Runtime Terror

6.0

Generated by Doxygen 1.9.1

1 Class Index 1

1.1 Class List . . . 1

2 File Index 3

2.1 File List . . . 3

3 Class Documentation 5

3.1 Alarm Struct Reference . . . 5
3.2 CMCB Struct Reference . . . 5
3.3 context Struct Reference . . . 6
3.4 date_time Struct Reference . . . 6
3.5 DCB Struct Reference . . . 7
3.6 footer Struct Reference . . . 7
3.7 gdt_descriptor_struct Struct Reference . . . 7
3.8 gdt_entry_struct Struct Reference . . . 8
3.9 header Struct Reference . . . 8
3.10 heap Struct Reference . . . 8
3.11 idt_entry_struct Struct Reference . . . 9
3.12 idt_struct Struct Reference . . . 9
3.13 index_entry Struct Reference . . . 9
3.14 index_table Struct Reference . . . 9
3.15 List Struct Reference . . . 10
3.16 MemList Struct Reference . . . 10
3.17 page_dir Struct Reference . . . 10
3.18 page_entry Struct Reference . . . 10
3.19 page_table Struct Reference . . . 11
3.20 param Struct Reference . . . 11
3.21 PCB Struct Reference . . . 11
3.22 Queue Struct Reference . . . 11

4 File Documentation 13

4.1 mpx_core/include/core/asm.h File Reference . . . 13
4.2 mpx_core/include/core/interrupts.h File Reference . . . 13
4.3 mpx_core/include/core/io.h File Reference . . . 13
4.3.1 Macro Definition Documentation . . . 13
4.3.1.1 inb . . . 13
4.4 mpx_core/include/core/serial.h File Reference . . . 14
4.5 mpx_core/include/core/tables.h File Reference . . . 14
4.6 mpx_core/include/mem/heap.h File Reference . . . 15
4.7 mpx_core/include/mem/paging.h File Reference . . . 15
4.8 mpx_core/include/string.h File Reference . . . 16
4.8.1 Function Documentation . . . 16
4.8.1.1 atoi() . . . 16

4.8.1.2 isspace() . . . 16
4.8.1.3 memset() . . . 17
4.8.1.4 strcat() . . . 17
4.8.1.5 strcmp() . . . 17
4.8.1.6 strcpy() . . . 18
4.8.1.7 strlen() . . . 18
4.8.1.8 strtok() . . . 18
4.9 mpx_core/include/system.h File Reference . . . 18
4.10 mpx_core/kernel/core/interrupts.c File Reference . . . 19
4.11 mpx_core/kernel/core/kmain.c File Reference . . . 21
4.12 mpx_core/kernel/core/serial.c File Reference . . . 21
4.13 mpx_core/kernel/core/system.c File Reference . . . 22
4.14 mpx_core/kernel/core/tables.c File Reference . . . 22
4.15 mpx_core/kernel/mem/heap.c File Reference . . . 22
4.16 mpx_core/kernel/mem/paging.c File Reference . . . 23
4.17 mpx_core/lib/string.c File Reference . . . 24
4.17.1 Function Documentation . . . 24
4.17.1.1 atoi() . . . 24
4.17.1.2 isspace() . . . 24
4.17.1.3 memset() . . . 25
4.17.1.4 strcat() . . . 25
4.17.1.5 strcmp() . . . 25
4.17.1.6 strcpy() . . . 26
4.17.1.7 strlen() . . . 26
4.17.1.8 strtok() . . . 26
4.18 mpx_core/modules/mpx_supt.c File Reference . . . 26
4.19 mpx_core/modules/mpx_supt.h File Reference . . . 27
4.20 mpx_core/modules/R1/comHand.h File Reference . . . 28
4.20.1 Function Documentation . . . 28
4.20.1.1 comHand() . . . 28
4.21 mpx_core/modules/R1/userFunctions.c File Reference . . . 28
4.21.1 Function Documentation . . . 29
4.21.1.1 BCDtoDec() . . . 29
4.21.1.2 Block() . . . 30
4.21.1.3 Create_PCB() . . . 30
4.21.1.4 DectoBCD() . . . 30
4.21.1.5 Delete_PCB() . . . 31
4.21.1.6 EdgeCase() . . . 31
4.21.1.7 GetDate() . . . 31
4.21.1.8 GetTime() . . . 31
4.21.1.9 Help() . . . 32
4.21.1.10 itoa() . . . 32

