MacaroniOS

R1 !!!DRAFT!!!

Generated on Wed Sep 10 2025 14:01:16 for MacaroniOS by Doxygen 1.9.8

Wed Sep 10 2025 14:01:16

1 Macaroni Penguins 1
1.1 GETTING STARTED . . . 1
1.2 CONTRIBUTING . . . 1
1.3 DOCUMENTATION . . . 1
1.4 DOXYGEN . . . 2

2 File Index 3
2.1 File List . . . 3

3 File Documentation 5
3.1 include/comhand.h File Reference . . . 5
3.1.1 Detailed Description . . . 5
3.1.2 Function Documentation . . . 5
3.1.2.1 trim_Input() . . . 5
3.2 comhand.h . . . 6
3.3 include/ctype.h File Reference . . . 6
3.3.1 Detailed Description . . . 6
3.3.2 Function Documentation . . . 6
3.3.2.1 isspace() . . . 6
3.4 ctype.h . . . 6
3.5 include/exit.h File Reference . . . 7
3.5.1 Detailed Description . . . 7
3.5.2 Function Documentation . . . 7
3.5.2.1 exit_command() . . . 7
3.6 exit.h . . . 7
3.7 include/help.h File Reference . . . 8
3.7.1 Detailed Description . . . 8
3.8 help.h . . . 8
3.9 include/itoa.h File Reference . . . 8
3.9.1 Detailed Description . . . 8
3.9.2 Function Documentation . . . 8
3.9.2.1 itoa() . . . 8
3.10 itoa.h . . . 9
3.11 include/memory.h File Reference . . . 9
3.12 memory.h . . . 9
3.13 device.h . . . 9
3.14 include/mpx/gdt.h File Reference . . . 9
3.14.1 Detailed Description . . . 10
3.14.2 Function Documentation . . . 10
3.14.2.1 gdt_init() . . . 10
3.15 gdt.h . . . 10
3.16 include/mpx/interrupts.h File Reference . . . 10
3.16.1 Detailed Description . . . 10

3.16.2 Macro Definition Documentation . . . 11
3.16.2.1 cli . . . 11
3.16.2.2 sti . . . 11
3.16.3 Function Documentation . . . 11
3.16.3.1 idt_init() . . . 11
3.16.3.2 idt_install() . . . 11
3.16.3.3 irq_init() . . . 11
3.16.3.4 pic_init() . . . 11
3.17 interrupts.h . . . 12
3.18 include/mpx/io.h File Reference . . . 12
3.18.1 Detailed Description . . . 12
3.18.2 Macro Definition Documentation . . . 12
3.18.2.1 inb . . . 12
3.18.2.2 outb . . . 13
3.19 io.h . . . 13
3.20 include/mpx/panic.h File Reference . . . 13
3.20.1 Detailed Description . . . 13
3.20.2 Function Documentation . . . 14
3.20.2.1 __attribute__() . . . 14
3.21 panic.h . . . 14
3.22 include/mpx/serial.h File Reference . . . 14
3.22.1 Detailed Description . . . 14
3.22.2 Function Documentation . . . 14
3.22.2.1 serial_init() . . . 14
3.22.2.2 serial_out() . . . 15
3.22.2.3 serial_poll() . . . 15
3.23 serial.h . . . 16
3.24 include/mpx/vm.h File Reference . . . 16
3.24.1 Detailed Description . . . 16
3.24.2 Function Documentation . . . 16
3.24.2.1 kmalloc() . . . 16
3.24.2.2 vm_init() . . . 17
3.25 vm.h . . . 17
3.26 include/processes.h File Reference . . . 17
3.26.1 Detailed Description . . . 17
3.26.2 Function Documentation . . . 17
3.26.2.1 proc1() . . . 17
3.26.2.2 proc2() . . . 18
3.26.2.3 proc3() . . . 18
3.26.2.4 proc4() . . . 18
3.26.2.5 proc5() . . . 18
3.26.2.6 sys_idle_process() . . . 18

