

MAMA

Generated by Doxygen 1.9.3

1 MAMA	1
2 Who did what table	3
3 Class Index	5
3.1 Class List	5
4 File Index	7
4.1 File List	7
5 Class Documentation	9
5.1 cmd_mapping Struct Reference	9
5.1.1 Member Data Documentation	9
5.1.1.1 cmd_handler	9
5.1.1.2 cmd_name	9
5.2 date_time Struct Reference	9
5.2.1 Member Data Documentation	10
5.2.1.1 day_m	10
5.2.1.2 day_w	10
5.2.1.3 day_y	10
5.2.1.4 hour	10
5.2.1.5 min	10
5.2.1.6 mon	11
5.2.1.7 sec	11
5.2.1.8 year	11
5.3 footer Struct Reference	11
5.3.1 Member Data Documentation	11
5.3.1.1 head	11
5.4 gdt_descriptor_struct Struct Reference	11
5.4.1 Member Data Documentation	12
5.4.1.1 base	12
5.4.1.2 limit	12
5.5 gdt_entry_struct Struct Reference	12
5.5.1 Member Data Documentation	12
5.5.1.1 access	12
5.5.1.2 base_high	13
5.5.1.3 base_low	13
5.5.1.4 base_mid	13
5.5.1.5 flags	13
5.5.1.6 limit_low	13
5.6 header Struct Reference	13
5.6.1 Member Data Documentation	13
5.6.1.1 index_id	14
5.6.1.2 size	14

5.7 heap Struct Reference	14
5.7.1 Member Data Documentation	14
5.7.1.1 base	14
5.7.1.2 index	14
5.7.1.3 max_size	15
5.7.1.4 min_size	15
5.8 idt_entry_struct Struct Reference	15
5.8.1 Member Data Documentation	15
5.8.1.1 base_high	15
5.8.1.2 base_low	15
5.8.1.3 flags	16
5.8.1.4 sselect	16
5.8.1.5 zero	16
5.9 idt_struct Struct Reference	16
5.9.1 Member Data Documentation	16
5.9.1.1 base	16
5.9.1.2 limit	16
5.10 index_entry Struct Reference	17
5.10.1 Member Data Documentation	17
5.10.1.1 block	17
5.10.1.2 empty	17
5.10.1.3 size	17
5.11 index_table Struct Reference	17
5.11.1 Member Data Documentation	18
5.11.1.1 id	18
5.11.1.2 table	18
5.12 page_dir Struct Reference	18
5.12.1 Member Data Documentation	18
5.12.1.1 tables	18
5.12.1.2 tables_phys	18
5.13 page_entry Struct Reference	19
5.13.1 Member Data Documentation	19
5.13.1.1 accessed	19
5.13.1.2 dirty	19
5.13.1.3 frameaddr	19
5.13.1.4 present	19
5.13.1.5 reserved	19
5.13.1.6 usermode	20
5.13.1.7 writeable	20
5.14 page_table Struct Reference	20
5.14.1 Member Data Documentation	20
5.14.1.1 pages	20

5.15 param Struct Reference	20
5.15.1 Member Data Documentation	21
5.15.1.1 buffer_ptr	21
5.15.1.2 count_ptr	21
5.15.1.3 device_id	21
5.15.1.4 op_code	21
5.16 parsed_args Struct Reference	21
5.16.1 Member Data Documentation	22
5.16.1.1 flag_count	22
5.16.1.2 flags	22
5.16.1.3 named_arg_count	22
5.16.1.4 named_arg_names	22
5.16.1.5 named_arg_values	22
5.16.1.6 unnamed_arg_count	22
5.16.1.7 unnamed_args	22
5.16.1.8 unnamed_args_used_so_far	23
5.17 pcb_node_t Struct Reference	23
5.17.1 Detailed Description	23
5.17.2 Member Data Documentation	23
5.17.2.1 pcb	23
5.17.2.2 pcbn_next_pcb	23
5.17.2.3 pcbn_prev_pcb	24
5.18 pcb_queue Struct Reference	24
5.18.1 Detailed Description	24
5.18.2 Member Data Documentation	24
5.18.2.1 pcbq_count	24
5.18.2.2 pcbq_head	25
5.18.2.3 pcbq_tail	25
5.18.2.4 queue_order	25
5.19 pcb_t Struct Reference	25
5.19.1 Detailed Description	26
5.19.2 Member Data Documentation	26
5.19.2.1 pcb_name	26
5.19.2.2 pcb_priority	26
5.19.2.3 pcb_process_class	26
5.19.2.4 pcb_process_state	26
5.19.2.5 pcb_stack_bottom	26
5.19.2.6 pcb_stack_top	26
6 File Documentation	27
6.1 /home/maximillian/Desktop/MAMA/include/core/asm.h File Reference	27
6.2 asm.h	27