4.21.1.11 Resume() . . . 32
4.21.1.12 Set_Priority() . . . 33
4.21.1.13 SetDate() . . . 33
4.21.1.14 SetTime() . . . 33
4.21.1.15 Show_All() . . . 34
4.21.1.16 Show_Blocked() . . . 34
4.21.1.17 Show_PCB() . . . 34
4.21.1.18 Show_Ready() . . . 34
4.21.1.19 Suspend() . . . 35
4.21.1.20 toLowercase() . . . 35
4.21.1.21 Unblock() . . . 35
4.21.1.22 Version() . . . 36
4.21.2 Variable Documentation . . . 36
4.21.2.1 AlarmList . . . 36
4.22 mpx_core/modules/R1/userFunctions.h File Reference . . . 36
4.22.1 Function Documentation . . . 37
4.22.1.1 BCDtoDec() . . . 37
4.22.1.2 Block() . . . 38
4.22.1.3 Create_PCB() . . . 38
4.22.1.4 DectoBCD() . . . 38
4.22.1.5 Delete_PCB() . . . 39
4.22.1.6 EdgeCase() . . . 39
4.22.1.7 GetDate() . . . 39
4.22.1.8 GetTime() . . . 39
4.22.1.9 Help() . . . 40
4.22.1.10 itoa() . . . 40
4.22.1.11 Resume() . . . 40
4.22.1.12 Set_Priority() . . . 41
4.22.1.13 SetDate() . . . 41
4.22.1.14 SetTime() . . . 41
4.22.1.15 Show_All() . . . 42
4.22.1.16 Show_Blocked() . . . 42
4.22.1.17 Show_PCB() . . . 42
4.22.1.18 Show_Ready() . . . 42
4.22.1.19 Suspend() . . . 43
4.22.1.20 toLowercase() . . . 43
4.22.1.21 Unblock() . . . 43
4.22.1.22 Version() . . . 44
4.23 mpx_core/modules/sys_proc_loader.c File Reference . . . 44
4.24 mpx_core/modules/sys_proc_loader.h File Reference . . . 44

Index **45**

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Alarm . . . 5
CMCB . . . 5
context . . . 6
date_time . . . 6
DCB . . . 7
footer . . . 7
gdt_descriptor_struct . . . 7
gdt_entry_struct . . . 8
header . . . 8
heap . . . 8
idt_entry_struct . . . 9
idt_struct . . . 9
index_entry . . . 9
index_table . . . 9
List . . . 10
MemList . . . 10
page_dir . . . 10
page_entry . . . 10
page_table . . . 11
param . . . 11
PCB . . . 11
Queue . . . 11

Chapter 2

File Index

2.1 File List

Here is a list of all documented files with brief descriptions:

mpx_core/include/string.h . . . 16
mpx_core/include/system.h . . . 18
mpx_core/include/core/asm.h . . . 13
mpx_core/include/core/interrupts.h . . . 13
mpx_core/include/core/io.h . . . 13
mpx_core/include/core/serial.h . . . 14
mpx_core/include/core/tables.h . . . 14
mpx_core/include/mem/heap.h . . . 15
mpx_core/include/mem/paging.h . . . 15
mpx_core/kernel/core/interrupts.c . . . 19
mpx_core/kernel/core/kmain.c . . . 21
mpx_core/kernel/core/serial.c . . . 21
mpx_core/kernel/core/system.c . . . 22
mpx_core/kernel/core/tables.c . . . 22
mpx_core/kernel/mem/heap.c . . . 22
mpx_core/kernel/mem/paging.c . . . 23
mpx_core/lib/string.c . . . 24
mpx_core/modules/**DCB.h** . . . **??**
mpx_core/modules/mpx_supt.c . . . 26
mpx_core/modules/mpx_supt.h . . . 27
mpx_core/modules/**PIC.h** . . . **??**
mpx_core/modules/**procsr3.h** . . . **??**
mpx_core/modules/**SerialPortDriver.h** . . . **??**
mpx_core/modules/sys_proc_loader.c . . . 44
mpx_core/modules/sys_proc_loader.h . . . 44
mpx_core/modules/R1/comHand.h . . . 28
mpx_core/modules/R1/userFunctions.c . . . 28
mpx_core/modules/R1/userFunctions.h . . . 36
mpx_core/modules/R2/**PCB.h** . . . **??**
mpx_core/modules/R5/**MCB.h** . . . **??**
mpx_core/modules/R5/**R5commands.h** . . . **??**

Chapter 3

Class Documentation

3.1 Alarm Struct Reference

Public Attributes

- int **hour**
- int **minute**
- int **second**
- char **message** [85]
- struct Alarm * **next**
- struct Alarm * **prev**

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

- mpx_core/modules/R1/userFunctions.h

3.2 CMCB Struct Reference

Public Attributes

- u32int **size**
- struct CMCB * **prev**
- struct CMCB * **next**
- char **Process_name** [10]
- u32int **address**
- int **MEMState**

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

- mpx_core/modules/R5/MCB.h

3.3 context Struct Reference

Public Attributes

- 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**

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

- mpx_core/modules/R2/PCB.h

3.4 date_time Struct Reference

Public Attributes

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

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

- mpx_core/include/system.h

3.5 DCB Struct Reference

Public Attributes

- int **portFlag**
- int * **eventPtr**
- int **status**
- char * **inBuffAddress**
- int **inBuffCounter**
- char * **outBufAddress**
- int **outBuffCounter**
- char **ringBuffer** [100]
- int **inIndex**
- int **outIndex**
- int **rbCounter**