3.27 processes.h . . . 18
3.28 include/stdlib.h File Reference . . . 19
3.28.1 Detailed Description . . . 19
3.28.2 Function Documentation . . . 19
3.28.2.1 atoi() . . . 19
3.29 stdlib.h . . . 19
3.30 include/string.h File Reference . . . 19
3.30.1 Detailed Description . . . 20
3.30.2 Function Documentation . . . 20
3.30.2.1 memcpy() . . . 20
3.30.2.2 memset() . . . 20
3.30.2.3 strcmp() . . . 21
3.30.2.4 strlen() . . . 21
3.30.2.5 strtok() . . . 21
3.31 string.h . . . 22
3.32 include/sys_req.h File Reference . . . 22
3.32.1 Detailed Description . . . 22
3.32.2 Function Documentation . . . 22
3.32.2.1 sys_req() . . . 22
3.33 sys_req.h . . . 23
3.34 include/version.h File Reference . . . 23
3.34.1 Detailed Description . . . 24
3.34.2 Function Documentation . . . 24
3.34.2.1 version_command() . . . 24
3.35 version.h . . . 24

Index **25**

Chapter 1

Macaroni Penguins

CS450: Operating Systems Structure

1.1 GETTING STARTED

Install WSL if you need to:
```
wsl --install -d ubuntu
```

Clone this repo into a linux environment (WSL, Ubuntu, etc):
```
git clone https://github.com/WVU-CS450/MacaroniPenguins.git
```

Prep your linux environment by running the following commands or `./setup.sh`:
```
sudo apt update
sudo apt install -y clang make nasm git binutils-i686-linux-gnu qemu-system-x86 gdb doxygen
```

Then run `make` and `./mpx.sh`.

1.2 CONTRIBUTING

After making changes, running `version` will show that your working directory is 'dirty'. This simply means that you have uncommitted changes.

Ensure you have checked out the correct branch and pulled its latest changes. Stage/add the relevant files before committing them.

Now you can run `make clean` and `make` again, run `./mpx.sh`, and finally run `version` to see your latest commit hash and showing that your working directory is 'clean'.

When you're done, add your contributions to `CONTRIBUTIONS.docx` and save it as a pdf.

1.3 DOCUMENTATION

See the `doc/` folder for:

- [User's Manual]()
- [Programmer's Manual]()
- `CONTRIBUTIONS`

1.4 DOXYGEN

Create the configuration file (convention is a Doxyfile):
```
doxygen -g <config-file>
```

Edit the file to your liking, reference the `doxygen manual` if needed, then run doxygen:
```
doxygen <config-file>
```

When releasing a new version of MacaroniOS, remember to change the `PROJECT_NUMER` (to R1, R2, etc) and `OUTPUT_DIRECTORY` (from `dev/doxygen` to `doc`). Also remember to change `user/version.c`.

Chapter 2

File Index

2.1 File List

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

include/comhand.h
: Command handler interface for the OS. Reads from the polling input and executes commands 5

include/ctype.h
: A subset of standard C library functions . . . 6

include/exit.h
: Header file for the exit command used in the command handler. Exits the terminal when called and confirmed by the user . . . 7

include/help.h
: Header for the help command used in command handler. Used to list the commands available to the user . . . 8

include/itoa.h
: Declaration for interger-to-ASCII conversion . . . 8

include/memory.h
: MPX-specific dynamic memory functions . . . 9

include/processes.h
: Provided system process and user processes for testing . . . 17

include/stdlib.h
: A subset of standard C library functions . . . 19

include/string.h
: A subset of standard C library functions . . . 19

include/sys_req.h
: System request function and constants . . . 22

include/version.h
: Displays the current version of MacaroniOS . . . 23

include/mpx/device.h . . . 9

include/mpx/gdt.h
: Kernel functions to initialize the Global Descriptor Table . . . 9

include/mpx/interrupts.h
: Kernel functions related to software and hardware interrupts . . . 10

include/mpx/io.h
: Kernel macros to read and write I/O ports . . . 12

include/mpx/panic.h
: Common system functions and definitions . . . 13

include/mpx/serial.h
: Kernel functions and constants for handling serial I/O . . . 14

include/mpx/vm.h
: Kernel functions for virtual memory and primitive allocation . . . 16

Chapter 3

File Documentation

3.1 include/comhand.h File Reference

Command handler interface for the OS. Reads from the polling input and executes commands.

Functions

- void **com_startup** (void)

  Prints a welcome message and penguin ASCII art to the terminal.
- void trim_Input (char ∗str)

  Trim function to remove \n and \r from the string.
- void **comhand** (void)

  Enters a loop and waits for the user to input commands.

3.1.1 Detailed Description

Command handler interface for the OS. Reads from the polling input and executes commands.

