm

```
#define asm __asm__
```

Definition at line 11 of file system.h.

4.9.1.2 cli

```
#define cli( ) asm volatile ("cli"::)
```

Definition at line 15 of file system.h.

4.9.1.3 GDT_CS_ID

```
#define GDT_CS_ID 0x01
```

Definition at line 20 of file system.h.

4.9.1.4 GDT_DS_ID

```
#define GDT_DS_ID 0x02
```

Definition at line 21 of file system.h.

4.9.1.5 hlt

```
#define hlt( ) asm volatile ("hlt"::)
```

Definition at line 17 of file system.h.

4.9.1.6 iret

```
#define iret( ) asm volatile ("iret"::)
```

Definition at line 18 of file system.h.

4.9.1.7 no_warn

```
#define no_warn(
             p ) if (p) while (1) break
```

Definition at line 7 of file system.h.

4.9.1.8 nop

```
#define nop( ) asm volatile ("nop"::)
```

Definition at line 16 of file system.h.

4.9.1.9 NULL

```
#define NULL 0
```

Definition at line 4 of file system.h.

4.9.1.10 sti

```
#define sti( ) asm volatile ("sti"::)
```

Definition at line 14 of file system.h.

4.9.1.11 volatile

```
#define volatile __volatile__
```

Definition at line 12 of file system.h.

4.9.2 Typedef Documentation

4.9.2.1 size_t

```
typedef unsigned int size_t
```

Definition at line 24 of file system.h.

4.9.2.2 u16int

```
typedef unsigned short u16int
```

Definition at line 26 of file system.h.

4.9.2.3 u32int

```
typedef unsigned long u32int
```

Definition at line 27 of file system.h.

4.9.2.4 u8int

```
typedef unsigned char u8int
```

Definition at line 25 of file system.h.

4.9.3 Function Documentation

4.9.3.1 klogv()

```
void klogv (
            const char * msg )
```

Definition at line 11 of file system.c.

4.9.3.2 kpanic()

```
void kpanic (
            const char * msg )
```

Definition at line 24 of file system.c.

4.10 MPX-cs450/mpx_core/kernel/core/interrupts.c File Reference

```
#include <system.h>
#include <core/io.h>
#include <core/serial.h>
#include <core/tables.h>
#include <core/interrupts.h>
```

Macros

- #define PIC1 0x20
- #define PIC2 0xA0
- #define ICW1 0x11
- #define ICW4 0x01
- #define io_wait() asm volatile ("outb $0x80")

Functions

- void divide_error ()
- void debug ()
- void nmi ()
- void breakpoint ()
- void overflow ()
- void bounds ()
- void invalid_op ()
- void device_not_available ()
- void double_fault ()
- void coprocessor_segment ()
- void invalid_tss ()
- void segment_not_present ()
- void stack_segment ()
- void general_protection ()

- void page_fault ()
- void reserved ()
- void coprocessor ()
- void rtc_isr ()
- void sys_call_isr ()
- void isr0 ()
- void do_isr ()
- void init_irq (void)
- void init_pic (void)
- void do_divide_error ()
- void do_debug ()
- void do_nmi ()
- void do_breakpoint ()
- void do_overflow ()
- void do_bounds ()
- void do_invalid_op ()
- void do_device_not_available ()
- void do_double_fault ()
- void do_coprocessor_segment ()
- void do_invalid_tss ()
- void do_segment_not_present ()
- void do_stack_segment ()
- void do_general_protection ()
- void do_page_fault ()
- void do_reserved ()
- void do_coprocessor ()

Variables

- idt_entry idt_entries [256]

4.10.1 Macro Definition Documentation

4.10.1.1 ICW1

```
#define ICW1 0x11
```

Definition at line 20 of file interrupts.c.

4.10.1.2 ICW4

```
#define ICW4 0x01
```

Definition at line 21 of file interrupts.c.

4.10.1.3 io_wait

```
#define io_wait( ) asm volatile ("outb $0x80")
```

Definition at line 28 of file interrupts.c.

4.10.1.4 PIC1

```
#define PIC1 0x20
```

Definition at line 16 of file interrupts.c.

4.10.1.5 PIC2

```
#define PIC2 0xA0
```

Definition at line 17 of file interrupts.c.

4.10.2 Function Documentation

4.10.2.1 bounds()

```
void bounds ( )
```

4.10.2.2 breakpoint()

```
void breakpoint ( )
```

4.10.2.3 coprocessor()

```
void coprocessor ( )
```

4.10.2.4 coprocessor_segment()

```
void coprocessor_segment ( )
```

4.10.2.5 debug()

```
void debug ( )
```

4.10.2.6 device_not_available()

```
void device_not_available ( )
```

4.10.2.7 divide_error()

```
void divide_error ( )
```

4.10.2.8 do_bounds()

```
void do_bounds ( )
```

Definition at line 151 of file interrupts.c.

4.10.2.9 do_breakpoint()

```
void do_breakpoint ( )
```

Definition at line 143 of file interrupts.c.

4.10.2.10 do_coprocessor()

```
void do_coprocessor ( )
```

Definition at line 195 of file interrupts.c.

4.10.2.11 do_coprocessor_segment()

```
void do_coprocessor_segment ( )
```

Definition at line 167 of file interrupts.c.

4.10.2.12 do_debug()

```
void do_debug ( )
```

Definition at line 135 of file interrupts.c.

4.10.2.13 do_device_not_available()

```
void do_device_not_available ( )
```

Definition at line 159 of file interrupts.c.

4.10.2.14 do_divide_error()

```
void do_divide_error ( )
```

Definition at line 131 of file interrupts.c.

4.10.2.15 do_double_fault()

```
void do_double_fault ( )
```

Definition at line 163 of file interrupts.c.

4.10.2.16 do_general_protection()

```
void do_general_protection ( )
```

Definition at line 183 of file interrupts.c.

4.10.2.17 do_invalid_op()

```
void do_invalid_op ( )
```

Definition at line 155 of file interrupts.c.

4.10.2.18 do_invalid_tss()

```
void do_invalid_tss ( )
```

Definition at line 171 of file interrupts.c.

4.10.2.19 do_isr()

```
void do_isr ( )
```

Definition at line 54 of file interrupts.c.

4.10.2.20 do_nmi()

```
void do_nmi ( )
```

Definition at line 139 of file interrupts.c.

4.10.2.21 do_overflow()

```
void do_overflow ( )
```

Definition at line 147 of file interrupts.c.

4.10.2.22 do_page_fault()

```
void do_page_fault ( )
```

Definition at line 187 of file interrupts.c.

4.10.2.23 do_reserved()

```
void do_reserved ( )
```

Definition at line 191 of file interrupts.c.

4.10.2.24 do_segment_not_present()

```
void do_segment_not_present ( )
```

Definition at line 175 of file interrupts.c.

4.10.2.25 do_stack_segment()

```
void do_stack_segment ( )
```

Definition at line 179 of file interrupts.c.

4.10.2.26 double_fault()

```
void double_fault ( )
```

4.10.2.27 general_protection()

```
void general_protection ( )
```

4.10.2.28 init_irq()

```
void init_irq (
             void  )
```

Definition at line 67 of file interrupts.c.

4.10.2.29 init_pic()

```
void init_pic (
            void  )
```

Definition at line 108 of file interrupts.c.

4.10.2.30 invalid_op()

```
void invalid_op ( )
```

4.10.2.31 invalid_tss()

```
void invalid_tss ( )
```

4.10.2.32 isr0()

```
void isr0 ( )
```

4.10.2.33 nmi()

```
void nmi ( )
```

4.10.2.34 overflow()

```
void overflow ( )
```

4.10.2.35 page_fault()

```
void page_fault ( )
```

4.10.2.36 reserved()

```
void reserved ( )
```

4.10.2.37 rtc_isr()

```
void rtc_isr ( )
```

4.10.2.38 segment_not_present()

```
void segment_not_present ( )
```

4.10.2.39 stack_segment()

```
void stack_segment ( )
```

4.10.2.40 sys_call_isr()

```
void sys_call_isr ( )
```

4.10.3 Variable Documentation

4.10.3.1 idt_entries

```
idt_entry idt_entries[256]  [extern]
```

Definition at line 17 of file tables.c.