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

- mpx_core/modules/DCB.h

3.6 footer Struct Reference

Public Attributes

- header **head**

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

- mpx_core/include/mem/heap.h

3.7 gdt_descriptor_struct Struct Reference

Public Attributes

- u16int **limit**
- u32int **base**

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

- mpx_core/include/core/tables.h

3.8 gdt_entry_struct Struct Reference

Public Attributes

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

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

- mpx_core/include/core/tables.h

3.9 header Struct Reference

Public Attributes

- int **size**
- int **index_id**

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

- mpx_core/include/mem/heap.h

3.10 heap Struct Reference

Public Attributes

- index_table **index**
- u32int **base**
- u32int **max_size**
- u32int **min_size**

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

- mpx_core/include/mem/heap.h

3.11 idt_entry_struct Struct Reference

Public Attributes

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

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

- mpx_core/include/core/tables.h

3.12 idt_struct Struct Reference

Public Attributes

- u16int **limit**
- u32int **base**

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

- mpx_core/include/core/tables.h

3.13 index_entry Struct Reference

Public Attributes

- int **size**
- int **empty**
- u32int **block**

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

- mpx_core/include/mem/heap.h

3.14 index_table Struct Reference

Public Attributes

- index_entry **table** [TABLE_SIZE]
- int **id**

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

- mpx_core/include/mem/heap.h

3.15 List Struct Reference

Public Attributes

- Alarm * **head**
- Alarm * **tail**

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

- mpx_core/modules/R1/userFunctions.h

3.16 MemList Struct Reference

Public Attributes

- CMCB * **head**

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

- mpx_core/modules/R5/MCB.h

3.17 page_dir Struct Reference

Public Attributes

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

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

- mpx_core/include/mem/paging.h

3.18 page_entry Struct Reference

Public Attributes

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

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

- mpx_core/include/mem/paging.h

3.19 page_table Struct Reference

Public Attributes

- page_entry **pages** [1024]

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

- mpx_core/include/mem/paging.h

3.20 param Struct Reference

Public Attributes

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

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

- mpx_core/modules/mpx_supt.h

3.21 PCB Struct Reference

Public Attributes

- unsigned char **stack** [MEM1K]
- unsigned char * **stackTop**
- struct PCB * **prev**
- struct PCB * **next**
- char **Process_Name** [10]
- int **Process_Class**
- int **Priority**
- int **ReadyState**
- int **SuspendedState**

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

- mpx_core/modules/R2/PCB.h

3.22 Queue Struct Reference

Public Attributes

- int **count**
- PCB * **head**
- PCB * **tail**

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

- mpx_core/modules/R2/PCB.h

Chapter 4

File Documentation

4.1 mpx_core/include/core/asm.h File Reference

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

4.2 mpx_core/include/core/interrupts.h File Reference

Functions

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

4.3 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;                             \
   })
```

4.4 mpx_core/include/core/serial.h File Reference

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)

4.5 mpx_core/include/core/tables.h File Reference

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

Classes

- 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**

4.6 mpx_core/include/mem/heap.h File Reference

Classes

- 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)

4.7 mpx_core/include/mem/paging.h File Reference

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

Classes

- 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)

4.8 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 )
```

Description: Convert an ASCII string to an integer

Parameters

s	String

4.8.1.2 isspace()

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

Description: Determine if a character is whitespace.

Parameters

c	character to check

4.8.1.3 memset()

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

Description: Set a region of memory.

Parameters

s	destination
c	byte to write
n	count

4.8.1.4 strcat()

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

Description: Concatenate the contents of one string onto another.

Parameters

s1	destination
s2	source

4.8.1.5 strcmp()

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

Description: String comparison

Parameters

s1	string 1
s2	string 2

4.8.1.6 strcpy()

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

Description: Copy one string to another.

Parameters

s1	destination
s2	source

4.8.1.7 strlen()

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

Description: Returns the length of a string.

Parameters

s	input string

4.8.1.8 strtok()

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

Description: Split string into tokens

Parameters

s1	String
s2	delimiter

4.9 mpx_core/include/system.h File Reference

Classes

- 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)

4.10 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]

4.11 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/R1/comHand.h"
#include "modules/sys_proc_loader.h"
#include "modules/R1/userFunctions.h"
#include "modules/R5/R5commands.h"
#include "modules/R5/MCB.h"
```

Functions

- void **kmain** (void)

4.12 mpx_core/kernel/core/serial.c File Reference

```
#include <stdint.h>
#include <string.h>
#include <core/io.h>
#include <core/serial.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 ∗cmdBuffer, int ∗count)

Variables

- int **serial_port_out** = 0
- int **serial_port_in** = 0

4.13 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)

4.14 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]

