MPX_CORE GROUP 9

Version Version R1
Mon Feb 8 2021

Table of Contents

Table of contents

MPX_Core Project

This project is about building a primitive operating system that includes a command line interface, process management and memory management

Data Structure Index

Data Structures

Here are the data structures with brief descriptions:

File Index

File List

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

Data Structure Documentation

date_time Struct Reference

Data Fields

- 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:

- include/system.h

footer Struct Reference

Collaboration diagram for footer:

Data Fields

- **header head**

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

- include/mem/heap.h

gdt_descriptor_struct Struct Reference

Data Fields

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

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

- include/core/tables.h

gdt_entry_struct Struct Reference

Data Fields

- 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:

- include/core/tables.h

header Struct Reference

Data Fields

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

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

- include/mem/heap.h

heap Struct Reference

Collaboration diagram for heap:

Data Fields

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

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

- include/mem/heap.h

idt_entry_struct Struct Reference

Data Fields

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

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

- include/core/tables.h

idt_struct Struct Reference

Data Fields

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

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

- include/core/tables.h

index_entry Struct Reference

Data Fields

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

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

- include/mem/heap.h

index_table Struct Reference

Collaboration diagram for index_table:

Data Fields

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

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

- include/mem/heap.h

page_dir Struct Reference

Collaboration diagram for page_dir:

Data Fields

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

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

- include/mem/paging.h

page_entry Struct Reference

Data Fields

- 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:

- include/mem/paging.h

page_table Struct Reference

Collaboration diagram for page_table:

Data Fields

- **page_entry pages** [1024]

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

- include/mem/paging.h

param Struct Reference

Data Fields

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

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

- modules/**mpx_supt.h**

File Documentation

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.h"
```

Include dependency graph for kmain.c:

Functions

- void **kmain** (void)

Detailed Description

Kernel main. The first function called after the bootloader. Initialization of hardware, system structures, devices, and initial processes happens here.

Initial Kernel – by Forrest Desjardin, 2013, Modifications by: Andrew Duncan 2014, John Jacko 2017 Ben Smith 2018, and Alex Wilson 2019

kernel/core/serial.c File Reference

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

Include dependency graph for serial.c:

Macros

- #define **NO_ERROR**   0

Functions

- int **init_serial** (int device)
  Initializes serial 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)

Variables

- int **serial_port_out** = 0
  Active devices used for serial output.

- int **serial_port_in** = 0
  Active devices used for serial output.

- int **i** = 0
  counter for polling

- int **cursor** =0
  Keepts track of the cursor position in the terminal.

Detailed Description

Contains methods and variables used for serial input and output.

Function Documentation

int init_serial (int *device*)

Initializes serial device.

Parameters

int	device

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

Repeatedly checks status register to see if a bit has been entered, stores and prints, or does another action to the input.

Parameters

	char *buffer, int *count

int serial_print (const char * *msg*)

Writes a message to the active serial output device.

Parameters

const	char *msg

int serial_println (const char * *msg*)

Writes a message to the active serial output device. Appends a newline character.

Parameters

const	char *msg

int set_serial_in (int *device*)

Sets serial_port_in to the given device address. All serial input, such as console input via a virtual machine, QEMU/Bochs/etc, will be directed to this device.

Parameters

int	device

int set_serial_out (int *device*)

Sets serial_port_out to the given device address. All serial output, such as that from serial_println, will be directed to this device.

Parameters

int	device

lib/string.c File Reference

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

Include dependency graph for string.c:

Functions

- int **strlen** (const char *s)
- char * **strcpy** (char *s1, const char *s2)
- int **atoi** (const char *s)
- char * **itoa** (int num, char *buffer, int base)
- char * **reverse** (char *str, int **i**, int j)
- void **swap** (char *x, char *y)
- int **strcmp** (const char *s1, const char *s2)
- int **strncmp** (const char *s1, const char *s2, size_t n)
- 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)

Detailed Description

Implementation of C string functions

Function Documentation

int atoi (const char * *s*)

Convert an ASCII string to an integer

Parameters

const	char *s

int isspace (const char * *c*)

Determine if a character is whitespace.

Parameters

const	char *c-character to check

char* itoa (int *num*, char * *buffer*, int *base*)

Convert an integer to ASCII string

Parameters

int	num, char *buffer, int base

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

Set a region of memory.

Parameters

void	*s-destination, int c-byte to write, size_t n-count

char* reverse (char * *str*, int *i*, int *j*)

reverses contents of string

Parameters

char	*str, int i, int j

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

Concatenate the contents of one string onto another.

Parameters

char	*s1-destination, const char *s2-source

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

String comparison

Parameters

const	char *s1-string, const char *s2-string

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

Copy one string to another.

Parameters

char	*s1-destination, char *s2-source

int strlen (const char * *s*)

Returns the length of a string.

Parameters

const	char *s

int strncmp (const char * *s1*, const char * *s2*, size_t *n*)

String comparison for a given number of characters

Parameters

const	char *s1-string 1, const char *s2-string 2, n-size_t

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

Split string into tokens

Parameters

char	*s1-string, s2-delimiter

void swap (char * *x*, char * *y*)

swaps two char values

Parameters

char	*x, char *y

modules/comhand.c File Reference