4.11 MPX-cs450/mpx_core/kernel/core/kmain.c File Reference

```
#include <stdint.h>
#include <string.h>
#include <system.h>
#include <core/io.h>
#include <core/serial.h>
#include <core/tables.h>
#include <core/interrupts.h>
#include <mem/heap.h>
#include <mem/paging.h>
#include "modules/mpx_supt.h"
#include "modules/comhand.c"
#include "modules/R2.h"
#include "modules/Startup.h"
#include "modules/R5.h"
```

Functions

- void makeComhand ()
- void makeIdle ()
- void kmain (void)

4.11.1 Function Documentation

4.11.1.1 kmain()

```
void kmain (
            void  )
```

Definition at line 31 of file kmain.c.

4.11.1.2 makeComhand()

```
void makeComhand ( )
```

Definition at line 108 of file kmain.c.

4.11.1.3 makeIdle()

```
void makeIdle ( )
```

Definition at line 126 of file kmain.c.

4.12 MPX-cs450/mpx_core/kernel/core/serial.c File Reference

```
#include <stdint.h>
#include <string.h>
#include <core/io.h>
#include <core/serial.h>
#include "modules/mpx_supt.h"
```

Macros

- #define NO_ERROR 0

Functions

- int init_serial (int device)
- int serial_println (const char *msg)
- int serial_print (const char *msg)
- int set_serial_out (int device)
- int set_serial_in (int device)
- int * polling (char *buffer, int *count)
- void append (char *s, char c)
- void insertAtEnd (struct history **current, char *newdata)

Variables

- int serial_port_out = 0
- int serial_port_in = 0
- struct history * current =NULL

4.12.1 Macro Definition Documentation

4.12.1.1 NO_ERROR

```
#define NO_ERROR 0
```

Definition at line 16 of file serial.c.

4.12.2 Function Documentation

4.12.2.1 append()

```
void append (
            char * s,
            char c )
```

Definition at line 272 of file serial.c.

4.12.2.2 init_serial()

```
int init_serial (
            int device )
```

Definition at line 26 of file serial.c.

4.12.2.3 insertAtEnd()

```
void insertAtEnd (
            struct history ** current,
            char * newdata )
```

Definition at line 281 of file serial.c.

4.12.2.4 polling()

```
int* polling (
            char * buffer,
            int * count )
```

DELETE AND BACKSPACE checks if special = delete and the buffer isn't at the end of the line OR checks if letter = backspace sequence and the char to be deleted isnt null and location is greater than 0

Definition at line 94 of file serial.c.

4.12.2.5 serial_print()

```
int serial_print (
            const char * msg )
```

Definition at line 59 of file serial.c.

4.12.2.6 serial_println()

```
int serial_println (
            const char * msg )
```

Definition at line 44 of file serial.c.

4.12.2.7 set_serial_in()

```
int set_serial_in (
            int device )
```

Definition at line 87 of file serial.c.

4.12.2.8 set_serial_out()

```
int set_serial_out (
            int device )
```

Definition at line 75 of file serial.c.

4.12.3 Variable Documentation

4.12.3.1 current

```
struct history* current =NULL
```

Definition at line 93 of file serial.c.

4.12.3.2 serial_port_in

```
int serial_port_in = 0
```

Definition at line 20 of file serial.c.

4.12.3.3 serial_port_out

```
int serial_port_out = 0
```

Definition at line 19 of file serial.c.

4.13 MPX-cs450/mpx_core/kernel/core/system.c File Reference

```
#include <string.h>
#include <system.h>
#include <core/serial.h>
```

Functions

- void klogv (const char *msg)
- void kpanic (const char *msg)

4.13.1 Function Documentation

4.13.1.1 klogv()

```
void klogv (
            const char * msg )
```

Definition at line 11 of file system.c.

4.13.1.2 kpanic()

```
void kpanic (
            const char * msg )
```

Definition at line 24 of file system.c.

4.14 MPX-cs450/mpx_core/kernel/core/tables.c File Reference

```
#include <string.h>
#include <core/tables.h>
```

Functions

- void write_gdt_ptr (u32int, size_t)
- void write_idt_ptr (u32int)
- void idt_set_gate (u8int idx, u32int base, u16int sel, u8int flags)
- void init_idt ()
- void gdt_init_entry (int idx, u32int base, u32int limit, u8int access, u8int flags)
- void init_gdt ()

Variables

- gdt_descriptor gdt_ptr
- gdt_entry gdt_entries [5]
- idt_descriptor idt_ptr
- idt_entry idt_entries [256]

4.14.1 Function Documentation

4.14.1.1 gdt_init_entry()

```
void gdt_init_entry (
            int idx,
            u32int base,
            u32int limit,
            u8int access,
            u8int flags )
```

Definition at line 57 of file tables.c.

4.14.1.2 idt_set_gate()

```
void idt_set_gate (
            u8int idx,
            u32int base,
            u16int sel,
            u8int flags )
```

Definition at line 27 of file tables.c.

4.14.1.3 init_gdt()

```
void init_gdt ( )
```

Definition at line 75 of file tables.c.

4.14.1.4 init_idt()

```
void init_idt ( )
```

Definition at line 43 of file tables.c.

4.14.1.5 write_gdt_ptr()

```
void write_gdt_ptr (
             u32int ,
             size_t  )
```

4.14.1.6 write_idt_ptr()

```
void write_idt_ptr (
             u32int  )
```

4.14.2 Variable Documentation

4.14.2.1 gdt_entries

```
gdt_entry gdt_entries[5]
```

Definition at line 13 of file tables.c.

4.14.2.2 gdt_ptr

```
gdt_descriptor gdt_ptr
```

Definition at line 12 of file tables.c.

4.14.2.3 idt_entries

```
idt_entry idt_entries[256]
```

Definition at line 17 of file tables.c.

4.14.2.4 idt_ptr

```
idt_descriptor idt_ptr
```

Definition at line 16 of file tables.c.

4.15 MPX-cs450/mpx_core/kernel/mem/heap.c File Reference

```
#include <system.h>
#include <string.h>
#include <core/serial.h>
#include <mem/heap.h>
#include <mem/paging.h>
```

Functions

- u32int _kmalloc (u32int size, int page_align, u32int *phys_addr)
- u32int kmalloc (u32int size)
- u32int alloc (u32int size, heap *h, int align)
- heap * make_heap (u32int base, u32int max, u32int min)

Variables

- heap * kheap = 0
- heap * curr_heap = 0
- page_dir * kdir
- void * end
- void _end
- void __end
- u32int phys_alloc_addr = (u32int)&end

4.15.1 Function Documentation

4.15.1.1 _kmalloc()

```
u32int _kmalloc (
            u32int size,
            int page_align,
            u32int * phys_addr )
```

Definition at line 24 of file heap.c.

4.15.1.2 alloc()

```
u32int alloc (
             u32int size,
             heap * h,
             int align )
```

Definition at line 57 of file heap.c.

4.15.1.3 kmalloc()

```
u32int kmalloc (
             u32int size )
```

Definition at line 52 of file heap.c.

4.15.1.4 make_heap()

```
heap* make_heap (
             u32int base,
             u32int max,
             u32int min )
```

Definition at line 71 of file heap.c.

4.15.2 Variable Documentation

4.15.2.1 __end

```
void __end
```

Definition at line 18 of file heap.c.

4.15.2.2 _end

```
void _end
```

Definition at line 18 of file heap.c.

4.15.2.3 curr_heap

```
heap* curr_heap = 0
```

Definition at line 15 of file heap.c.