4.15 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

4.16 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**

4.17 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 )
```

Description: Convert an ASCII string to an integer

Parameters

s	String

4.17.1.2 isspace()

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

Description: Determine if a character is whitespace.

Parameters

c	character to check

4.17.1.3 memset()

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

Description: Set a region of memory.

Parameters

s	destination
c	byte to write
n	count

4.17.1.4 strcat()

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

Description: Concatenate the contents of one string onto another.

Parameters

s1	destination
s2	source

4.17.1.5 strcmp()

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

Description: String comparison

Parameters

s1	string 1
s2	string 2

4.17.1.6 strcpy()

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

Description: Copy one string to another.

Parameters

s1	destination
s2	source

4.17.1.7 strlen()

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

Description: Returns the length of a string.

Parameters

s	input string

4.17.1.8 strtok()

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

Description: Split string into tokens

Parameters

s1	String
s2	delimiter

4.18 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** ()
- u32int * **sys_call** (context *registers)

Variables

- param **params**
- int **current_module** = -1
- u32int(* **student_malloc**)(u32int)
- int(* **student_free**)(void *)
- PCB * **cop**
- context * **initial**

4.19 mpx_core/modules/mpx_supt.h File Reference

```
#include <system.h>
#include "R2/PCB.h"
```

Classes

- 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** ()
- u32int * **sys_call** (context *registers)

4.20 mpx_core/modules/R1/comHand.h File Reference

Functions

- int comHand ()

4.20.1 Function Documentation

4.20.1.1 comHand()

```
int comHand ( )
```

Description: Interprets user input to call the appropriate user functions.

4.21 mpx_core/modules/R1/userFunctions.c File Reference

```
#include <stdint.h>
#include <string.h>
#include <system.h>
#include <core/serial.h>
#include <core/io.h>
#include "../mpx_supt.h"
#include "userFunctions.h"
#include "../procsr3.h"
#include "../sys_proc_loader.h"
```

Functions

- void **clear** ()
- char * itoa (int num)
- int BCDtoDec (int BCD)
- int DectoBCD (int Decimal)
- void **printf** (char msg[ ])
- int EdgeCase (char *pointer)
- void SetTime (int hours, int minutes, int seconds)
- void GetTime ()
- void SetDate (int day, int month, int millennium, int year)
- void GetDate ()
- void Version ()
- char toLowercase (char c)
- void Help (char *request)
- void Suspend (char *ProcessName)
- void Resume (char *ProcessName)
- void Set_Priority (char *ProcessName, int Priority)
- void Show_PCB (char *ProcessName)
- void Show_All ()
- void Show_Ready ()
- void Show_Blocked ()
- void Create_PCB (char *ProcessName, int Priority, int Class)
- void Delete_PCB (char *ProcessName)
- void Block (char *ProcessName)
- void Unblock (char *ProcessName)
- void **loader** ()
- void **loadr3** (char *name, u32int func)
- void **yield** ()
- void **loaderinfinite** ()
- List * **getList** ()
- void **loaderalarm** (char text[ ], int hours, int minutes, int seconds)

Variables

- List **AlarmList**

4.21.1 Function Documentation

4.21.1.1 BCDtoDec()

```
int BCDtoDec (
            int BCD )
```

Description: Changes binary number to decimal numbers.

Parameters

value	Binary number to be changed to decimal

4.21.1.2 Block()

```
void Block (
            char * ProcessName )
```

Brief Description: Places a PCD in the blocked state and reinserts it into the correct queue.

Description: Can except a string as a pointer that is the Process Name. The specified PCB will be places in a blocked state and reinserted into the appropriate queue. An error check for a valid name occurs.

Parameters

Process_Name	Character pointer that matches the name of process.

4.21.1.3 Create_PCB()

```
void Create_PCB (
            char * ProcessName,
            int Priority,
            int Class )
```

Brief Description: Calls SetupPCB() and inserts PCB into appropriate queue.

Description: Can except a string as a pointer that is the Process Name. Can accept two integers, Priority and Class. SetupPCB() will be called and the PCB will be inserted into the appropriate queue. An error check for unique and valid Process Name, an error check for valid process class, and an error check for process priority.

Parameters

Process_Name	Character pointer that matches the name of process.
Priority	integer that matches the priority number.
Class	integer that matches the class number.

4.21.1.4 DectoBCD()

```
int DectoBCD (
            int Decimal )
```

Description: Changes decimal numbers to binary numbers.

Parameters

Decimal	Decimal number to be changed to binary

4.21.1.5 Delete_PCB()

```
void Delete_PCB (
            char * ProcessName )
```

Brief Description: Removes PCB from appropriate queue and frees all associated memory.

Description: Can except a string as a pointer that is the Process Name. Removes PCB from the appropriate queue and then frees all associated memory. An error check to make sure process name is valid.

Parameters

Process_Name	Character pointer that matches the name of process.

4.21.1.6 EdgeCase()

```
int EdgeCase (
            char * pointer )
```

Description: Compares pointer char to validate if it is a number or not.

Parameters

Compares	pointer char to validate if it is a number or not.

4.21.1.7 GetDate()

```
void GetDate ( )
```

Description: Returns the full date back to the user in decimal form.

No parameters.

4.21.1.8 GetTime()