6.3 /home/maximillian/Desktop/MAMA/include/core/comhand.h File Reference	27
6.3.1 Function Documentation	27
6.3.1.1 comhand()	27
6.4 comhand.h	28
6.5 /home/maximillian/Desktop/MAMA/include/core/interrupts.h File Reference	28
6.5.1 Function Documentation	28
6.5.1.1 init_irq()	28
6.5.1.2 init_pic()	28
6.6 interrupts.h	28
6.7 /home/maximillian/Desktop/MAMA/include/core/io.h File Reference	29
6.7.1 Macro Definition Documentation	29
6.7.1.1 inb	29
6.7.1.2 outb	29
6.8 io.h	29
6.9 /home/maximillian/Desktop/MAMA/include/core/serial.h File Reference	30
6.9.1 Macro Definition Documentation	30
6.9.1.1 COM1	30
6.9.1.2 COM2	30
6.9.1.3 COM3	30
6.9.1.4 COM4	30
6.9.2 Function Documentation	31
6.9.2.1 init_serial()	31
6.9.2.2 polling()	31
6.9.2.3 serial_print()	31
6.9.2.4 serial_println()	31
6.9.2.5 set_serial_in()	32
6.9.2.6 set_serial_out()	32
6.10 serial.h	32
6.11 /home/maximillian/Desktop/MAMA/include/core/tables.h File Reference	33
6.11.1 Function Documentation	33
6.11.1.1 __attribute__()	33
6.11.1.2 gdt_init_entry()	34
6.11.1.3 idt_set_gate()	34
6.11.1.4 init_gdt()	34
6.11.1.5 init_idt()	34
6.11.2 Variable Documentation	34
6.11.2.1 access	34
6.11.2.2 base	34
6.11.2.3 base_high	35
6.11.2.4 base_low	35
6.11.2.5 base_mid	35
6.11.2.6 flags	35

6.11.2.7 limit	35
6.11.2.8 limit_low	35
6.11.2.9 sselect	35
6.11.2.10 zero	35
6.12 tables.h	36
6.13 /home/maximillian/Desktop/MAMA/include/mem/heap.h File Reference	36
6.13.1 Macro Definition Documentation	37
6.13.1.1 KHEAP_BASE	37
6.13.1.2 KHEAP_MIN	37
6.13.1.3 KHEAP_SIZE	37
6.13.1.4 TABLE_SIZE	37
6.13.2 Function Documentation	37
6.13.2.1 _kmalloc()	37
6.13.2.2 alloc()	38
6.13.2.3 init_kheap()	38
6.13.2.4 kfree()	38
6.13.2.5 kmalloc()	38
6.13.2.6 make_heap()	38
6.14 heap.h	39
6.15 /home/maximillian/Desktop/MAMA/include/mem/paging.h File Reference	40
6.15.1 Macro Definition Documentation	40
6.15.1.1 PAGE_SIZE	40
6.15.2 Function Documentation	40
6.15.2.1 clear_bit()	40
6.15.2.2 first_free()	41
6.15.2.3 get_bit()	41
6.15.2.4 get_page()	41
6.15.2.5 init_paging()	41
6.15.2.6 load_page_dir()	41
6.15.2.7 new_frame()	41
6.15.2.8 set_bit()	41
6.16 paging.h	42
6.17 /home/maximillian/Desktop/MAMA/include/string.h File Reference	43
6.17.1 Function Documentation	43
6.17.1.1 atoi()	43
6.17.1.2 isspace()	43
6.17.1.3 itoa()	43
6.17.1.4 memset()	44
6.17.1.5 strcat()	44
6.17.1.6 strcmp()	44
6.17.1.7 strcpy()	44
6.17.1.8 strlen()	44

6.17.1.9 strtok()	45
6.18 string.h	45
6.19 /home/maximillian/Desktop/MAMA/include/system.h File Reference	46
6.19.1 Macro Definition Documentation	46
6.19.1.1 asm	46
6.19.1.2 cli	46
6.19.1.3 GDT_CS_ID	47
6.19.1.4 GDT_DS_ID	47
6.19.1.5 hlt	47
6.19.1.6 iret	47
6.19.1.7 no_warn	47
6.19.1.8 nop	47
6.19.1.9 NULL	47
6.19.1.10 sti	48
6.19.1.11 volatile	48
6.19.2 Typedef Documentation	48
6.19.2.1 size_t	48
6.19.2.2 u16int	48
6.19.2.3 u32int	48
6.19.2.4 u8int	48
6.19.3 Function Documentation	48
6.19.3.1 klogv()	48
6.19.3.2 kpanic()	49
6.20 system.h	49
6.21 /home/maximillian/Desktop/MAMA/kernel/core/interrupts.c File Reference	49
6.21.1 Macro Definition Documentation	51
6.21.1.1 ICW1	51
6.21.1.2 ICW4	51
6.21.1.3 io_wait	51
6.21.1.4 PIC1	51
6.21.1.5 PIC2	51
6.21.2 Function Documentation	51
6.21.2.1 bounds()	51
6.21.2.2 breakpoint()	52
6.21.2.3 coprocessor()	52
6.21.2.4 coprocessor_segment()	52
6.21.2.5 debug()	52
6.21.2.6 device_not_available()	52
6.21.2.7 divide_error()	52
6.21.2.8 do_bounds()	52
6.21.2.9 do_breakpoint()	52
6.21.2.10 do_coprocessor()	53