```
#include "comhand.h"
#include "mpx_supt.h"
#include "version.h"
#include "help.h"
#include "gettime.h"
#include "getdate.h"
#include "settime.h"
#include "setdate.h"
#include <core/serial.h>
#include <string.h>
```

Include dependency graph for comhand.c:

Functions

- int **comhandler** ()

Detailed Description

handles the input commands from the command line

Function Documentation

int comhandler ()

Calls the polling function in **serial.c** and interprets the commands given to it

modules/comhand.h File Reference

This graph shows which files directly or indirectly include this file:

Macros

- #define **_COMMAND_H**
- #define **BUFFER** 100

Functions

- int **comhandler** ()

Detailed Description

comhand header file

Function Documentation

int comhandler ()

Calls the polling function in **serial.c** and interprets the commands given to it

modules/getdate.c File Reference

```
#include "getdate.h"
#include "gettime.h"
#include <core/serial.h>
#include "mpx_supt.h"
#include <core/io.h>
#include <string.h>
```

Include dependency graph for getdate.c:

Functions

- void **getdate** ()

Detailed Description

Contains function **getdate()** to display the current date

Function Documentation

void getdate ()

Displays the current date on the machine

modules/getdate.h File Reference

This graph shows which files directly or indirectly include this file:

Functions

- void **getdate** ()

Function Documentation

void getdate ()

Displays the current date on the machine

modules/gettime.c File Reference

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

Include dependency graph for gettime.c:

Functions

- void **gettime** ()
- int **BCDToDecimal** (int BCD)
- int **DecimalToBCD** (int decimal)

Detailed Description

Contains function **gettime()** to display the current time

Function Documentation

int BCDToDecimal (int *BCD*)

Converts BCD (Binary Coded Decimal) to Decimal

Parameters

int	BCD

int DecimalToBCD (int *decimal*)

Converts Decimal to BCD (Binary Coded Deciaml)

Parameters

int	decimal

void gettime ()

Gets the current time running on the system

modules/gettime.h File Reference

This graph shows which files directly or indirectly include this file:

Functions

- void **gettime** ()
- int **BCDToDecimal** (int BCD)
- int **DecimalToBCD** (int decimal)

Function Documentation

int BCDToDecimal (int *BCD*)

Converts BCD (Binary Coded Decimal) to Decimal

Parameters

int	BCD

int DecimalToBCD (int *decimal*)

Converts Decimal to BCD (Binary Coded Deciaml)

Parameters

int	decimal

void gettime ()

Gets the current time running on the system

modules/help.c File Reference