```
void GetTime ( )
```

Description: retrieve and return the time values for hours, minutes, and seconds form the clock register using inb(Port,address).

No parameters.

4.21.1.9 Help()

```
void Help (
            char * request )
```

Brief Description: Gives helpful information for one of the functions

Description: Can except a string as a pointer, if the pointer is null then the function will print a complete list of avaliable commands to the console. If the pointer is a avaliable commands then instructions on how to use the command will be printed. If the command does not exist then a message explaining that it is not a valid command will be displayed.

Parameters

request	Character pointer that matches the name of the function that you need help with.

4.21.1.10 itoa()

```
char* itoa (
            int num )
```

Description: An integer is taken and seperated into individual chars and then all placed into a character array. Adapted from geeksforgeeks.org.

Parameters

num	integer to be put into array Title: itoa Author: Neha Mahajan Date: 29 May, 2017 Availability: https://www.geeksforgeeks.org/implement-itoa/

4.21.1.11 Resume()

```
void Resume (
            char * ProcessName )
```

Brief Description: Places a PCD in the not suspended state and reinserts it into the appropriate queue.

Description: Can except a string as a pointer that is the Process Name. Places a PCB in the not suspended state and reinserts it into the appropriate queue. An error check for valid Process Name.

Parameters

Process_Name	Character pointer that matches the name of process.

4.21.1.12 Set_Priority()

```
void Set_Priority (
            char * ProcessName,
            int Priority )
```

Brief Description: Sets PCB priority and reinserts the process into the correct place in the correct queue.

Description: Can except a string as a pointer that is the Process Name. Can accept and integer than is the Priority. Sets a PCB's priority and reinserts the process into the correct place in the correct queue. An error check for valid Process Name and an error check for a valid priority 1 - 9.

Parameters

Process_Name	Character pointer that matches the name of process.
Priority	integer that matches the priority number.

4.21.1.13 SetDate()

```
void SetDate (
            int day,
            int month,
            int millennium,
            int year )
```

Description: Sets the date register to the new values that the user inputed, all values must be inputed as SetDime(day, month, millenial, year).

Parameters

day	Integer to be set in the Day position
month	Integer to be set in the Month position
millenial	Integer to be set in the Millenial position
year	Integer to be set in the Year position

4.21.1.14 SetTime()

```
void SetTime (
            int hours,
            int minutes,
            int seconds )
```

Description: sets the time register to the new values that the user inputed, all values must be inputed as SetTime(Hours, Minutes, Seconds).

Parameters

hours	Integer to be set in the Hour position
minutes	Integer to be set in the Minutes position
seconds	Integer to be set in the Seconds position

4.21.1.15 Show_All()

```
void Show_All ( )
```

Brief Description: Displays the process name, class, state, suspended status, and priority of all PCB in the ready and blocked queues.

Description: The process name, claas, state, suspend status, and priority of each of he PCB's in the ready and blocked queues.

4.21.1.16 Show_Blocked()

```
void Show_Blocked ( )
```

Brief Description: Displays the process name, class, state, suspended status, and priority of all PCB in the blocked queue.

Description: The process name, claas, state, suspend status, and priority of each of he PCB's in the blocked queue.

4.21.1.17 Show_PCB()

```
void Show_PCB (
            char * ProcessName )
```

Brief Description: Displays the process name, class, state, suspended status, and priority of a PCB.

Description: Can except a string as a pointer that is the Process Name. The process name, claas, state, suspend status, and priority of a PCB are displayed. An error check for a valid name occurs.

Parameters

Process_Name	Character pointer that matches the name of process

4.21.1.18 Show_Ready()

```
void Show_Ready ( )
```

Brief Description: Displays the process name, class, state, suspended status, and priority of all PCB in the ready queue.

Description: The process name, claas, state, suspend status, and priority of each of he PCB's in the ready queue.

4.21.1.19 Suspend()

```
void Suspend (
            char * ProcessName )
```

Brief Description: Places a PCD in the suspended state and reinserts it into the appropriate queue.

Description: Can except a string as a pointer that is the Process Name. Places a PCB in the suspended state and reinserts it into the appropriate queue. An error check for valid Process Name.

Parameters

Process_Name	Character pointer that matches the name of process.

4.21.1.20 toLowercase()

```
char toLowercase (
            char c )
```

Description: If a letter is uppercase, it changes it to lowercase. (char)

Parameters

c	Character that is to be changed to its lowercase equivalent

4.21.1.21 Unblock()

```
void Unblock (
            char * ProcessName )
```

Brief Description: Places a PCD in the unblocked state and reinserts it into the correct queue.

Description: Can except a string as a pointer that is the Process Name. The specified PCB will be places in an unblocked state and reinserted into the appropriate queue. An error check for a valid name occurs.

Parameters

Process_Name	Character pointer that matches the name of process.

4.21.1.22 Version()

```
void Version ( )
```

Description: Simply returns a char containing "Version: R(module).(the iteration that module is currently on).

No parameters.

4.21.2 Variable Documentation

4.21.2.1 AlarmList