3.1.2 Function Documentation

3.1.2.1 trim_Input()

```
void trim_Input (
            char * str )
```

Trim function to remove \n and \r from the string.

Parameters

str	string variable to trim

3.2 comhand.h

Go to the documentation of this file.

```
00001 #ifndef COMHAND_H
00002 #define COMHAND_H
00003
00013 void com_startup(void);
00014
00019 void trim_Input(char *str);
00020
00024 void comhand(void);
00025
00026 #endif
```

3.3 include/ctype.h File Reference

A subset of standard C library functions.

Functions

- int isspace (int c)

3.3.1 Detailed Description

A subset of standard C library functions.

3.3.2 Function Documentation

3.3.2.1 isspace()

```
int isspace (
            int c )
```

Determine if a character is whitespace.

Parameters

c	Character to check

Returns

Non-zero if space, 0 if not space

3.4 ctype.h

Go to the documentation of this file.

```
00001 #ifndef MPX_CTYPE_H
00002 #define MPX_CTYPE_H
00003
00014 int isspace(int c);
00015
00016 #endif
```

3.5 include/exit.h File Reference

Header file for the exit command used in the command handler. Exits the terminal when called and confirmed by the user.

Functions

- void **exit_help** (void)
- int exit_command (const char *args)

  Begins the shutdown process when the user types 'exit' in the terminal. Confirmation by typing 'Y' or 'n' is then required to completely exit.

3.5.1 Detailed Description

Header file for the exit command used in the command handler. Exits the terminal when called and confirmed by the user.

Author

Caleb Edwards

3.5.2 Function Documentation

3.5.2.1 exit_command()

```
int exit_command (
            const char * args )
```

Begins the shutdown process when the user types 'exit' in the terminal. Confirmation by typing 'Y' or 'n' is then required to completely exit.

Parameters

arg_counter	Counts the number of arguments input.
arg_vector	Stores the arguments.

Returns

int return 1 to confirm exit and 0 to return to terminal.

3.6 exit.h

Go to the documentation of this file.

```
00001 #ifndef EXIT_H
00002 #define EXIT_H
00010 void exit_help(void);
00011
00020 int exit_command(const char *args);
00021
00022 #endif
```

3.7 include/help.h File Reference

Header for the help command used in command handler. Used to list the commands available to the user.

Functions

- void **help_message** (void)
- void **help_command** (const char ∗args)

  Prints all commands available into the terminal when the user types 'help' in the input.

3.7.1 Detailed Description

Header for the help command used in command handler. Used to list the commands available to the user.

Author

Caleb Edwards

3.8 help.h

Go to the documentation of this file.

```
00001 #ifndef HELP_H
00002 #define HELP_H
00010 void help_message(void);
00011
00016 void help_command(const char *args);
00017
00018 #endif
```

3.9 include/itoa.h File Reference

Declaration for interger-to-ASCII conversion.

Functions

- void itoa (int num, char ∗buffer)

  Converts an integer to a C-string.

3.9.1 Detailed Description

Declaration for interger-to-ASCII conversion.

3.9.2 Function Documentation

3.9.2.1 itoa()

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

Converts an integer to a C-string.

Parameters

num	The integer to convert.
buffer	Pointer to an array to store the string.

3.10 itoa.h

Go to the documentation of this file.

```
00001 #ifndef ITOA_H
00002 #define ITOA_H
00003
00015 void itoa(int num, char* buffer);
00016
00017 #endif
```

3.11 include/memory.h File Reference

MPX-specific dynamic memory functions.

`#include <stddef.h>`
Include dependency graph for memory.h:

3.12 memory.h

Go to the documentation of this file.

```
00001 #ifndef MPX_MEMORY_H
00002 #define MPX_MEMORY_H
00003
00004 #include <stddef.h>
00005
00016 void *sys_alloc_mem(size_t size);
00017
00023 int sys_free_mem(void *ptr);
00024
00030 void sys_set_heap_functions(void * (*alloc_fn)(size_t), int (*free_fn)(void *));
00031
00032 #endif
```

3.13 device.h

```
00001 #ifndef MPX_DEVICES_H
00002 #define MPX_DEVICES_H
00003
00004 typedef enum {
00005     COM1 = 0x3f8,
00006     COM2 = 0x2f8,
00007     COM3 = 0x3e8,
00008     COM4 = 0x2e8,
00009 } device;
00010
00011 #endif
```

3.14 include/mpx/gdt.h File Reference

Kernel functions to initialize the Global Descriptor Table.