6.21.2.11 do_coprocessor_segment()	53
6.21.2.12 do_debug()	53
6.21.2.13 do_device_not_available()	53
6.21.2.14 do_divide_error()	53
6.21.2.15 do_double_fault()	53
6.21.2.16 do_general_protection()	53
6.21.2.17 do_invalid_op()	53
6.21.2.18 do_invalid_tss()	54
6.21.2.19 do_isr()	54
6.21.2.20 do_nmi()	54
6.21.2.21 do_overflow()	54
6.21.2.22 do_page_fault()	54
6.21.2.23 do_reserved()	54
6.21.2.24 do_segment_not_present()	54
6.21.2.25 do_stack_segment()	54
6.21.2.26 double_fault()	55
6.21.2.27 general_protection()	55
6.21.2.28 init_irq()	55
6.21.2.29 init_pic()	55
6.21.2.30 invalid_op()	55
6.21.2.31 invalid_tss()	55
6.21.2.32 isr0()	55
6.21.2.33 nmi()	56
6.21.2.34 overflow()	56
6.21.2.35 page_fault()	56
6.21.2.36 reserved()	56
6.21.2.37 rtc_isr()	56
6.21.2.38 segment_not_present()	56
6.21.2.39 stack_segment()	56
6.21.3 Variable Documentation	56
6.21.3.1 idt_entries	57
6.22 /home/maximillian/Desktop/MAMA/kernel/core/kmain.c File Reference	57
6.22.1 Function Documentation	57
6.22.1.1 kmain()	57
6.23 /home/maximillian/Desktop/MAMA/kernel/core/serial.c File Reference	57
6.23.1 Macro Definition Documentation	58
6.23.1.1 DELETE	58
6.23.1.2 DOWN_ARROW	58
6.23.1.3 LEFT_ARROW	58
6.23.1.4 NO_ERROR	59
6.23.1.5 RIGHT_ARROW	59
6.23.1.6 UP_ARROW	59

6.23.2 Function Documentation	59
6.23.2.1 consume_special()	59
6.23.2.2 init_serial()	59
6.23.2.3 polling()	59
6.23.2.4 serial_print()	60
6.23.2.5 serial_println()	60
6.23.2.6 set_serial_in()	60
6.23.2.7 set_serial_out()	60
6.23.3 Variable Documentation	60
6.23.3.1 serial_port_in	60
6.23.3.2 serial_port_out	60
6.24 /home/maximillian/Desktop/MAMA/kernel/core/system.c File Reference	61
6.24.1 Function Documentation	61
6.24.1.1 klogv()	61
6.24.1.2 kpanic()	61
6.25 /home/maximillian/Desktop/MAMA/kernel/core/tables.c File Reference	61
6.25.1 Function Documentation	62
6.25.1.1 gdt_init_entry()	62
6.25.1.2 idt_set_gate()	62
6.25.1.3 init_gdt()	62
6.25.1.4 init_idt()	62
6.25.1.5 write_gdt_ptr()	63
6.25.1.6 write_idt_ptr()	63
6.25.2 Variable Documentation	63
6.25.2.1 gdt_entries	63
6.25.2.2 gdt_ptr	63
6.25.2.3 idt_entries	63
6.25.2.4 idt_ptr	63
6.26 /home/maximillian/Desktop/MAMA/kernel/mem/heap.c File Reference	63
6.26.1 Function Documentation	64
6.26.1.1 _kmalloc()	64
6.26.1.2 alloc()	64
6.26.1.3 kmalloc()	64
6.26.1.4 make_heap()	65
6.26.2 Variable Documentation	65
6.26.2.1 __end	65
6.26.2.2 _end	65
6.26.2.3 curr_heap	65
6.26.2.4 end	65
6.26.2.5 kdir	65
6.26.2.6 kheap	65
6.26.2.7 phys_alloc_addr	66

6.27 /home/maximillian/Desktop/MAMA/kernel/mem/paging.c File Reference	66
6.27.1 Function Documentation	66
6.27.1.1 clear_bit()	66
6.27.1.2 find_free()	67
6.27.1.3 get_bit()	67
6.27.1.4 get_page()	67
6.27.1.5 init_paging()	67
6.27.1.6 load_page_dir()	67
6.27.1.7 new_frame()	67
6.27.1.8 set_bit()	67
6.27.2 Variable Documentation	68
6.27.2.1 cdir	68
6.27.2.2 frames	68
6.27.2.3 kdir	68
6.27.2.4 kheap	68
6.27.2.5 mem_size	68
6.27.2.6 nframes	68
6.27.2.7 page_size	68
6.27.2.8 phys_alloc_addr	69
6.28 /home/maximillian/Desktop/MAMA/lib/out.c File Reference	69
6.28.1 Function Documentation	69
6.28.1.1 print()	69
6.28.1.2 printc()	69
6.28.1.3 printf()	69
6.28.1.4 println()	70
6.28.1.5 read()	70
6.29 /home/maximillian/Desktop/MAMA/lib/out.h File Reference	70
6.29.1 Function Documentation	71
6.29.1.1 blockHelp()	71
6.29.1.2 cmd_help()	71
6.29.1.3 createpcbHelp()	72
6.29.1.4 deletepcbHelp()	72
6.29.1.5 getdateHelp()	72
6.29.1.6 gettimeHelp()	72
6.29.1.7 helpHelp()	72
6.29.1.8 helpList()	72
6.29.1.9 print()	73
6.29.1.10 printc()	73
6.29.1.11 printf()	73
6.29.1.12 println()	73
6.29.1.13 read()	73
6.29.1.14 resumeHelp()	73