```
#include <core/serial.h>
#include "mpx_supt.h"
#include "help.h"
#include <string.h>
#include "comhand.h"
```

Include dependency graph for help.c:

Functions

- void **help** (char *msg)

Detailed Description

Handles the help pages for all commands on the system

Function Documentation

void help (char * *msg*)

Displays the correct help page for the given command

Parameters

char	*msg

modules/help.h File Reference

This graph shows which files directly or indirectly include this file:

Functions

- void **help** (char *msg)

Function Documentation

void help (char * *msg*)

Displays the correct help page for the given command

Parameters

char	*msg

modules/mpx_supt.c File Reference

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

Include dependency graph for mpx_supt.c:

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** ()

Variables

- **param params**
  global variable containing parameter used when making system calls via sys_req
- int **current_module** = -1
  global for the current module
- u32int(* **student_malloc**)(u32int)
- int(* **student_free**)(void *)

Detailed Description

contains the MPX support functions

Function Documentation

void idle ()

The idle process, used in dispatching it will only be dispatched if NO other processes are available to execute.

void mpx_init (int *cur_mod*)

```
Initialize MPX support software, based
    on the current module.  The operation of
    MPX will changed based on the module selected.
    THIS must be called as the first executable
    statement inside your command handler.
```

Parameters

int	cur_mod

void* sys_alloc_mem (u32int *size*)

Allocates a block of memory (similar to malloc)

Parameters

u32int	size

int sys_free_mem (void * *ptr*)

Frees memory

Parameters

void	*ptr

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

This function is use to issue system requests for service.

Parameters

int	op_code, int device_id, char *buffer_ptr, int *count_ptr

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

Sets the memory free function for sys_free_mem

Parameters

s1-destination,s2-source	

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

Sets the memory allocation function for sys_alloc_mem

Parameters

Function	pointer

modules/mpx_supt.h File Reference

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

Include dependency graph for mpx_supt.h:

This graph shows which files directly or indirectly include this file:

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** ()

Function Documentation

void idle ()

The idle process, used in dispatching it will only be dispatched if NO other processes are available to execute.

void mpx_init (int *cur_mod*)

```
Initialize MPX support software, based
```

```
    on the current module.  The operation of
    MPX will changed based on the module selected.
    THIS must be called as the first executable
    statement inside your command handler.
```

Parameters

int	cur_mod

void* sys_alloc_mem (u32int *size*)

Allocates a block of memory (similar to malloc)

Parameters

u32int	size

int sys_free_mem (void * *ptr*)

Frees memory

Parameters

void	*ptr

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

This function is use to issue system requests for service.

Parameters

int	op_code, int device_id, char *buffer_ptr, int *count_ptr

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

Sets the memory free function for sys_free_mem

Parameters

s1-destination,s2-source	

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

Sets the memory allocation function for sys_alloc_mem

Parameters

Function	pointer

modules/setdate.c File Reference

```
#include "gettime.h"
#include "setdate.h"
#include <core/serial.h>
#include "mpx_supt.h"
#include <core/io.h>
#include <string.h>
```

Include dependency graph for setdate.c:

Functions

- void **setdate** (char *date)

Detailed Description

contains **setdate(char *date)** function to set a new date on the system

Function Documentation

void setdate (char * *date*)

sets the date to the given input

Parameters

char	*date

modules/setdate.h File Reference

This graph shows which files directly or indirectly include this file:

Functions

- void **setdate** (char *date)

Function Documentation

void setdate (char * *date*)

sets the date to the given input

Parameters

char	*date

modules/settime.c File Reference

```
#include "gettime.h"
#include "settime.h"
#include <core/serial.h>
#include "mpx_supt.h"
#include <core/io.h>
#include <string.h>
```

Include dependency graph for settime.c:

Functions

- void **settime** (char *time)

Detailed Description

Sets a new time given by the user

Function Documentation

void settime (char * *time*)

Allows user to change the time on the system

Parameters

char	*time

modules/settime.h File Reference

This graph shows which files directly or indirectly include this file:

Functions

- void **settime** (char *time)

Function Documentation

void settime (char * *time*)

Allows user to change the time on the system

Parameters

char	*time

modules/version.c File Reference

```
#include "version.h"
#include <core/serial.h>
#include "mpx_supt.h"
```

Include dependency graph for version.c:

Functions

- int **version** ()

Detailed Description

Displays the version number of the mpx_core

Function Documentation

int version ()

Helps display the version number of the current system.

modules/version.h File Reference

This graph shows which files directly or indirectly include this file:

Macros

- #define **VERSION**   "Version R1"

Functions

- int **version** ()

Function Documentation

int version ()

Helps display the version number of the current system.

Index

INDEX