Functions

- void gdt_init (void)

3.14.1 Detailed Description

Kernel functions to initialize the Global Descriptor Table.

3.14.2 Function Documentation

3.14.2.1 gdt_init()

```
void gdt_init (
            void  )
```

Creates and installs the Global Descriptor Table.

3.15 gdt.h

Go to the documentation of this file.

```
00001 #ifndef MPX_GDT_H
00002 #define MPX_GDT_H
00003
00010 void gdt_init(void);
00011
00012 #endif
```

3.16 include/mpx/interrupts.h File Reference

Kernel functions related to software and hardware interrupts.

Macros

- #define cli() __asm__ volatile ("cli")
- #define sti() __asm__ volatile ("sti")

Functions

- void irq_init (void)
- void pic_init (void)
- void idt_init (void)
- void idt_install (int vector, void(*handler)(void *))

3.16.1 Detailed Description

Kernel functions related to software and hardware interrupts.

3.16.2 Macro Definition Documentation

3.16.2.1 cli

```
#define cli( ) __asm__ volatile ("cli")
```

Disable interrupts

3.16.2.2 sti

```
#define sti( ) __asm__ volatile ("sti")
```

Enable interrupts

3.16.3 Function Documentation

3.16.3.1 idt_init()

```
void idt_init (
            void  )
```

Creates and installs the Interrupt Descriptor Table.

3.16.3.2 idt_install()

```
void idt_install (
            int vector,
            void(*)(void *) handler )
```

Installs an interrupt handler

3.16.3.3 irq_init()

```
void irq_init (
            void  )
```

Installs the initial interrupt handlers for the first 32 IRQ lines. Most do a panic for now.

3.16.3.4 pic_init()

```
void pic_init (
            void  )
```

Initializes the programmable interrupt controllers and performs the necessary remapping of IRQs. Leaves interrupts turned off.

3.17 interrupts.h

Go to the documentation of this file.

```
#ifndef MPX_INTERRUPTS_H
#define MPX_INTERRUPTS_H

#define cli() __asm__ volatile ("cli")

#define sti() __asm__ volatile ("sti")

void irq_init(void);

void pic_init(void);

void idt_init(void);

void idt_install(int vector, void (*handler)(void *));

#endif
```

3.18 include/mpx/io.h File Reference

Kernel macros to read and write I/O ports.

Macros

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

3.18.1 Detailed Description

Kernel macros to read and write I/O ports.

3.18.2 Macro Definition Documentation

3.18.2.1 inb

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

Value:

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

Read one byte from an I/O port

Parameters

port	The port to read from

Returns

A byte of data read from the port

3.18.2.2 outb

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

Write one byte to an I/O port

Parameters

port	The port to write to
data	The byte to write to the port

3.19 io.h

Go to the documentation of this file.

```
00001 #ifndef MPX_IO_H
00002 #define MPX_IO_H
00003
00014 #define outb(port, data)                              \
00015     __asm__ volatile ("outb %%al, %%dx" :: "a" (data), "d" (port))
00016
00022 #pragma clang diagnostic ignored "-Wgnu-statement-expression"
00023 #define inb(port) ({                                  \
00024       unsigned char r;                                \
00025       __asm__ volatile ("inb %%dx, %%al" : "=a" (r) : "d" (port));  \
00026       r;                                      \
00027     })
00028
00029 #endif
```

3.20 include/mpx/panic.h File Reference

Common system functions and definitions.

`#include <stdnoreturn.h>`
Include dependency graph for panic.h:

Functions

- noreturn __attribute__ ((no_caller_saved_registers)) void kpanic(const char *msg)

3.20.1 Detailed Description

Common system functions and definitions.

3.20.2 Function Documentation

3.20.2.1 __attribute__()

```
noreturn __attribute__ (
             (no_caller_saved_registers)  ) const
```

Kernel panic. Prints an error message and halts.