4.15.2.4 end

```
void* end  [extern]
```

4.15.2.5 kdir

```
page_dir* kdir  [extern]
```

Definition at line 21 of file paging.c.

4.15.2.6 kheap

```
heap* kheap = 0
```

Definition at line 14 of file heap.c.

4.15.2.7 phys_alloc_addr

```
u32int phys_alloc_addr = (u32int)&end
```

Definition at line 22 of file heap.c.

4.16 MPX-cs450/mpx_core/kernel/mem/paging.c File Reference

```
#include <system.h>
#include <string.h>
#include "mem/heap.h"
#include "mem/paging.h"
```

Functions

- void set_bit (u32int addr)
- void clear_bit (u32int addr)
- u32int get_bit (u32int addr)
- u32int find_free ()
- page_entry * get_page (u32int addr, page_dir *dir, int make_table)
- void init_paging ()
- void load_page_dir (page_dir *new_dir)
- void new_frame (page_entry *page)

Variables

- u32int mem_size = 0x4000000
- u32int page_size = 0x1000
- u32int nframes
- u32int * frames
- page_dir * kdir = 0
- page_dir * cdir = 0
- u32int phys_alloc_addr
- heap * kheap

4.16.1 Function Documentation

4.16.1.1 clear_bit()

```
void clear_bit (
            u32int addr )
```

Definition at line 44 of file paging.c.

4.16.1.2 find_free()

```
u32int find_free ( )
```

Definition at line 68 of file paging.c.

4.16.1.3 get_bit()

```
u32int get_bit (
            u32int addr )
```

Definition at line 56 of file paging.c.

4.16.1.4 get_page()

```
page_entry* get_page (
            u32int addr,
            page_dir * dir,
            int make_table )
```

Definition at line 85 of file paging.c.

4.16.1.5 init_paging()

```
void init_paging ( )
```

Definition at line 111 of file paging.c.

4.16.1.6 load_page_dir()

```
void load_page_dir (
            page_dir * new_dir )
```

Definition at line 158 of file paging.c.

4.16.1.7 new_frame()

```
void new_frame (
            page_entry * page )
```

Definition at line 173 of file paging.c.

4.16.1.8 set_bit()

```
void set_bit (
            u32int addr )
```

Definition at line 32 of file paging.c.

4.16.2 Variable Documentation

4.16.2.1 cdir

```
page_dir* cdir = 0
```

Definition at line 22 of file paging.c.

4.16.2.2 frames

```
u32int* frames
```

Definition at line 19 of file paging.c.

4.16.2.3 kdir

```
page_dir* kdir = 0
```

Definition at line 21 of file paging.c.

4.16.2.4 kheap

```
heap* kheap  [extern]
```

Definition at line 14 of file heap.c.

4.16.2.5 mem_size

```
u32int mem_size = 0x4000000
```

Definition at line 15 of file paging.c.

4.16.2.6 nframes

```
u32int nframes
```

Definition at line 18 of file paging.c.

4.16.2.7 page_size

```
u32int page_size = 0x1000
```

Definition at line 16 of file paging.c.

4.16.2.8 phys_alloc_addr

```
u32int phys_alloc_addr  [extern]
```

Definition at line 22 of file heap.c.

4.17 MPX-cs450/mpx_core/lib/string.c File Reference

```
#include <system.h>
#include <string.h>
```

Functions

- int strlen (const char *s)
- char * strcpy (char *s1, const char *s2)
- int atoi (const char *s)
- int strcmp (const char *s1, const char *s2)
- char * strcat (char *s1, const char *s2)
- int isspace (const char *c)
- void * memset (void *s, int c, size_t n)
- char * strtok (char *s1, const char *s2)

4.17.1 Function Documentation

4.17.1.1 atoi()

```
int atoi (
            const char * s )
```

Definition at line 48 of file string.c.

4.17.1.2 isspace()

```
int isspace (
            const char * c )
```

Definition at line 119 of file string.c.

4.17.1.3 memset()

```
void* memset (
            void * s,
            int c,
            size_t n )
```

Definition at line 137 of file string.c.

4.17.1.4 strcat()

```
char* strcat (
            char * s1,
            const char * s2 )
```

Definition at line 106 of file string.c.

4.17.1.5 strcmp()

```
int strcmp (
            const char * s1,
            const char * s2 )
```

Definition at line 79 of file string.c.

4.17.1.6 strcpy()

```
char* strcpy (
            char * s1,
            const char * s2 )
```

Definition at line 36 of file string.c.

4.17.1.7 strlen()

```
int strlen (
            const char * s )
```

Definition at line 24 of file string.c.

4.17.1.8 strtok()

```
char* strtok (
            char * s1,
            const char * s2 )
```

Definition at line 151 of file string.c.

4.18 MPX-cs450/mpx_core/modules/BCDConversions.c File Reference

```
#include <system.h>
#include "BCDConversions.h"
```

Functions

- int BCDtoInt (unsigned char c)
- unsigned char InttoBCD (int i)

4.18.1 Function Documentation

4.18.1.1 BCDtoInt()

```
int BCDtoInt (
            unsigned char c )
```

BCDtoInt converts a value stored as BCD (unsigned char) to an integer Inputs: c - BCD value to be converted Outputs: result - converted value as an integer

Definition at line 9 of file BCDConversions.c.

4.18.1.2 InttoBCD()

```
unsigned char InttoBCD (
            int i )
```

InttoBCD converts a value stored as integer to BCD (unsigned char) Inputs: i - integer value to be converted Outputs: result - converted value as BCD

Definition at line 32 of file BCDConversions.c.

4.19 MPX-cs450/mpx_core/modules/BCDConversions.h File Reference

```
#include <system.h>
```

Functions

- int BCDtoInt (unsigned char c)
- unsigned char InttoBCD (int i)

4.19.1 Function Documentation

4.19.1.1 BCDtoInt()

```
int BCDtoInt (
            unsigned char c )
```

BCDtoInt converts a value stored as BCD (unsigned char) to an integer Inputs: c - BCD value to be converted Outputs: result - converted value as an integer

Definition at line 9 of file BCDConversions.c.

4.19.1.2 InttoBCD()

```
unsigned char InttoBCD (
            int i )
```

InttoBCD converts a value stored as integer to BCD (unsigned char) Inputs: i - integer value to be converted Outputs: result - converted value as BCD

Definition at line 32 of file BCDConversions.c.

4.20 MPX-cs450/mpx_core/modules/comhand.c File Reference

```
#include <stdint.h>
#include <string.h>
#include <system.h>
#include <core/io.h>
#include <core/serial.h>
#include <core/tables.h>
#include <core/interrupts.h>
#include <mem/heap.h>
#include <mem/paging.h>
#include "mpx_supt.h"
#include "DateTime.h"
#include "R1.h"
#include "R2.h"
#include "R3.h"
#include "R4.h"
#include "R5.h"
```

Functions

- void comhand ()

4.20.1 Function Documentation

4.20.1.1 comhand()

```
void comhand ( )
```

Definition at line 19 of file comhand.c.

4.21 MPX-cs450/mpx_core/modules/DateTime.c File Reference

```
#include <system.h>
#include <string.h>
#include "DateTime.h"
#include "mpx_supt.h"
#include <core/io.h>
#include "BCDConversions.h"
#include "itoa.h"
```

Functions

- void getdate (char *args)
- void gettime (char *args)
- void setdate (char *date)
- void settime (char *time)
- int isLeapYear (int year)
- int gettimeseconds ()

4.21.1 Function Documentation

4.21.1.1 getdate()

```
void getdate (
            char * args )
```

getdate extracts and formats the date in mm/dd/yyyy format from RTC registers Inputs: date - character array (buffer) to write date into

Definition at line 14 of file DateTime.c.

4.21.1.2 gettime()

```
void gettime (
            char * args )
```

gettime extracts and formats the time in hh:mm:ss format from RTC registers Inputs: time - character array (buffer) to write time into

Definition at line 128 of file DateTime.c.