```
List AlarmList
```

Initial value:

```
={
  .head = NULL,
  .tail = NULL
}
```

4.22 mpx_core/modules/R1/userFunctions.h File Reference

Classes

- struct Alarm
- struct List

Macros

- #define **RED** "\x1B[31m"
- #define **GRN** "\x1B[32m"
- #define **YEL** "\x1B[33m"
- #define **BLU** "\x1B[34m"
- #define **MAG** "\x1B[35m"
- #define **CYN** "\x1B[36m"
- #define **WHT** "\x1B[37m"
- #define **RESET** "\x1B[0m"

Typedefs

- typedef struct Alarm **Alarm**
- typedef struct List **List**

Functions

- void SetTime (int hours, int minutes, int seconds)
- void GetTime ()
- int DectoBCD (int Decimal)
- void **clear** ()
- char * itoa (int num)
- void SetDate (int day, int month, int millennium, int year)
- int BCDtoDec (int BCD)
- void GetDate ()
- void Version ()
- void Help (char *request)
- void **printf** (char msg[])
- int EdgeCase (char *pointer)
- char toLowercase (char c)
- void Suspend (char *ProcessName)
- void Resume (char *ProcessName)
- void Set_Priority (char *ProcessName, int Priority)
- void Show_PCB (char *ProcessName)
- void Show_All ()
- void Show_Ready ()
- void Show_Blocked ()
- void Create_PCB (char *ProcessName, int Priority, int Class)
- void Delete_PCB (char *ProcessName)
- void Block (char *ProcessName)
- void Unblock (char *ProcessName)
- void **loader** ()
- void **loadr3** (char *name, u32int func)
- void **yield** ()
- void **loaderinfinite** ()
- List * **getList** ()
- void **loaderalarm** ()

4.22.1 Function Documentation

4.22.1.1 BCDtoDec()

```
int BCDtoDec (
            int BCD )
```

Description: Changes binary number to decimal numbers.

Parameters

value	Binary number to be changed to decimal

4.22.1.2 Block()

```
void Block (
            char * ProcessName )
```

Brief Description: Places a PCD in the blocked state and reinserts it into the correct queue.

Description: Can except a string as a pointer that is the Process Name. The specified PCB will be places in a blocked state and reinserted into the appropriate queue. An error check for a valid name occurs.

Parameters

Process_Name	Character pointer that matches the name of process.

4.22.1.3 Create_PCB()

```
void Create_PCB (
            char * ProcessName,
            int Priority,
            int Class )
```

Brief Description: Calls SetupPCB() and inserts PCB into appropriate queue.

Description: Can except a string as a pointer that is the Process Name. Can accept two integers, Priority and Class. SetupPCB() will be called and the PCB will be inserted into the appropriate queue. An error check for unique and valid Process Name, an error check for valid process class, and an error check for process priority.

Parameters

Process_Name	Character pointer that matches the name of process.
Priority	integer that matches the priority number.
Class	integer that matches the class number.

4.22.1.4 DectoBCD()

```
int DectoBCD (
            int Decimal )
```

Description: Changes decimal numbers to binary numbers.

Parameters

Decimal	Decimal number to be changed to binary

4.22.1.5 Delete_PCB()

```
void Delete_PCB (
            char * ProcessName )
```

Brief Description: Removes PCB from appropriate queue and frees all associated memory.

Description: Can except a string as a pointer that is the Process Name. Removes PCB from the appropriate queue and then frees all associated memory. An error check to make sure process name is valid.

Parameters

Process_Name	Character pointer that matches the name of process.

4.22.1.6 EdgeCase()

```
int EdgeCase (
            char * pointer )
```

Description: Compares pointer char to validate if it is a number or not.

Parameters

Compares	pointer char to validate if it is a number or not.

4.22.1.7 GetDate()

```
void GetDate ( )
```

Description: Returns the full date back to the user in decimal form.

No parameters.

4.22.1.8 GetTime()

```
void GetTime ( )
```

Description: retrieve and return the time values for hours, minutes, and seconds form the clock register using inb(Port,address).

No parameters.

4.22.1.9 Help()

```
void Help (
            char * request )
```

Brief Description: Gives helpful information for one of the functions

Description: Can except a string as a pointer, if the pointer is null then the function will print a complete list of avaliable commands to the console. If the pointer is a avaliable commands then instructions on how to use the command will be printed. If the command does not exist then a message explaining that it is not a valid command will be displayed.

Parameters

request	Character pointer that matches the name of the function that you need help with.

4.22.1.10 itoa()

```
char* itoa (
            int num )
```

Description: An integer is taken and seperated into individual chars and then all placed into a character array. Adapted from geeksforgeeks.org.

Parameters

num	integer to be put into array Title: itoa Author: Neha Mahajan Date: 29 May, 2017 Availability: https://www.geeksforgeeks.org/implement-itoa/

4.22.1.11 Resume()