6.29.1.15 setdateHelp()	74
6.29.1.16 setpriorityHelp()	74
6.29.1.17 settimeHelp()	74
6.29.1.18 showallpcbHelp()	74
6.29.1.19 showblockedpcbHelp()	74
6.29.1.20 showpcbHelp()	74
6.29.1.21 showreadypcbHelp()	75
6.29.1.22 shutdownHelp()	75
6.29.1.23 suspendHelp()	75
6.29.1.24 unblockHelp()	75
6.29.1.25 versionHelp()	75
6.30 out.h	76
6.31 /home/maximillian/Desktop/MAMA/lib/string.c File Reference	76
6.31.1 Function Documentation	77
6.31.1.1 atoi()	77
6.31.1.2 isspace()	77
6.31.1.3 itoa()	77
6.31.1.4 memset()	77
6.31.1.5 strcat()	78
6.31.1.6 strcmp()	78
6.31.1.7 strcpy()	78
6.31.1.8 strlen()	78
6.31.1.9 strtok()	78
6.32 /home/maximillian/Desktop/MAMA/modules/mpx_supt.c File Reference	78
6.32.1 Function Documentation	79
6.32.1.1 idle()	79
6.32.1.2 mpx_init()	79
6.32.1.3 sys_alloc_mem()	79
6.32.1.4 sys_free_mem()	79
6.32.1.5 sys_req()	80
6.32.1.6 sys_set_free()	80
6.32.1.7 sys_set_malloc()	80
6.32.2 Variable Documentation	80
6.32.2.1 current_module	80
6.32.2.2 params	80
6.32.2.3 student_free	80
6.32.2.4 student_malloc	81
6.33 /home/maximillian/Desktop/MAMA/modules/mpx_supt.h File Reference	81
6.33.1 Macro Definition Documentation	82
6.33.1.1 COM_PORT	82
6.33.1.2 DEFAULT_DEVICE	82
6.33.1.3 EXIT	82

6.33.1.4 FALSE	82
6.33.1.5 IDLE	82
6.33.1.6 INVALID_BUFFER	82
6.33.1.7 INVALID_COUNT	82
6.33.1.8 INVALID_OPERATION	83
6.33.1.9 IO_MODULE	83
6.33.1.10 MEM_MODULE	83
6.33.1.11 MODULE_F	83
6.33.1.12 MODULE_R1	83
6.33.1.13 MODULE_R2	83
6.33.1.14 MODULE_R3	83
6.33.1.15 MODULE_R4	83
6.33.1.16 MODULE_R5	84
6.33.1.17 READ	84
6.33.1.18 TRUE	84
6.33.1.19 WRITE	84
6.33.2 Function Documentation	84
6.33.2.1 idle()	84
6.33.2.2 mpx_init()	84
6.33.2.3 sys_alloc_mem()	84
6.33.2.4 sys_free_mem()	85
6.33.2.5 sys_req()	85
6.33.2.6 sys_set_free()	85
6.33.2.7 sys_set_malloc()	85
6.34 mpx_supt.h	85
6.35 /home/maximillian/Desktop/MAMA/README.md File Reference	86
6.36 /home/maximillian/Desktop/MAMA/term/args.c File Reference	86
6.36.1 Macro Definition Documentation	87
6.36.1.1 MAX_PARSE_STACK_SIZE	87
6.36.2 Function Documentation	87
6.36.2.1 flag()	87
6.36.2.2 get_token()	88
6.36.2.3 named_arg()	88
6.36.2.4 next_unnamed_arg()	88
6.36.2.5 parse_args()	88
6.36.2.6 stack_empty()	88
6.36.2.7 stack_peek()	88
6.36.2.8 stack_pop()	89
6.36.2.9 stack_push()	89
6.36.3 Variable Documentation	89
6.36.3.1 cur_state	89
6.36.3.2 last_state	89

6.36.3.3	parse_stack	89
6.36.3.4	stack_size	89
6.37	/home/maximillian/Desktop/MAMA/term/args.h File Reference	89
6.37.1	Typedef Documentation	90
6.37.1.1	parsed_args	90
6.37.2	Function Documentation	90
6.37.2.1	parse_args()	90
6.38	args.h	90
6.39	/home/maximillian/Desktop/MAMA/term/ascii/mama.c File Reference	90
6.39.1	Function Documentation	91
6.39.1.1	mama()	91
6.40	/home/maximillian/Desktop/MAMA/term/ascii/mama.h File Reference	91
6.40.1	Function Documentation	91
6.40.1.1	mama()	91
6.41	mama.h	91
6.42	/home/maximillian/Desktop/MAMA/term/cmds/argtest.c File Reference	92
6.42.1	Function Documentation	92
6.42.1.1	cmd_argtest()	92
6.43	/home/maximillian/Desktop/MAMA/term/cmds/echo.c File Reference	92
6.43.1	Function Documentation	92
6.43.1.1	cmd_echo()	92
6.44	/home/maximillian/Desktop/MAMA/help.c File Reference	92
6.44.1	Function Documentation	93
6.44.1.1	cmd_help()	93
6.44.1.2	getdateHelp()	93
6.44.1.3	gettimeHelp()	94
6.44.1.4	helpHelp()	94
6.44.1.5	helpList()	94
6.44.1.6	setdateHelp()	94
6.44.1.7	settimeHelp()	94
6.44.1.8	shutdownHelp()	94
6.44.1.9	versionOs()	95
6.45	/home/maximillian/Desktop/MAMA/term/cmds/help.c File Reference	95
6.45.1	Function Documentation	96
6.45.1.1	blockHelp()	96
6.45.1.2	cmd_help()	96
6.45.1.3	createpcbHelp()	96
6.45.1.4	deletepcbHelp()	96
6.45.1.5	getdateHelp()	97
6.45.1.6	gettimeHelp()	97
6.45.1.7	helpHelp()	97
6.45.1.8	helpList()	97