4.21.1.3 gettimeseconds()

```
int gettimeseconds ( )
```

gettimeseconds extracts and calculates current time in seconds Outputs: result - current time in seconds

Definition at line 461 of file DateTime.c.

4.21.1.4 isLeapYear()

```
int isLeapYear (
            int year )
```

isLeapYear checks to see if given year is a leap year Inputs: year - year to check as an integer Outputs: 0 if the year is not a leap year, 1 if it is a leap year

Definition at line 431 of file DateTime.c.

4.21.1.5 setdate()

```
void setdate (
            char * date )
```

setdate sets the system date to the given point in time Inputs: date - character array containing date to set system date to in mm/dd/yyyy format

Definition at line 229 of file DateTime.c.

4.21.1.6 settime()

```
void settime (
            char * time )
```

settime sets the system clock to the given time Inputs: date - character array containing date to set system date to in hh/mm/ss format

Definition at line 348 of file DateTime.c.

4.22 MPX-cs450/mpx_core/modules/DateTime.h File Reference

```
#include <system.h>
```

Functions

- void settime (char *time)
- void setdate (char *date)
- void gettime (char *args)
- void getdate (char *args)
- int isLeapYear (int year)
- int gettimeseconds ()

4.22.1 Function Documentation

4.22.1.1 getdate()

```
void getdate (
            char * args )
```

getdate extracts and formats the date in mm/dd/yyyy format from RTC registers Inputs: date - character array (buffer) to write date into

Definition at line 14 of file DateTime.c.

4.22.1.2 gettime()

```
void gettime (
            char * args )
```

gettime extracts and formats the time in hh:mm:ss format from RTC registers Inputs: time - character array (buffer) to write time into

Definition at line 128 of file DateTime.c.

4.22.1.3 gettimeseconds()

```
int gettimeseconds ( )
```

gettimeseconds extracts and calculates current time in seconds Outputs: result - current time in seconds

Definition at line 461 of file DateTime.c.

4.22.1.4 isLeapYear()

```
int isLeapYear (
            int year )
```

isLeapYear checks to see if given year is a leap year Inputs: year - year to check as an integer Outputs: 0 if the year is not a leap year, 1 if it is a leap year

Definition at line 431 of file DateTime.c.

4.22.1.5 setdate()

```
void setdate (
            char * date )
```

setdate sets the system date to the given point in time Inputs: date - character array containing date to set system date to in mm/dd/yyyy format

Definition at line 229 of file DateTime.c.

4.22.1.6 settime()

```
void settime (
            char * time )
```

settime sets the system clock to the given time Inputs: date - character array containing date to set system date to in hh/mm/ss format

Definition at line 348 of file DateTime.c.

4.23 MPX-cs450/mpx_core/modules/itoa.c File Reference

```
#include "itoa.h"
#include <system.h>
#include "mpx_supt.h"
```

Functions

- void itoa (int i, char *buffer)
- char * itoareturn (int i)

4.23.1 Function Documentation

4.23.1.1 itoa()

```
void itoa (
            int i,
            char * buffer )
```

Definition at line 6 of file itoa.c.

4.23.1.2 itoareturn()

```
char* itoareturn (
            int i )
```

Definition at line 66 of file itoa.c.

4.24 MPX-cs450/mpx_core/modules/itoa.h File Reference

```
#include <system.h>
```

Functions

- void itoa (int i, char *buffer)
- char * itoareturn (int i)

4.24.1 Function Documentation

4.24.1.1 itoa()

```
void itoa (
            int i,
            char * buffer )
```

Definition at line 6 of file itoa.c.

4.24.1.2 itoareturn()

```
char* itoareturn (
            int i )
```

Definition at line 66 of file itoa.c.

4.25 MPX-cs450/mpx_core/modules/mpx_supt.c File Reference

```
#include "mpx_supt.h"
#include <mem/heap.h>
#include <string.h>
#include <core/serial.h>
```

Functions

- int sys_req (int op_code, int device_id, char *buffer_ptr, int *count_ptr)
- void mpx_init (int cur_mod)
- void sys_set_malloc (u32int(*func)(u32int))
- void sys_set_free (int(*func)(void *))
- void * sys_alloc_mem (u32int size)
- int sys_free_mem (void *ptr)
- void idle ()
- param * getparam ()

Variables

- param params
- int current_module = -1
- u32int(* student_malloc)(u32int)
- int(* student_free)(void *)

4.25.1 Function Documentation

4.25.1.1 getparam()

```
param* getparam ( )
```

Definition at line 189 of file mpx_supt.c.

4.25.1.2 idle()

```
void idle ( )
```

Definition at line 173 of file mpx_supt.c.

4.25.1.3 mpx_init()

```
void mpx_init (
            int cur_mod )
```

Definition at line 106 of file mpx_supt.c.

4.25.1.4 sys_alloc_mem()

```
void* sys_alloc_mem (
            u32int size )
```

Definition at line 144 of file mpx_supt.c.

4.25.1.5 sys_free_mem()

```
int sys_free_mem (
            void * ptr )
```

Definition at line 158 of file mpx_supt.c.

4.25.1.6 sys_req()

```
int sys_req (
            int op_code,
            int device_id,
            char * buffer_ptr,
            int * count_ptr )
```

Definition at line 49 of file mpx_supt.c.

4.25.1.7 sys_set_free()

```
void sys_set_free (
            int(*)(void *) func )
```

Definition at line 134 of file mpx_supt.c.

4.25.1.8 sys_set_malloc()

```
void sys_set_malloc (
            u32int(*)(u32int) func )
```

Definition at line 124 of file mpx_supt.c.

4.25.2 Variable Documentation

4.25.2.1 current_module

```
int current_module = -1
```

Definition at line 18 of file mpx_supt.c.

4.25.2.2 params

```
param params
```

Definition at line 15 of file mpx_supt.c.

4.25.2.3 student_free

```
int(* student_free) (void *) (
              void *  )
```

Definition at line 28 of file mpx_supt.c.

4.25.2.4 student_malloc

```
u32int(* student_malloc) (u32int) (
              u32int  )
```

Definition at line 24 of file mpx_supt.c.

4.26 MPX-cs450/mpx_core/modules/mpx_supt.h File Reference

```
#include <system.h>
```

Data Structures

- struct param

Macros

- #define EXIT 0
- #define IDLE 1
- #define READ 2
- #define WRITE 3
- #define INVALID_OPERATION 4
- #define TRUE 1
- #define FALSE 0
- #define MODULE_R1 0
- #define MODULE_R2 1
- #define MODULE_R3 2
- #define MODULE_R4 4
- #define MODULE_R5 8
- #define MODULE_F 9
- #define IO_MODULE 10
- #define MEM_MODULE 11
- #define INVALID_BUFFER 1000
- #define INVALID_COUNT 2000
- #define DEFAULT_DEVICE 111
- #define COM_PORT 222

Functions

- int sys_req (int op_code, int device_id, char ∗buffer_ptr, int ∗count_ptr)
- void mpx_init (int cur_mod)
- void sys_set_malloc (u32int(∗func)(u32int))
- void sys_set_free (int(∗func)(void ∗))
- void ∗ sys_alloc_mem (u32int size)
- int sys_free_mem (void ∗ptr)
- void idle ()
- param ∗ getparam ()

4.26.1 Macro Definition Documentation

4.26.1.1 COM_PORT

```
#define COM_PORT 222
```

Definition at line 29 of file mpx_supt.h.

4.26.1.2 DEFAULT_DEVICE

```
#define DEFAULT_DEVICE 111
```

Definition at line 28 of file mpx_supt.h.

4.26.1.3 EXIT

```
#define EXIT 0
```

Definition at line 6 of file mpx_supt.h.

4.26.1.4 FALSE

```
#define FALSE 0
```

Definition at line 13 of file mpx_supt.h.