Parameters

msg	A message to display before halting

3.21 panic.h

Go to the documentation of this file.

```
00001 #ifndef MPX_PANIC_H
00002 #define MPX_PANIC_H
00003
00004 #include <stdnoreturn.h>
00005
00015 /*
00016  non-standard attribute is required for clang < 15
00017  */
00018 noreturn __attribute__((no_caller_saved_registers)) void kpanic(const char *msg);
00019
00020 #endif
```

3.22 include/mpx/serial.h File Reference

Kernel functions and constants for handling serial I/O.

```
#include <stddef.h>
#include <mpx/device.h>
```

Include dependency graph for serial.h:

Functions

- int serial_init (device dev)
- int serial_out (device dev, const char *buffer, size_t len)
- int serial_poll (device dev, char *buffer, size_t len)

3.22.1 Detailed Description

Kernel functions and constants for handling serial I/O.

3.22.2 Function Documentation

3.22.2.1 serial_init()

```
int serial_init (
            device dev )
```

Initializes devices for user input and output

Parameters

device	A serial port to initialize (COM1, COM2, COM3, or COM4)

Returns

0 on success, non-zero on failure

3.22.2.2 serial_out()

```
int serial_out (
            device dev,
            const char * buffer,
            size_t len )
```

Writes a buffer to a serial port

Parameters

device	The serial port to output to
buffer	A pointer to an array of characters to output
len	The number of bytes to write

Returns

The number of bytes written

3.22.2.3 serial_poll()

```
int serial_poll (
            device dev,
            char * buffer,
            size_t len )
```

Reads a string from a serial port

Parameters

device	The serial port to read data from
buffer	A buffer to write data into as it is read from the serial port
count	The maximum number of bytes to read

Returns

The number of bytes read on success, a negative number on failure

3.23 serial.h

Go to the documentation of this file.

```
00001 #ifndef MPX_SERIAL_H
00002 #define MPX_SERIAL_H
00003
00004 #include <stddef.h>
00005 #include <mpx/device.h>
00006
00017 int serial_init(device dev);
00018
00026 int serial_out(device dev, const char *buffer, size_t len);
00027
00035 int serial_poll(device dev, char *buffer, size_t len);
00036
00037 #endif
```

3.24 include/mpx/vm.h File Reference

Kernel functions for virtual memory and primitive allocation.

`#include <stddef.h>`
Include dependency graph for vm.h:

Functions

- void * kmalloc (size_t size, int align, void **phys_addr)
- void vm_init (void)

3.24.1 Detailed Description

Kernel functions for virtual memory and primitive allocation.

3.24.2 Function Documentation

3.24.2.1 kmalloc()

```
void * kmalloc (
            size_t size,
            int align,
            void ** phys_addr )
```

Allocates memory from a primitive heap.

Parameters

size	The size of memory to allocate
align	If non-zero, align the allocation to a page boundary
phys_addr	If non-NULL, a pointer to a pointer that will hold the physical address of the new memory

Returns

The newly allocated memory

3.24.2.2 vm_init()

```
void vm_init (
              void  )
```

Initializes the kernel page directory and initial kernel heap area. Performs identity mapping of the kernel frames such that the virtual addresses are equivalent to the physical addresses.

3.25 vm.h

Go to the documentation of this file.

```
00001 #ifndef MPX_VM_H
00002 #define MPX_VM_H
00003
00009 #include <stddef.h>
00010
00019 void *kmalloc(size_t size, int align, void **phys_addr);
00020
00026 void vm_init(void);
00027
00028 #endif
```

3.26 include/processes.h File Reference

Provided system process and user processes for testing.

Functions

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

3.26.1 Detailed Description

Provided system process and user processes for testing.

3.26.2 Function Documentation

3.26.2.1 proc1()

```
void proc1 (
              void  )
```

A test process that prints a message then yields, exiting after 1 iteration.

3.26.2.2 proc2()

```
void proc2 (
            void  )
```

A test process that prints a message then yields, exiting after 2 iterations.

3.26.2.3 proc3()

```
void proc3 (
            void  )
```

A test process that prints a message then yields, exiting after 3 iterations.

3.26.2.4 proc4()

```
void proc4 (
            void  )
```

A test process that prints a message then yields, exiting after 4 iterations.

3.26.2.5 proc5()

```
void proc5 (
            void  )
```

A test process that prints a message then yields, exiting after 5 iterations.

3.26.2.6 sys_idle_process()

```
void sys_idle_process (
            void  )
```

System idle process. Used in dispatching. It will be dispatched if NO other processes are available to execute. Must be a system process.