6.45.1.9 resumeHelp()	97
6.45.1.10 setdateHelp()	97
6.45.1.11 setpriorityHelp()	98
6.45.1.12 settimeHelp()	98
6.45.1.13 showallpcbHelp()	98
6.45.1.14 showblockedpcbHelp()	98
6.45.1.15 showpcbHelp()	98
6.45.1.16 showreadypcbHelp()	98
6.45.1.17 shutdownHelp()	99
6.45.1.18 suspendHelp()	99
6.45.1.19 unblockHelp()	99
6.45.1.20 versionHelp()	99
6.46 /home/maximillian/Desktop/MAMA/term/cmds/shutdown.c File Reference	99
6.46.1 Function Documentation	99
6.46.1.1 cmd_shutdown()	99
6.47 /home/maximillian/Desktop/MAMA/term/cmds/version.c File Reference	100
6.47.1 Function Documentation	100
6.47.1.1 cmd_version()	100
6.48 /home/maximillian/Desktop/MAMA/term/commands.h File Reference	100
6.49 commands.h	101
6.50 /home/maximillian/Desktop/MAMA/term/commhand.c File Reference	101
6.50.1 Typedef Documentation	102
6.50.1.1 cmd_func_t	102
6.50.1.2 cmd_mapping	102
6.50.2 Function Documentation	102
6.50.2.1 commhand()	102
6.50.2.2 extract_cmd_name()	103
6.50.2.3 fetch_cmd_handler()	103
6.50.2.4 is_name_char()	103
6.50.3 Variable Documentation	103
6.50.3.1 cmd_mappings	103
6.51 /home/maximillian/Desktop/MAMA/term/commhand.h File Reference	103
6.51.1 Macro Definition Documentation	104
6.51.1.1 MAX_CMD_ARG_NAME_LEN	104
6.51.1.2 MAX_CMD_ARG_VALUE_LEN	104
6.51.1.3 MAX_CMD_FLAG_COUNT	104
6.51.1.4 MAX_CMD_HIST_LEN	104
6.51.1.5 MAX_CMD_NAME_LEN	104
6.51.1.6 MAX_CMD_NAMED_ARG_COUNT	105
6.51.1.7 MAX_CMD_STRING_LEN	105
6.51.1.8 MAX_CMD_UNNAMED_ARG_COUNT	105
6.51.2 Function Documentation	105

6.51.2.1 commhand()	105
6.52 commhand.h	105
6.53 /home/maximillian/Desktop/MAMA/term/dnt/dnt.c File Reference	106
6.53.1 Function Documentation	106
6.53.1.1 BCDtol()	106
6.53.1.2 daysInMonth()	107
6.53.1.3 getdate()	107
6.53.1.4 gettime()	108
6.53.1.5 intToDayOfWeek()	108
6.53.1.6 intToMonth()	108
6.53.1.7 ltoBCD()	109
6.53.1.8 setdate()	109
6.53.1.9 setDateInMemory()	110
6.53.1.10 settime()	110
6.53.1.11 setTimeInMemory()	110
6.54 /home/maximillian/Desktop/MAMA/term/dnt/dnt.h File Reference	111
6.54.1 Macro Definition Documentation	112
6.54.1.1 DAYS_IN_LEAP_YEAR	112
6.54.1.2 DAYS_IN_YEAR	112
6.54.1.3 EPOCH_FIRST_DAY_OF_WEEK_OF_YEAR	113
6.54.1.4 EPOCH_FIRST_DAY_OF_YEAR	113
6.54.1.5 EPOCH_FIRST_MONTH_OF_YEAR	113
6.54.1.6 EPOCH_YEAR	113
6.54.1.7 MAX_DAY	113
6.54.1.8 MAX_HOURS	113
6.54.1.9 MAX_MINUTES	114
6.54.1.10 MAX_MONTH	114
6.54.1.11 MAX_SECONDS	114
6.54.1.12 MAX_YEAR	114
6.54.1.13 MIN	114
6.54.1.14 MIN_DAY	114
6.54.1.15 MIN_MONTH	115
6.54.1.16 MIN_YEAR	115
6.54.2 Function Documentation	115
6.54.2.1 BCDtol()	115
6.54.2.2 daysInMonth()	115
6.54.2.3 getdate()	116
6.54.2.4 gettime()	116
6.54.2.5 intToDayOfWeek()	117
6.54.2.6 intToMonth()	117
6.54.2.7 ltoBCD()	118
6.54.2.8 setdate()	118