4.26.1.5 IDLE

```
#define IDLE 1
```

Definition at line 7 of file mpx_supt.h.

4.26.1.6 INVALID_BUFFER

```
#define INVALID_BUFFER 1000
```

Definition at line 25 of file mpx_supt.h.

4.26.1.7 INVALID_COUNT

```
#define INVALID_COUNT 2000
```

Definition at line 26 of file mpx_supt.h.

4.26.1.8 INVALID_OPERATION

```
#define INVALID_OPERATION 4
```

Definition at line 10 of file mpx_supt.h.

4.26.1.9 IO_MODULE

```
#define IO_MODULE 10
```

Definition at line 21 of file mpx_supt.h.

4.26.1.10 MEM_MODULE

```
#define MEM_MODULE 11
```

Definition at line 22 of file mpx_supt.h.

4.26.1.11 MODULE_F

```
#define MODULE_F 9
```

Definition at line 20 of file mpx_supt.h.

4.26.1.12 MODULE_R1

```
#define MODULE_R1 0
```

Definition at line 15 of file mpx_supt.h.

4.26.1.13 MODULE_R2

```
#define MODULE_R2 1
```

Definition at line 16 of file mpx_supt.h.

4.26.1.14 MODULE_R3

```
#define MODULE_R3 2
```

Definition at line 17 of file mpx_supt.h.

4.26.1.15 MODULE_R4

```
#define MODULE_R4 4
```

Definition at line 18 of file mpx_supt.h.

4.26.1.16 MODULE_R5

```
#define MODULE_R5 8
```

Definition at line 19 of file mpx_supt.h.

4.26.1.17 READ

```
#define READ 2
```

Definition at line 8 of file mpx_supt.h.

4.26.1.18 TRUE

```
#define TRUE 1
```

Definition at line 12 of file mpx_supt.h.

4.26.1.19 WRITE

```
#define WRITE 3
```

Definition at line 9 of file mpx_supt.h.

4.26.2 Function Documentation

4.26.2.1 getparam()

```
param* getparam ( )
```

Definition at line 189 of file mpx_supt.c.

4.26.2.2 idle()

```
void idle ( )
```

Definition at line 173 of file mpx_supt.c.

4.26.2.3 mpx_init()

```
void mpx_init (
            int cur_mod )
```

Definition at line 106 of file mpx_supt.c.

4.26.2.4 sys_alloc_mem()

```
void* sys_alloc_mem (
            u32int size )
```

Definition at line 144 of file mpx_supt.c.

4.26.2.5 sys_free_mem()

```
int sys_free_mem (
            void * ptr )
```

Definition at line 158 of file mpx_supt.c.

4.26.2.6 sys_req()

```
int sys_req (
            int op_code,
            int device_id,
            char * buffer_ptr,
            int * count_ptr )
```

Definition at line 49 of file mpx_supt.c.

4.26.2.7 sys_set_free()

```
void sys_set_free (
            int(*)(void *) func )
```

Definition at line 134 of file mpx_supt.c.

4.26.2.8 sys_set_malloc()

```
void sys_set_malloc (
            u32int(*)(u32int) func )
```

Definition at line 124 of file mpx_supt.c.

4.27 MPX-cs450/mpx_core/modules/procsr3.c File Reference

```
#include "../include/system.h"
#include "../include/core/serial.h"
#include "mpx_supt.h"
#include "procsr3.h"
```

Macros

- #define RC_1 1
- #define RC_2 2
- #define RC_3 3
- #define RC_4 4
- #define RC_5 5

Functions

- void proc1 ()
- void proc2 ()
- void proc3 ()
- void proc4 ()
- void proc5 ()

Variables

- char * msg1 = "proc1 dispatched"
- char * msg2 = "proc2 dispatched"
- char * msg3 = "proc3 dispatched"
- char * msg4 = "proc4 dispatched"
- char * msg5 = "proc5 dispatched"
- int msgSize = 17
- char * er1 = "proc1 ran after it was terminated"
- char * er2 = "proc2 ran after it was terminated"
- char * er3 = "proc3 ran after it was terminated"
- char * er4 = "proc4 ran after it was terminated"
- char * er5 = "proc5 ran after it was terminated"
- int erSize = 34

4.27.1 Macro Definition Documentation

4.27.1.1 RC_1

```
#define RC_1 1
```

Definition at line 7 of file procsr3.c.

4.27.1.2 RC_2

```
#define RC_2 2
```

Definition at line 8 of file procsr3.c.

4.27.1.3 RC_3

```
#define RC_3 3
```

Definition at line 9 of file procsr3.c.

4.27.1.4 RC_4

```
#define RC_4 4
```

Definition at line 10 of file procsr3.c.

4.27.1.5 RC_5

```
#define RC_5 5
```

Definition at line 11 of file procsr3.c.

4.27.2 Function Documentation

4.27.2.1 proc1()

```
void proc1 ( )
```

Definition at line 27 of file procsr3.c.

4.27.2.2 proc2()

```
void proc2 ( )
```

Definition at line 44 of file procsr3.c.

4.27.2.3 proc3()

```
void proc3 ( )
```

Definition at line 59 of file procsr3.c.

4.27.2.4 proc4()

```
void proc4 ( )
```

Definition at line 74 of file procsr3.c.

4.27.2.5 proc5()

```
void proc5 ( )
```

Definition at line 89 of file procsr3.c.

4.27.3 Variable Documentation

4.27.3.1 er1

```
char* er1 = "proc1 ran after it was terminated"
```

Definition at line 20 of file procsr3.c.

4.27.3.2 er2

```
char* er2 = "proc2 ran after it was terminated"
```

Definition at line 21 of file procsr3.c.

4.27.3.3 er3

```
char* er3 = "proc3 ran after it was terminated"
```

Definition at line 22 of file procsr3.c.

4.27.3.4 er4

```
char* er4 = "proc4 ran after it was terminated"
```

Definition at line 23 of file procsr3.c.

4.27.3.5 er5

```
char* er5 = "proc5 ran after it was terminated"
```

Definition at line 24 of file procsr3.c.

4.27.3.6 erSize

```
int erSize = 34
```

Definition at line 25 of file procsr3.c.

4.27.3.7 msg1

```
char* msg1 = "proc1 dispatched"
```

Definition at line 13 of file procsr3.c.

4.27.3.8 msg2

```
char* msg2 = "proc2 dispatched"
```

Definition at line 14 of file procsr3.c.

4.27.3.9 msg3

```
char* msg3 = "proc3 dispatched"
```

Definition at line 15 of file procsr3.c.

4.27.3.10 msg4

```
char* msg4 = "proc4 dispatched"
```

Definition at line 16 of file procsr3.c.

4.27.3.11 msg5

```
char* msg5 = "proc5 dispatched"
```

Definition at line 17 of file procsr3.c.

4.27.3.12 msgSize

```
int msgSize = 17
```

Definition at line 18 of file procsr3.c.

4.28 MPX-cs450/mpx_core/modules/procsr3.h File Reference

```
#include <system.h>
```

Functions

- void proc1 ()
- void proc2 ()
- void proc3 ()
- void proc4 ()
- void proc5 ()

4.28.1 Function Documentation

4.28.1.1 proc1()

```
void proc1 ( )
```

Definition at line 27 of file procsr3.c.

4.28.1.2 proc2()

```
void proc2 ( )
```

Definition at line 44 of file procsr3.c.

4.28.1.3 proc3()

```
void proc3 ( )
```

Definition at line 59 of file procsr3.c.

4.28.1.4 proc4()

```
void proc4 ( )
```

Definition at line 74 of file procsr3.c.

4.28.1.5 proc5()

```
void proc5 ( )
```

Definition at line 89 of file procsr3.c.