3.27 processes.h

Go to the documentation of this file.

```
00001 #ifndef MPX_PROCESSES_H
00002 #define MPX_PROCESSES_H
00003
00009 /* ************************************************************************
00010  The following functions are needed for Module R3.
00011 ************************************************************************ */
00012
00016 void proc1(void);
00017
00021 void proc2(void);
00022
00026 void proc3(void);
00027
00031 void proc4(void);
00032
00036 void proc5(void);
00037
00038 /* ************************************************************************
00039  The following function is needed for Module R4.
00040 ************************************************************************ */
00041
00046 void sys_idle_process(void);
00047
00048 #endif
```

3.28 include/stdlib.h File Reference

A subset of standard C library functions.

Functions

- int atoi (const char *s)

3.28.1 Detailed Description

A subset of standard C library functions.

3.28.2 Function Documentation

3.28.2.1 atoi()

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

Convert an ASCII string to an integer

Parameters

s	A NUL-terminated string

Returns

The value of the string converted to an integer

3.29 stdlib.h

Go to the documentation of this file.

```
00001 #ifndef MPX_STDLIB_H
00002 #define MPX_STDLIB_H
00003
00014 int atoi(const char *s);
00015
00016 #endif
```

3.30 include/string.h File Reference

A subset of standard C library functions.

`#include <stddef.h>`
Include dependency graph for string.h:

Functions

- void * memcpy (void *restrict dst, const void *restrict src, size_t n)
- void * memset (void *address, int c, size_t n)
- int strcmp (const char *s1, const char *s2)
- int **strncmp** (const char *s1, const char *s2, unsigned int n)
- size_t strlen (const char *s)
- char * strtok (char *restrict s1, const char *restrict s2)

3.30.1 Detailed Description

A subset of standard C library functions.

3.30.2 Function Documentation

3.30.2.1 memcpy()

```
void * memcpy (
            void *restrict dst,
            const void *restrict src,
            size_t n )
```

Copy a region of memory.

Parameters

dst	The destination memory region
src	The source memory region
n	The number of bytes to copy

Returns

A pointer to the destination memory region

3.30.2.2 memset()

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

Fill a region of memory.

Parameters

address	The start of the memory region
c	The byte to fill memory with
n	The number of bytes to fill

Returns

A pointer to the filled memory region

3.30.2.3 strcmp()

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

Compares two strings

Parameters

s1	The first string to compare
s2	The second string to compare

Returns

0 if strings are equal, <0 if s1 is lexicographically before s2, >0 otherwise

3.30.2.4 strlen()

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

Returns the length of a string.

Parameters

s	A NUL-terminated string

Returns

The number of bytes in the string (not counting NUL terminator)

3.30.2.5 strtok()

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

Split string into tokens TODO

3.31 string.h

Go to the documentation of this file.

```
00001 #ifndef MPX_STRING_H
00002 #define MPX_STRING_H
00003
00004 #include <stddef.h>
00005
00018 void* memcpy(void * restrict dst, const void * restrict src, size_t n);
00019
00027 void* memset(void *address, int c, size_t n);
00028
00035 int strcmp(const char *s1, const char *s2);
00036
00037 int strncmp(const char *s1, const char *s2, unsigned int n);
00038
00044 size_t strlen(const char *s);
00045
00050 char* strtok(char * restrict s1, const char * restrict s2);
00051
00052 #endif
```

3.32 include/sys_req.h File Reference

System request function and constants.

`#include <mpx/device.h>`
Include dependency graph for sys_req.h:

Macros

- #define **INVALID_OPERATION** (-1)
- #define **INVALID_BUFFER** (-2)
- #define **INVALID_COUNT** (-3)

Enumerations

- enum **op_code** { **EXIT** , **IDLE** , **READ** , **WRITE** }

Functions

- int sys_req (op_code op,...)

3.32.1 Detailed Description

System request function and constants.

3.32.2 Function Documentation

3.32.2.1 sys_req()

```
int sys_req (
            op_code op,
             ...  )
```

Request an MPX kernel operation.

Parameters

op_code	One of READ, WRITE, IDLE, or EXIT
...	As required for READ or WRITE

Returns

Varies by operation

3.33 sys_req.h

Go to the documentation of this file.

```
00001 #ifndef MPX_SYS_REQ_H
00002 #define MPX_SYS_REQ_H
00003
00004 #include <mpx/device.h>
00005
00011 typedef enum {
00012     EXIT,
00013     IDLE,
00014     READ,
00015     WRITE,
00016 } op_code;
00017
00018 // error codes
00019 #define INVALID_OPERATION   (-1)
00020 #define INVALID_BUFFER      (-2)
00021 #define INVALID_COUNT       (-3)
00022
00029 int sys_req(op_code op, ...);
00030
00031 #endif
```

3.34 include/version.h File Reference

Displays the current version of MacaroniOS.