6.54.2.9 setDateInMemory()	119
6.54.2.10 setTime()	119
6.54.2.11 setTimeInMemory()	120
6.55 dnt.h	120
6.56 /home/maximillian/Desktop/MAMA/term/history.c File Reference	121
6.56.1 Function Documentation	121
6.56.1.1 circular_next_index()	121
6.56.1.2 circular_prev_index()	122
6.56.1.3 hist_discard_last_frame()	122
6.56.1.4 hist_forward()	122
6.56.1.5 hist_next_frame()	123
6.56.1.6 hist_rewind()	123
6.56.1.7 write_hist_to_buf()	123
6.57 /home/maximillian/Desktop/MAMA/term/history.h File Reference	124
6.57.1 Function Documentation	124
6.57.1.1 hist_forward()	124
6.57.1.2 hist_next_frame()	125
6.57.1.3 hist_rewind()	125
6.58 history.h	125
6.59 /home/maximillian/Desktop/MAMA/term/cmds/pcb.c File Reference	125
6.60 /home/maximillian/Desktop/MAMA/term/pcb/pcb.c File Reference	125
6.60.1 Function Documentation	127
6.60.1.1 allocatePCB()	127
6.60.1.2 blockPCB()	127
6.60.1.3 createPCB()	127
6.60.1.4 deletePCB()	128
6.60.1.5 findPCB()	128
6.60.1.6 freePCB()	128
6.60.1.7 initPCB()	129
6.60.1.8 insertPCB()	129
6.60.1.9 removePCB()	129
6.60.1.10 resumePCB()	130
6.60.1.11 setPriority()	130
6.60.1.12 setupPCB()	131
6.60.1.13 showAll()	131
6.60.1.14 showBlocked()	131
6.60.1.15 showPCB()	131
6.60.1.16 showReady()	132
6.60.1.17 suspendPCB()	132
6.60.1.18 unblockPCB()	132
6.60.2 Variable Documentation	133
6.60.2.1 f_queue	133

6.60.2.2	fifo_queue	133
6.60.2.3	p_queue	133
6.60.2.4	priority_queue	133
6.61	/home/maximillian/Desktop/MAMA/term/pcb/pcb.h File Reference	133
6.61.1	Macro Definition Documentation	135
6.61.1.1	MAX_NAME_SIZE	135
6.61.1.2	MAX_PRIORITY	135
6.61.1.3	MAX_STACK_SIZE	135
6.61.1.4	MIN_PRIORITY	136
6.61.2	Typedef Documentation	136
6.61.2.1	pcb_node_t	136
6.61.2.2	pcb_queue_t	136
6.61.3	Enumeration Type Documentation	136
6.61.3.1	p_state_t	136
6.61.3.2	pcb_queue_order_t	137
6.61.4	Function Documentation	137
6.61.4.1	allocatePCB()	137
6.61.4.2	blockPCB()	137
6.61.4.3	createPCB()	137
6.61.4.4	deletePCB()	138
6.61.4.5	findPCB()	138
6.61.4.6	freePCB()	139
6.61.4.7	initPCB()	139
6.61.4.8	insertPCB()	139
6.61.4.9	removePCB()	140
6.61.4.10	resumePCB()	140
6.61.4.11	setPriority()	141
6.61.4.12	setupPCB()	141
6.61.4.13	showAll()	141
6.61.4.14	showBlocked()	142
6.61.4.15	showPCB()	142
6.61.4.16	showReady()	142
6.61.4.17	suspendPCB()	143
6.61.4.18	unblockPCB()	143
6.62	pcb.h	144
6.63	/home/maximillian/Desktop/MAMA/term/syntax.c File Reference	145
6.63.1	Function Documentation	145
6.63.1.1	changes_state()	145
6.63.1.2	get_state()	146
6.64	/home/maximillian/Desktop/MAMA/term/syntax.h File Reference	146
6.64.1	Enumeration Type Documentation	146
6.64.1.1	SyntaxState	146

6.64.2 Function Documentation	146
6.64.2.1 changes_state()	147
6.64.2.2 get_state()	147
6.65 syntax.h	147
6.66 /home/maximillian/Desktop/MAMA/term/utils.c File Reference	147
6.66.1 Function Documentation	147
6.66.1.1 is_name_char()	148
6.66.1.2 skip_ws()	148
6.67 /home/maximillian/Desktop/MAMA/term/utils.h File Reference	148
6.67.1 Function Documentation	148
6.67.1.1 is_name_char()	149
6.67.1.2 skip_ws()	150
6.68 utils.h	150
6.69 /home/maximillian/Desktop/MAMA/term/visuals/colorize.c File Reference	150
6.69.1 Macro Definition Documentation	151
6.69.1.1 START_SEQ	151
6.69.2 Enumeration Type Documentation	151
6.69.2.1 Color	151
6.69.3 Function Documentation	152
6.69.3.1 display_bg_color()	152
6.69.3.2 display_fg_color()	152
6.69.3.3 display_italicize()	152
6.69.3.4 display_reset()	152
6.69.3.5 print_color_code()	153
6.70 /home/maximillian/Desktop/MAMA/term/visuals/colorize.h File Reference	153
6.70.1 Enumeration Type Documentation	153
6.70.1.1 Color	153
6.70.2 Function Documentation	154
6.70.2.1 display_bg_color()	154
6.70.2.2 display_fg_color()	154
6.70.2.3 display_italicize()	155
6.70.2.4 display_reset()	155
6.71 colorize.h	155
6.72 /home/maximillian/Desktop/MAMA/term/visuals/cursor.c File Reference	155
6.72.1 Function Documentation	156
6.72.1.1 cursor_down()	156
6.72.1.2 cursor_left()	156
6.72.1.3 cursor_return()	156
6.72.1.4 cursor_right()	156
6.72.1.5 cursor_up()	157
6.73 /home/maximillian/Desktop/MAMA/term/visuals/cursor.h File Reference	157
6.73.1 Function Documentation	157