4.29 MPX-cs450/mpx_core/modules/R1.c File Reference

```
#include "R1.h"
```

Functions

- char * version ()
- void help_help ()
- void help_version ()
- void help_getdate ()
- void help_setdate ()
- void help_gettime ()
- void help_settime ()
- void help_shutdown ()
- void help_suspendpcb ()
- void help_resumepcb ()
- void help_setpcbpriority ()
- void help_showpcb ()
- void help_showallprocesses ()
- void help_showreadyprocesses ()
- void help_showblockedprocesses ()
- void help_deletePCB ()
- void help_loadr3 ()
- void help_color ()
- void help_yield ()
- void help_inf ()
- void help_alarm ()
- void help (char *instr)
- void color (char *instr)

4.29.1 Function Documentation

4.29.1.1 color()

```
void color (
            char * instr )
```

Definition at line 277 of file R1.c.

4.29.1.2 help()

```
void help (
            char * instr )
```

Definition at line 152 of file R1.c.

4.29.1.3 help_alarm()

```
void help_alarm ( )
```

Definition at line 145 of file R1.c.

4.29.1.4 help_color()

```
void help_color ( )
```

Definition at line 123 of file R1.c.

4.29.1.5 help_deletePCB()

```
void help_deletePCB ( )
```

Definition at line 109 of file R1.c.

4.29.1.6 help_getdate()

```
void help_getdate ( )
```

Definition at line 26 of file R1.c.

4.29.1.7 help_gettime()

```
void help_gettime ( )
```

Definition at line 39 of file R1.c.

4.29.1.8 help_help()

```
void help_help ( )
```

Definition at line 11 of file R1.c.

4.29.1.9 help_inf()

```
void help_inf ( )
```

Definition at line 138 of file R1.c.

4.29.1.10 help_loadr3()

```
void help_loadr3 ( )
```

Definition at line 116 of file R1.c.

4.29.1.11 help_resumepcb()

```
void help_resumepcb ( )
```

Definition at line 67 of file R1.c.

4.29.1.12 help_setdate()

```
void help_setdate ( )
```

Definition at line 32 of file R1.c.

4.29.1.13 help_setpcbpriority()

```
void help_setpcbpriority ( )
```

Definition at line 74 of file R1.c.

4.29.1.14 help_settime()

```
void help_settime ( )
```

Definition at line 46 of file R1.c.

4.29.1.15 help_showallprocesses()

```
void help_showallprocesses ( )
```

Definition at line 88 of file R1.c.

4.29.1.16 help_showblockedprocesses()

```
void help_showblockedprocesses ( )
```

Definition at line 102 of file R1.c.

4.29.1.17 help_showpcb()

```
void help_showpcb ( )
```

Definition at line 81 of file R1.c.

4.29.1.18 help_showreadyprocesses()

```
void help_showreadyprocesses ( )
```

Definition at line 95 of file R1.c.

4.29.1.19 help_shutdown()

```
void help_shutdown ( )
```

Definition at line 53 of file R1.c.

4.29.1.20 help_suspendpcb()

```
void help_suspendpcb ( )
```

Definition at line 60 of file R1.c.

4.29.1.21 help_version()

```
void help_version ( )
```

Definition at line 18 of file R1.c.

4.29.1.22 help_yield()

```
void help_yield ( )
```

Definition at line 130 of file R1.c.

4.29.1.23 version()

```
char* version ( )
```

Definition at line 3 of file R1.c.

4.30 MPX-cs450/mpx_core/modules/R1.h File Reference

```
#include <stdint.h>
#include <string.h>
#include <system.h>
#include <core/io.h>
#include <core/serial.h>
#include <core/tables.h>
#include <core/interrupts.h>
#include <mem/heap.h>
#include <mem/paging.h>
#include "mpx_supt.h"
```

Functions

- char * version ()
- void help (char *instr)
- void color (char *instr)

4.30.1 Function Documentation

4.30.1.1 color()

```
void color (
            char * instr )
```

Definition at line 277 of file R1.c.

4.30.1.2 help()

```
void help (
            char * instr )
```

Definition at line 152 of file R1.c.

4.30.1.3 version()

```
char* version ( )
```

Definition at line 3 of file R1.c.

4.31 MPX-cs450/mpx_core/modules/R2.c File Reference

```
#include "R2.h"
```

Functions

- void emptyqueues ()
- int FreePCB (struct pcb *pointer)
- void insert (struct pcb *pcbPtr, struct Queue *q)
- void insertblocked (struct pcb *pcbPtr, struct Queue *q)
- void * AllocatePCB ()
- void * SetupPCB (char *processname, int processclass, int processpriority)
- struct pcb * FindPCB (char *name)
- void InsertPCB (struct pcb *pointer)
- void RemovePCB (struct pcb *pointer)
- void createPCB (char *processName, int processClass, int processPriority)
- void blockPCB (char *processName)
- void unblockPCB (char *processName)
- void deletePCB (char *name)
- void suspendPCB (char *name)
- void resumePCB (char *name)
- void setPCBPriority (char *name, int priority)
- void showPCB (char *processname)
- void showReady ()
- void showBlocked ()
- void showAll ()
- u32int sys_call (struct context *registers)

Variables

- int ready = 0
- int blocked = 1
- int running = 2
- struct pcb * cop
- struct context * saved_regs
- struct Queue readyQ = {0, NULL, NULL}
- struct Queue * y = &readyQ
- struct Queue blockedQ = {0, NULL, NULL}
- struct Queue * z = &blockedQ
- int buffersize = 100

4.31.1 Function Documentation

4.31.1.1 AllocatePCB()

```
void* AllocatePCB ( )
```

Definition at line 122 of file R2.c.

4.31.1.2 blockPCB()

```
void blockPCB (
            char * processName )
```

Definition at line 326 of file R2.c.

4.31.1.3 createPCB()

```
void createPCB (
            char * processName,
            int processClass,
            int processPriority )
```

Definition at line 289 of file R2.c.

4.31.1.4 deletePCB()

```
void deletePCB (
            char * name )
```

Definition at line 401 of file R2.c.

4.31.1.5 emptyqueues()

```
void emptyqueues ( )
```

Definition at line 20 of file R2.c.

4.31.1.6 FindPCB()

```
struct pcb* FindPCB (
            char * name )
```

Definition at line 185 of file R2.c.

4.31.1.7 FreePCB()

```
int FreePCB (
            struct pcb * pointer )
```

Definition at line 35 of file R2.c.

4.31.1.8 insert()

```
void insert (
            struct pcb * pcbPtr,
            struct Queue * q )
```

insert generic insert function to be used by InsertPCB function, inserts a given PCB into a given Queue. Sorted in H->L priority, FIFO order Inputs: pcbPtr - pointer to PCB that is to be inserted q - pointer to Queue that PCB is to be inserted into

Definition at line 50 of file R2.c.

4.31.1.9 insertblocked()

```
void insertblocked (
            struct pcb * pcbPtr,
            struct Queue * q )
```

Definition at line 101 of file R2.c.

4.31.1.10 InsertPCB()

```
void InsertPCB (
            struct pcb * pointer )
```

InsertPCB inserts a given PCB into the corresponding queue Inputs: pointer - pointer to the PCB to be inserted

Definition at line 214 of file R2.c.

4.31.1.11 RemovePCB()

```
void RemovePCB (
            struct pcb * pointer )
```

Definition at line 235 of file R2.c.

4.31.1.12 resumePCB()

```
void resumePCB (
            char * name )
```

Definition at line 462 of file R2.c.

4.31.1.13 setPCBPriority()

```
void setPCBPriority (
            char * name,
            int priority )
```

Definition at line 494 of file R2.c.

4.31.1.14 SetupPCB()

```
void* SetupPCB (
            char * processname,
            int processclass,
            int processpriority )
```

Definition at line 135 of file R2.c.

4.31.1.15 showAll()

```
void showAll ( )
```

Definition at line 606 of file R2.c.

4.31.1.16 showBlocked()

```
void showBlocked ( )
```

Definition at line 590 of file R2.c.