Macros

- #define **GIT_DATE** "unknown"
- #define **GIT_HASH** "unknown"
- #define **GIT_DIRTY** "unknown"

Functions

- void **version_help** (void)

  Prints help information related to the version command.
- void **version_latest** (void)

  Displays the latest version.
- void **version_history** (void)

  Displays the past and present versions.
- void version_command (const char ∗args)

  Main handler for the version command.

3.34.1 Detailed Description

Displays the current version of MacaroniOS.

3.34.2 Function Documentation

3.34.2.1 version_command()

```
void version_command (
            const char * args )
```

Main handler for the version command.

Parameters

args	The argument string passed after 'version'

3.35 version.h

Go to the documentation of this file.

```
00001 #ifndef VERSION_H
00002 #define VERSION_H
00003
00004 #ifndef GIT_DATE
00005 #define GIT_DATE "unknown"
00006 #endif
00007
00008 #ifndef GIT_HASH
00009 #define GIT_HASH "unknown"
00010 #endif
00011
00012 #ifndef GIT_DIRTY
00013 #define GIT_DIRTY "unknown"
00014 #endif
00015
00025 void version_help(void);
00026
00030 void version_latest(void);
00031
00035 void version_history(void);
00036
00041 void version_command(const char *args);
00042
00043 #endif
```

Index

__attribute__
    panic.h, 14

atoi
    stdlib.h, 19

cli
    interrupts.h, 11
comhand.h
    trim_Input, 5
ctype.h
    isspace, 6

exit.h
    exit_command, 7
exit_command
    exit.h, 7

gdt.h
    gdt_init, 10
gdt_init
    gdt.h, 10

idt_init
    interrupts.h, 11
idt_install
    interrupts.h, 11
inb
    io.h, 12
include/comhand.h, 5, 6
include/ctype.h, 6
include/exit.h, 7
include/help.h, 8
include/itoa.h, 8, 9
include/memory.h, 9
include/mpx/device.h, 9
include/mpx/gdt.h, 9, 10
include/mpx/interrupts.h, 10, 12
include/mpx/io.h, 12, 13
include/mpx/panic.h, 13, 14
include/mpx/serial.h, 14, 16
include/mpx/vm.h, 16, 17
include/processes.h, 17, 18
include/stdlib.h, 19
include/string.h, 19, 22
include/sys_req.h, 22, 23
include/version.h, 23, 24
interrupts.h
    cli, 11
    idt_init, 11
    idt_install, 11
    irq_init, 11
    pic_init, 11
    sti, 11
io.h
    inb, 12
    outb, 13
irq_init
    interrupts.h, 11
isspace
    ctype.h, 6
itoa
    itoa.h, 8
itoa.h
    itoa, 8

kmalloc
    vm.h, 16

Macaroni Penguins, 1
memcpy
    string.h, 20
memset
    string.h, 20

outb
    io.h, 13

panic.h
    __attribute__, 14
pic_init
    interrupts.h, 11
proc1
    processes.h, 17
proc2
    processes.h, 17
proc3
    processes.h, 18
proc4
    processes.h, 18
proc5
    processes.h, 18
processes.h
    proc1, 17
    proc2, 17
    proc3, 18
    proc4, 18
    proc5, 18
    sys_idle_process, 18

serial.h
    serial_init, 14

serial_out, 15
serial_poll, 15
serial_init
serial.h, 14
serial_out
serial.h, 15
serial_poll
serial.h, 15
stdlib.h
atoi, 19
sti
interrupts.h, 11
strcmp
string.h, 21
string.h
memcpy, 20
memset, 20
strcmp, 21
strlen, 21
strtok, 21
strlen
string.h, 21
strtok
string.h, 21
sys_idle_process
processes.h, 18
sys_req
sys_req.h, 22
sys_req.h
sys_req, 22

trim_Input
comhand.h, 5

version.h
version_command, 24
version_command
version.h, 24
vm.h
kmalloc, 16
vm_init, 17
vm_init
vm.h, 17