6.73.1.1 cursor_down()	157
6.73.1.2 cursor_left()	158
6.73.1.3 cursor_return()	158
6.73.1.4 cursor_right()	158
6.73.1.5 cursor_up()	158
6.74 cursor.h	159
6.75 /home/maximillian/Desktop/MAMA/term/visuals/hints.c File Reference	159
6.75.1 Function Documentation	159
6.75.1.1 hint_under_prompt()	159
6.76 /home/maximillian/Desktop/MAMA/term/visuals/hints.h File Reference	160
6.76.1 Function Documentation	160
6.76.1.1 hint_under_prompt()	160
6.77 hints.h	160
6.78 /home/maximillian/Desktop/MAMA/term/visuals/syntax_highlight.c File Reference	160
6.78.1 Function Documentation	161
6.78.1.1 color_for()	161
6.78.1.2 get_state_at()	162
6.78.1.3 switch_to()	162
6.78.1.4 syntax_disable_highlighting()	162
6.78.1.5 syntax_enable_highlighting()	163
6.78.1.6 syntax_handle_char()	163
6.78.1.7 syntax_init()	163
6.78.2 Variable Documentation	163
6.78.2.1 enabled	163
6.78.2.2 newest_switch	164
6.78.2.3 states	164
6.78.2.4 switch_indexes	164
6.79 /home/maximillian/Desktop/MAMA/term/visuals/syntax_highlight.h File Reference	164
6.79.1 Macro Definition Documentation	165
6.79.1.1 MAX_SYNTAX_SWITCHES	165
6.79.1.2 SYNTAX_COLOR_CMD_NAME	165
6.79.1.3 SYNTAX_COLOR_DEFAULT	165
6.79.1.4 SYNTAX_COLOR_DOUBLE_QUOTE_STRING	165
6.79.1.5 SYNTAX_COLOR_PARAM_NAME	165
6.79.1.6 SYNTAX_COLOR_PARAM_VALUE	165
6.79.1.7 SYNTAX_COLOR_SINGLE_QUOTE_STRING	165
6.79.2 Function Documentation	166
6.79.2.1 syntax_disable_highlighting()	166
6.79.2.2 syntax_enable_highlighting()	166
6.79.2.3 syntax_handle_char()	166
6.79.2.4 syntax_init()	166
6.80 syntax_highlight.h	166

6.81 /home/maximillian/Desktop/MAMA/WhoDidWhat.md File Reference	167
Index	169

Chapter 1

MAMA

Check out the [who did what markdown page](#) for a list of contributions from each user during each milestone.

Use the [user manual](#) to find information on available commands.

Use the [programmer manual](#) to find information on individual functions, structs, constants, and other code documentation.

Chapter 2

Who did what table

Update with your contributions every module

	R1	R2	R3	R4	R5	R6
Austin Williams	term/visuals/colorize.c term/visuals/cursor.c term/visuals/syntax_highlight.c term/history.c term/syntax.c term/args.c polling() commhand()					
Maximillian Campbell	polling() commhand() gettime() settime() getdate() setdate() cmd_help() cmd_shutdown() itoa() Setting up doxygen Help pages					
Mohammad Alenezi	print_color_code() display_fg_color() display_bg_color() display_reset() display_italicize() print_color_code() cursor_left() cursor_right() cursor_down() cursor_up() cursor_return()					
Abdullah Alqallaf	cmd_version() VersionOs() Some of Help.c comments for Manual					

Chapter 3

Class Index

3.1 Class List

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

cmd_mapping	9
date_time	9
footer	11
gdt_descriptor_struct	11
gdt_entry_struct	12
header	13
heap	14
idt_entry_struct	15
idt_struct	16
index_entry	17
index_table	17
page_dir	18
page_entry	19
page_table	20
param	20
parsed_args	21
pcb_node_t	
Individual PCB nodes. Each PCB is associated with one node	23
pcb_queue	
"Master" controller of the PCB queue	24
pcb_t	
Process Control Block Structure	25