4.31.1.17 showPCB()

```
void showPCB (
            char * processname )
```

Definition at line 525 of file R2.c.

4.31.1.18 showReady()

```
void showReady ( )
```

Definition at line 574 of file R2.c.

4.31.1.19 suspendPCB()

```
void suspendPCB (
            char * name )
```

Definition at line 431 of file R2.c.

4.31.1.20 sys_call()

```
u32int sys_call (
            struct context * registers )
```

Definition at line 617 of file R2.c.

4.31.1.21 unblockPCB()

```
void unblockPCB (
            char * processName )
```

Definition at line 369 of file R2.c.

4.31.2 Variable Documentation

4.31.2.1 blocked

```
int blocked = 1
```

Definition at line 7 of file R2.c.

4.31.2.2 blockedQ

```
struct Queue blockedQ = {0, NULL, NULL}
```

Definition at line 15 of file R2.c.

4.31.2.3 buffersize

```
int buffersize = 100
```

Definition at line 18 of file R2.c.

4.31.2.4 cop

```
struct pcb* cop
```

Definition at line 11 of file R2.c.

4.31.2.5 ready

```
int ready = 0
```

Definition at line 6 of file R2.c.

4.31.2.6 readyQ

```
struct Queue readyQ = {0, NULL, NULL}
```

Definition at line 12 of file R2.c.

4.31.2.7 running

```
int running = 2
```

Definition at line 8 of file R2.c.

4.31.2.8 saved_regs

```
struct context* saved_regs
```

Definition at line 12 of file R2.c.

4.31.2.9 y

```
struct Queue* y = &readyQ
```

Definition at line 15 of file R2.c.

4.31.2.10 z

```
struct Queue* z = &blockedQ
```

Definition at line 17 of file R2.c.

4.32 MPX-cs450/mpx_core/modules/R2.h File Reference

```
#include <stdint.h>
#include <string.h>
#include <system.h>
#include <core/io.h>
#include <core/serial.h>
#include <core/tables.h>
#include <core/interrupts.h>
#include <mem/heap.h>
#include <mem/paging.h>
#include "mpx_supt.h"
#include "itoa.h"
```

Data Structures

- struct pcb
- struct Queue
- struct context

Functions

- void emptyqueues ()
- int FreePCB (struct pcb *pointer)
- int isEmpty (struct Queue *q)
- void insert (struct pcb *pcbPtr, struct Queue *q)
- void insertblocked (struct pcb *pcbPtr, struct Queue *q)
- void * AllocatePCB ()
- void * SetupPCB (char *processname, int processclass, int processpriority)
- struct pcb * FindPCB (char *name)
- void InsertPCB (struct pcb *pointer)
- void RemovePCB (struct pcb *pointer)
- void createPCB (char *processName, int processClass, int processPriority)
- void blockPCB (char *processName)
- void unblockPCB (char *processName)
- void deletePCB (char *name)
- void suspendPCB (char *name)
- void resumePCB (char *name)
- void setPCBPriority (char *name, int priority)
- void showPCB (char *processname)
- void showReady ()
- void showBlocked ()
- void showAll ()
- u32int sys_call (struct context *registers)

4.32.1 Function Documentation

4.32.1.1 AllocatePCB()

```
void* AllocatePCB ( )
```

Definition at line 122 of file R2.c.

4.32.1.2 blockPCB()

```
void blockPCB (
            char * processName )
```

Definition at line 326 of file R2.c.

4.32.1.3 createPCB()

```
void createPCB (
            char * processName,
            int processClass,
            int processPriority )
```

Definition at line 289 of file R2.c.

4.32.1.4 deletePCB()

```
void deletePCB (
            char * name )
```

Definition at line 401 of file R2.c.

4.32.1.5 emptyqueues()

```
void emptyqueues ( )
```

Definition at line 20 of file R2.c.

4.32.1.6 FindPCB()

```
struct pcb* FindPCB (
            char * name )
```

Definition at line 185 of file R2.c.

4.32.1.7 FreePCB()

```
int FreePCB (
            struct pcb * pointer )
```

Definition at line 35 of file R2.c.

4.32.1.8 insert()

```
void insert (
            struct pcb * pcbPtr,
            struct Queue * q )
```

insert generic insert function to be used by InsertPCB function, inserts a given PCB into a given Queue. Sorted in H->L priority, FIFO order Inputs: pcbPtr - pointer to PCB that is to be inserted q - pointer to Queue that PCB is to be inserted into

Definition at line 50 of file R2.c.

4.32.1.9 insertblocked()

```
void insertblocked (
            struct pcb * pcbPtr,
            struct Queue * q )
```

Definition at line 101 of file R2.c.

4.32.1.10 InsertPCB()

```
void InsertPCB (
            struct pcb * pointer )
```

InsertPCB inserts a given PCB into the corresponding queue Inputs: pointer - pointer to the PCB to be inserted

Definition at line 214 of file R2.c.

4.32.1.11 isEmpty()

```
int isEmpty (
            struct Queue * q )
```

4.32.1.12 RemovePCB()

```
void RemovePCB (
            struct pcb * pointer )
```

Definition at line 235 of file R2.c.

4.32.1.13 resumePCB()

```
void resumePCB (
            char * name )
```

Definition at line 462 of file R2.c.

4.32.1.14 setPCBPriority()

```
void setPCBPriority (
            char * name,
            int priority )
```

Definition at line 494 of file R2.c.

4.32.1.15 SetupPCB()

```
void* SetupPCB (
            char * processname,
            int processclass,
            int processpriority )
```

Definition at line 135 of file R2.c.

4.32.1.16 showAll()

```
void showAll ( )
```

Definition at line 606 of file R2.c.

4.32.1.17 showBlocked()

```
void showBlocked ( )
```

Definition at line 590 of file R2.c.

4.32.1.18 showPCB()

```
void showPCB (
            char * processname )
```

Definition at line 525 of file R2.c.

4.32.1.19 showReady()

```
void showReady ( )
```

Definition at line 574 of file R2.c.

4.32.1.20 suspendPCB()

```
void suspendPCB (
            char * name )
```

Definition at line 431 of file R2.c.

4.32.1.21 sys_call()

```
u32int sys_call (
            struct context * registers )
```

Definition at line 617 of file R2.c.

4.32.1.22 unblockPCB()

```
void unblockPCB (
            char * processName )
```

Definition at line 369 of file R2.c.

4.33 MPX-cs450/mpx_core/modules/R3.c File Reference

```
#include <system.h>
#include <string.h>
#include <core/io.h>
#include <core/serial.h>
#include <core/tables.h>
#include <core/interrupts.h>
#include <mem/heap.h>
#include <mem/paging.h>
#include "mpx_supt.h"
#include "R2.h"
#include "R3.h"
#include "procsr3.h"
```

Functions

- void yield ()
- void * loadr3 ()

4.33.1 Function Documentation

4.33.1.1 loadr3()

```
void* loadr3 ( )
```

Definition at line 20 of file R3.c.

4.33.1.2 yield()

```
void yield ( )
```

Definition at line 16 of file R3.c.

4.34 MPX-cs450/mpx_core/modules/R3.h File Reference

```
#include <system.h>
```

Functions

- void yield ()
- void * loadr3 ()

4.34.1 Function Documentation

4.34.1.1 loadr3()

```
void* loadr3 ( )
```

Definition at line 20 of file R3.c.

4.34.1.2 yield()

```
void yield ( )
```

Definition at line 16 of file R3.c.