```
void Resume (
            char * ProcessName )
```

Brief Description: Places a PCD in the not suspended state and reinserts it into the appropriate queue.

Description: Can except a string as a pointer that is the Process Name. Places a PCB in the not suspended state and reinserts it into the appropriate queue. An error check for valid Process Name.

Parameters

Process_Name	Character pointer that matches the name of process.

4.22.1.12 Set_Priority()

```
void Set_Priority (
            char * ProcessName,
            int Priority )
```

Brief Description: Sets PCB priority and reinserts the process into the correct place in the correct queue.

Description: Can except a string as a pointer that is the Process Name. Can accept and integer than is the Priority. Sets a PCB's priority and reinserts the process into the correct place in the correct queue. An error check for valid Process Name and an error check for a valid priority 1 - 9.

Parameters

Process_Name	Character pointer that matches the name of process.
Priority	integer that matches the priority number.

4.22.1.13 SetDate()

```
void SetDate (
            int day,
            int month,
            int millennium,
            int year )
```

Description: Sets the date register to the new values that the user inputed, all values must be inputed as SetDime(day, month, millenial, year).

Parameters

day	Integer to be set in the Day position
month	Integer to be set in the Month position
millenial	Integer to be set in the Millenial position
year	Integer to be set in the Year position

4.22.1.14 SetTime()

```
void SetTime (
            int hours,
            int minutes,
            int seconds )
```

Description: sets the time register to the new values that the user inputed, all values must be inputed as SetTime(Hours, Minutes, Seconds).

Parameters

hours	Integer to be set in the Hour position
minutes	Integer to be set in the Minutes position
seconds	Integer to be set in the Seconds position

4.22.1.15 Show_All()

```
void Show_All ( )
```

Brief Description: Displays the process name, class, state, suspended status, and priority of all PCB in the ready and blocked queues.

Description: The process name, claas, state, suspend status, and priority of each of he PCB's in the ready and blocked queues.

4.22.1.16 Show_Blocked()

```
void Show_Blocked ( )
```

Brief Description: Displays the process name, class, state, suspended status, and priority of all PCB in the blocked queue.

Description: The process name, claas, state, suspend status, and priority of each of he PCB's in the blocked queue.

4.22.1.17 Show_PCB()

```
void Show_PCB (
            char * ProcessName )
```

Brief Description: Displays the process name, class, state, suspended status, and priority of a PCB.

Description: Can except a string as a pointer that is the Process Name. The process name, claas, state, suspend status, and priority of a PCB are displayed. An error check for a valid name occurs.

Parameters

Process_Name	Character pointer that matches the name of process

4.22.1.18 Show_Ready()

```
void Show_Ready ( )
```

Brief Description: Displays the process name, class, state, suspended status, and priority of all PCB in the ready queue.

Description: The process name, claas, state, suspend status, and priority of each of he PCB's in the ready queue.

4.22.1.19 Suspend()

```
void Suspend (
            char * ProcessName )
```

Brief Description: Places a PCD in the suspended state and reinserts it into the appropriate queue.

Description: Can except a string as a pointer that is the Process Name. Places a PCB in the suspended state and reinserts it into the appropriate queue. An error check for valid Process Name.

Parameters

Process_Name	Character pointer that matches the name of process.

4.22.1.20 toLowercase()

```
char toLowercase (
            char c )
```

Description: If a letter is uppercase, it changes it to lowercase. (char)

Parameters

c	Character that is to be changed to its lowercase equivalent

4.22.1.21 Unblock()

```
void Unblock (
            char * ProcessName )
```

Brief Description: Places a PCD in the unblocked state and reinserts it into the correct queue.

Description: Can except a string as a pointer that is the Process Name. The specified PCB will be places in an unblocked state and reinserted into the appropriate queue. An error check for a valid name occurs.

Parameters

Process_Name	Character pointer that matches the name of process.

4.22.1.22 Version()

```
void Version ( )
```

Description: Simply returns a char containing "Version: R(module).(the iteration that module is currently on).

No parameters.

4.23 mpx_core/modules/sys_proc_loader.c File Reference