4.35 MPX-cs450/mpx_core/modules/R4.c File Reference

```
#include <core/io.h>
#include <core/serial.h>
#include <core/tables.h>
#include <core/interrupts.h>
#include <mem/heap.h>
#include <mem/paging.h>
#include "mpx_supt.h"
#include "DateTime.h"
#include "R2.h"
#include "itoa.h"
#include "R4.h"
```

Functions

- void printtime ()
- int convertsec (char *msg)
- void alarm ()
- void createAlarm (char *message, char *time)
- void checkAlarm ()
- void createInfinite ()
- void infinite ()
- void insertalarm (struct alarm *alPtr)
- void removealarm (struct alarm *pointer)

Variables

- struct alList list = {0, NULL, NULL}
- struct alList * alarmlist = &list

4.35.1 Function Documentation

4.35.1.1 alarm()

```
void alarm ( )
```

Definition at line 43 of file R4.c.

4.35.1.2 checkAlarm()

```
void checkAlarm ( )
```

Definition at line 69 of file R4.c.

4.35.1.3 convertsec()

```
int convertsec (
            char * msg )
```

Definition at line 29 of file R4.c.

4.35.1.4 createAlarm()

```
void createAlarm (
            char * message,
            char * time )
```

Definition at line 61 of file R4.c.

4.35.1.5 createInfinite()

```
void createInfinite ( )
```

Definition at line 107 of file R4.c.

4.35.1.6 infinite()

```
void infinite ( )
```

Definition at line 127 of file R4.c.

4.35.1.7 insertalarm()

```
void insertalarm (
            struct alarm * alPtr )
```

Definition at line 142 of file R4.c.

4.35.1.8 printtime()

```
void printtime ( )
```

Definition at line 19 of file R4.c.

4.35.1.9 removealarm()

```
void removealarm (
            struct alarm * pointer )
```

Definition at line 158 of file R4.c.

4.35.2 Variable Documentation

4.35.2.1 alarmlist

```
struct alList* alarmlist = &list
```

Definition at line 16 of file R4.c.

4.35.2.2 list

```
struct alList list = {0, NULL, NULL}
```

Definition at line 1 of file R4.c.

4.36 MPX-cs450/mpx_core/modules/R4.h File Reference

```
#include <stdint.h>
#include <string.h>
#include <system.h>
#include <core/io.h>
#include <core/serial.h>
#include <core/tables.h>
#include <core/interrupts.h>
#include <mem/heap.h>
#include <mem/paging.h>
#include "mpx_supt.h"
#include "DateTime.h"
#include "R2.h"
#include "BCDConversions.h"
#include "itoa.h"
```

Data Structures

- struct alarm
- struct alList

Functions

- void alarm ()
- void printtime ()
- void checkAlarm ()
- void createAlarm (char *message, char *time)
- void infinite ()
- void createInfinite ()
- int convertsec (char *time)
- void insertalarm (struct alarm *al)
- void removealarm (struct alarm *pointer)

4.36.1 Function Documentation

4.36.1.1 alarm()

```
void alarm ( )
```

Definition at line 43 of file R4.c.

4.36.1.2 checkAlarm()

```
void checkAlarm ( )
```

Definition at line 69 of file R4.c.

4.36.1.3 convertsec()

```
int convertsec (
            char * time )
```

Definition at line 29 of file R4.c.

4.36.1.4 createAlarm()

```
void createAlarm (
            char * message,
            char * time )
```

Definition at line 61 of file R4.c.

4.36.1.5 createInfinite()

```
void createInfinite ( )
```

Definition at line 107 of file R4.c.

4.36.1.6 infinite()

```
void infinite ( )
```

Definition at line 127 of file R4.c.

4.36.1.7 insertalarm()

```
void insertalarm (
            struct alarm * al )
```

Definition at line 142 of file R4.c.

4.36.1.8 printtime()

```
void printtime ( )
```

Definition at line 19 of file R4.c.

4.36.1.9 removealarm()

```
void removealarm (
            struct alarm * pointer )
```

Definition at line 158 of file R4.c.

4.37 MPX-cs450/mpx_core/modules/R5.c File Reference

```
#include "R5.h"
```

Functions

- int isEmptyR5 ()
- void * initHeap (int size)
- void * allocateMem (int size)
- void freeMem (void *x)
- void showfree ()
- void showallocated ()
- void printaddr (int address)
- struct CMCB * findCMCB (void *x)

Variables

- struct heap * heapStart
- struct allocatedblocks * allocblocks
- struct freeblocks * freeblocks

4.37.1 Function Documentation

4.37.1.1 allocateMem()

```
void* allocateMem (
            int size )
```

Definition at line 55 of file R5.c.

4.37.1.2 findCMCB()

```
struct CMCB* findCMCB (
            void * x )
```

Definition at line 400 of file R5.c.

4.37.1.3 freeMem()

```
void freeMem (
            void * x )
```

Definition at line 187 of file R5.c.

4.37.1.4 initHeap()

```
void* initHeap (
            int size )
```

Definition at line 22 of file R5.c.

4.37.1.5 isEmptyR5()

```
int isEmptyR5 ( )
```

Definition at line 8 of file R5.c.

4.37.1.6 printaddr()

```
void printaddr (
            int address )
```

Definition at line 391 of file R5.c.

4.37.1.7 showallocated()

```
void showallocated ( )
```

Definition at line 363 of file R5.c.

4.37.1.8 showfree()

```
void showfree ( )
```

Definition at line 328 of file R5.c.

4.37.2 Variable Documentation

4.37.2.1 allocblocks

```
struct allocatedblocks* allocblocks
```

Definition at line 4 of file R5.c.

4.37.2.2 freeblocks

```
struct freeblocks* freeblocks
```

Definition at line 5 of file R5.c.

4.37.2.3 heapStart

```
struct heap* heapStart
```

Definition at line 3 of file R5.c.

4.38 MPX-cs450/mpx_core/modules/R5.h File Reference

```
#include <stdint.h>
#include <string.h>
#include <system.h>
#include <core/io.h>
#include <core/serial.h>
#include <core/tables.h>
#include <core/interrupts.h>
#include <mem/heap.h>
#include <mem/paging.h>
#include "mpx_supt.h"
#include "R2.h"
```

Data Structures

- struct heap
- struct CMCB
- struct allocatedblocks
- struct freeblocks

Functions

- int isEmptyR5 ()
- void * initHeap (int size)
- void * allocateMem (int size)
- void freeMem (void *x)
- void showfree ()
- void showallocated ()
- void printaddr (int address)
- struct CMCB * findCMCB (void *x)

4.38.1 Function Documentation

4.38.1.1 allocateMem()

```
void* allocateMem (
            int size )
```

Definition at line 55 of file R5.c.

4.38.1.2 findCMCB()

```
struct CMCB* findCMCB (
            void * x )
```

Definition at line 400 of file R5.c.

4.38.1.3 freeMem()

```
void freeMem (
            void * x )
```

Definition at line 187 of file R5.c.

4.38.1.4 initHeap()

```
void* initHeap (
            int size )
```

Definition at line 22 of file R5.c.

4.38.1.5 isEmptyR5()

```
int isEmptyR5 ( )
```

Definition at line 8 of file R5.c.

4.38.1.6 printaddr()

```
void printaddr (
            int address )
```

Definition at line 391 of file R5.c.

4.38.1.7 showallocated()

```
void showallocated ( )
```

Definition at line 363 of file R5.c.

4.38.1.8 showfree()

```
void showfree ( )
```

Definition at line 328 of file R5.c.

4.39 MPX-cs450/mpx_core/modules/Startup.c File Reference

```
#include "Startup.h"
```

Functions

- void startup ()

4.39.1 Function Documentation

4.39.1.1 startup()

```
void startup ( )
```

Definition at line 3 of file Startup.c.

4.40 MPX-cs450/mpx_core/modules/Startup.h File Reference

```
#include <stdint.h>
#include <string.h>
#include <system.h>
#include <core/io.h>
#include <core/serial.h>
#include <core/tables.h>
#include <core/interrupts.h>
#include <mem/heap.h>
#include <mem/paging.h>
#include "mpx_supt.h"
```

Functions

- void startup ()

4.40.1 Function Documentation

4.40.1.1 startup()

```
void startup ( )
```

Definition at line 3 of file Startup.c.

Index