```
#include <stdint.h>
#include <string.h>
#include <system.h>
#include <core/serial.h>
#include <core/io.h>
#include "mpx_supt.h"
#include "R1/userFunctions.h"
#include "procsr3.h"
#include "R1/comHand.h"
#include "sys_proc_loader.h"
```

Functions

- void **sysLoader** ()
- void **loadSysProc** (char ∗name, u32int func, int priority)
- void **InfiniteProc** ()
- void **AlarmProc** ()

4.24 mpx_core/modules/sys_proc_loader.h File Reference

Functions

- void **sysLoader** ()
- void **loadSysProc** (char ∗name, u32int func, int priority)
- void **InfiniteProc** ()
- void **AlarmProc** ()

Index

Alarm, 5
AlarmList
    userFunctions.c, 36
atoi
    string.c, 24
    string.h, 16

BCDtoDec
    userFunctions.c, 29
    userFunctions.h, 37
Block
    userFunctions.c, 30
    userFunctions.h, 37

CMCB, 5
comHand
    comHand.h, 28
comHand.h
    comHand, 28
context, 6
Create_PCB
    userFunctions.c, 30
    userFunctions.h, 38

date_time, 6
DCB, 7
DectoBCD
    userFunctions.c, 30
    userFunctions.h, 38
Delete_PCB
    userFunctions.c, 31
    userFunctions.h, 38

EdgeCase
    userFunctions.c, 31
    userFunctions.h, 39

footer, 7

gdt_descriptor_struct, 7
gdt_entry_struct, 8
GetDate
    userFunctions.c, 31
    userFunctions.h, 39
GetTime
    userFunctions.c, 31
    userFunctions.h, 39

header, 8
heap, 8
Help
    userFunctions.c, 31
    userFunctions.h, 39

idt_entry_struct, 9
idt_struct, 9
inb
    io.h, 13
index_entry, 9
index_table, 9
io.h
    inb, 13
isspace
    string.c, 24
    string.h, 16
itoa
    userFunctions.c, 32
    userFunctions.h, 40

List, 10

MemList, 10
memset
    string.c, 24
    string.h, 16
mpx_core/include/core/asm.h, 13
mpx_core/include/core/interrupts.h, 13
mpx_core/include/core/io.h, 13
mpx_core/include/core/serial.h, 14
mpx_core/include/core/tables.h, 14
mpx_core/include/mem/heap.h, 15
mpx_core/include/mem/paging.h, 15
mpx_core/include/string.h, 16
mpx_core/include/system.h, 18
mpx_core/kernel/core/interrupts.c, 19
mpx_core/kernel/core/kmain.c, 21
mpx_core/kernel/core/serial.c, 21
mpx_core/kernel/core/system.c, 22
mpx_core/kernel/core/tables.c, 22
mpx_core/kernel/mem/heap.c, 22
mpx_core/kernel/mem/paging.c, 23
mpx_core/lib/string.c, 24
mpx_core/modules/mpx_supt.c, 26
mpx_core/modules/mpx_supt.h, 27
mpx_core/modules/R1/comHand.h, 28
mpx_core/modules/R1/userFunctions.c, 28
mpx_core/modules/R1/userFunctions.h, 36
mpx_core/modules/sys_proc_loader.c, 44
mpx_core/modules/sys_proc_loader.h, 44

page_dir, 10

page_entry, 10
page_table, 11
param, 11
PCB, 11

Queue, 11

Resume
    userFunctions.c, 32
    userFunctions.h, 40

Set_Priority
    userFunctions.c, 32
    userFunctions.h, 40
SetDate
    userFunctions.c, 33
    userFunctions.h, 41
SetTime
    userFunctions.c, 33
    userFunctions.h, 41
Show_All
    userFunctions.c, 34
    userFunctions.h, 42
Show_Blocked
    userFunctions.c, 34
    userFunctions.h, 42
Show_PCB
    userFunctions.c, 34
    userFunctions.h, 42
Show_Ready
    userFunctions.c, 34
    userFunctions.h, 42
strcat
    string.c, 25
    string.h, 17
strcmp
    string.c, 25
    string.h, 17
strcpy
    string.c, 25
    string.h, 17
string.c
    atoi, 24
    isspace, 24
    memset, 24
    strcat, 25
    strcmp, 25
    strcpy, 25
    strlen, 26
    strtok, 26
string.h
    atoi, 16
    isspace, 16
    memset, 16
    strcat, 17
    strcmp, 17
    strcpy, 17
    strlen, 18
    strtok, 18
strlen
    string.c, 26
    string.h, 18
strtok
    string.c, 26
    string.h, 18
Suspend
    userFunctions.c, 35
    userFunctions.h, 43

toLowercase
    userFunctions.c, 35
    userFunctions.h, 43

Unblock
    userFunctions.c, 35
    userFunctions.h, 43
userFunctions.c
    AlarmList, 36
    BCDtoDec, 29
    Block, 30
    Create_PCB, 30
    DectoBCD, 30
    Delete_PCB, 31
    EdgeCase, 31
    GetDate, 31
    GetTime, 31
    Help, 31
    itoa, 32
    Resume, 32
    Set_Priority, 32
    SetDate, 33
    SetTime, 33
    Show_All, 34
    Show_Blocked, 34
    Show_PCB, 34
    Show_Ready, 34
    Suspend, 35
    toLowercase, 35
    Unblock, 35
    Version, 36
userFunctions.h
    BCDtoDec, 37
    Block, 37
    Create_PCB, 38
    DectoBCD, 38
    Delete_PCB, 38
    EdgeCase, 39
    GetDate, 39
    GetTime, 39
    Help, 39
    itoa, 40
    Resume, 40
    Set_Priority, 40
    SetDate, 41
    SetTime, 41
    Show_All, 42
    Show_Blocked, 42
    Show_PCB, 42

Show_Ready, 42
Suspend, 43
toLowercase, 43
Unblock, 43
Version, 44

Version
userFunctions.c, 36
userFunctions.h, 44