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

/home/maximillian/Desktop/MAMA/help.c	92
/home/maximillian/Desktop/MAMA/include/string.h	43
/home/maximillian/Desktop/MAMA/include/system.h	46
/home/maximillian/Desktop/MAMA/include/core/asm.h	27
/home/maximillian/Desktop/MAMA/include/core/comhand.h	27
/home/maximillian/Desktop/MAMA/include/core/interrupts.h	28
/home/maximillian/Desktop/MAMA/include/core/io.h	29
/home/maximillian/Desktop/MAMA/include/core/serial.h	30
/home/maximillian/Desktop/MAMA/include/core/tables.h	33
/home/maximillian/Desktop/MAMA/include/mem/heap.h	36
/home/maximillian/Desktop/MAMA/include/mem/paging.h	40
/home/maximillian/Desktop/MAMA/kernel/core/interrupts.c	49
/home/maximillian/Desktop/MAMA/kernel/core/kmain.c	57
/home/maximillian/Desktop/MAMA/kernel/core/serial.c	57
/home/maximillian/Desktop/MAMA/kernel/core/system.c	61
/home/maximillian/Desktop/MAMA/kernel/core/tables.c	61
/home/maximillian/Desktop/MAMA/kernel/mem/heap.c	63
/home/maximillian/Desktop/MAMA/kernel/mem/paging.c	66
/home/maximillian/Desktop/MAMA/lib/out.c	69
/home/maximillian/Desktop/MAMA/lib/out.h	70
/home/maximillian/Desktop/MAMA/lib/string.c	76
/home/maximillian/Desktop/MAMA/modules/mpx_supt.c	78
/home/maximillian/Desktop/MAMA/modules/mpx_supt.h	81
/home/maximillian/Desktop/MAMA/term/args.c	86
/home/maximillian/Desktop/MAMA/term/args.h	89
/home/maximillian/Desktop/MAMA/term/commands.h	100
/home/maximillian/Desktop/MAMA/term/commhand.c	101
/home/maximillian/Desktop/MAMA/term/commhand.h	103
/home/maximillian/Desktop/MAMA/term/history.c	121
/home/maximillian/Desktop/MAMA/term/history.h	124
/home/maximillian/Desktop/MAMA/term/syntax.c	145
/home/maximillian/Desktop/MAMA/term/syntax.h	146
/home/maximillian/Desktop/MAMA/term/utils.c	147
/home/maximillian/Desktop/MAMA/term/utils.h	148
/home/maximillian/Desktop/MAMA/term/ascii/mama.c	90

/home/maximillian/Desktop/MAMA/term/ascii/ mama.h	91
/home/maximillian/Desktop/MAMA/term/cmds/ argtest.c	92
/home/maximillian/Desktop/MAMA/term/cmds/ echo.c	92
/home/maximillian/Desktop/MAMA/term/cmds/ help.c	95
/home/maximillian/Desktop/MAMA/term/cmds/ pcb.c	125
/home/maximillian/Desktop/MAMA/term/cmds/ shutdown.c	99
/home/maximillian/Desktop/MAMA/term/cmds/ version.c	100
/home/maximillian/Desktop/MAMA/term/dnt/ dnt.c	106
/home/maximillian/Desktop/MAMA/term/dnt/ dnt.h	111
/home/maximillian/Desktop/MAMA/term/pcb/ pcb.c	125
/home/maximillian/Desktop/MAMA/term/pcb/ pcb.h	133
/home/maximillian/Desktop/MAMA/term/visuals/ colorize.c	150
/home/maximillian/Desktop/MAMA/term/visuals/ colorize.h	153
/home/maximillian/Desktop/MAMA/term/visuals/ cursor.c	155
/home/maximillian/Desktop/MAMA/term/visuals/ cursor.h	157
/home/maximillian/Desktop/MAMA/term/visuals/ hints.c	159
/home/maximillian/Desktop/MAMA/term/visuals/ hints.h	160
/home/maximillian/Desktop/MAMA/term/visuals/ syntax_highlight.c	160
/home/maximillian/Desktop/MAMA/term/visuals/ syntax_highlight.h	164

Chapter 5

Class Documentation

5.1 cmd_mapping Struct Reference

Public Attributes

- char * [cmd_name](#)
- [cmd_func_t](#) cmd_handler

5.1.1 Member Data Documentation

5.1.1.1 cmd_handler

[cmd_func_t](#) cmd_mapping::cmd_handler

5.1.1.2 cmd_name

char* cmd_mapping::cmd_name

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

- /home/maximillian/Desktop/MAMA/term/[commhand.c](#)

5.2 date_time Struct Reference

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

Public Attributes

- int [sec](#)
- int [min](#)
- int [hour](#)
- int [day_w](#)
- int [day_m](#)
- int [day_y](#)
- int [mon](#)
- int [year](#)

5.2.1 Member Data Documentation

5.2.1.1 [day_m](#)

```
int date_time::day_m
```

5.2.1.2 [day_w](#)

```
int date_time::day_w
```

5.2.1.3 [day_y](#)

```
int date_time::day_y
```

5.2.1.4 [hour](#)

```
int date_time::hour
```

5.2.1.5 [min](#)

```
int date_time::min
```


5.2.1.6 mon

```
int date_time::mon
```

5.2.1.7 sec

```
int date_time::sec
```

5.2.1.8 year

```
int date_time::year
```

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

- [/home/maximillian/Desktop/MAMA/include/system.h](#)

5.3 footer Struct Reference

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

Public Attributes

- [header head](#)

5.3.1 Member Data Documentation

5.3.1.1 head

```
header footer::head
```

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

- [/home/maximillian/Desktop/MAMA/include/mem/heap.h](#)

5.4 gdt_descriptor_struct Struct Reference

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

Public Attributes

- [u16int limit](#)
- [u32int base](#)

5.4.1 Member Data Documentation

5.4.1.1 base

```
u32int gdt_descriptor_struct::base
```

5.4.1.2 limit

```
u16int gdt_descriptor_struct::limit
```

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

- </home/maximillian/Desktop/MAMA/include/core/tables.h>

5.5 gdt_entry_struct Struct Reference

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

Public Attributes

- [u16int limit_low](#)
- [u16int base_low](#)
- [u8int base_mid](#)
- [u8int access](#)
- [u8int flags](#)
- [u8int base_high](#)

5.5.1 Member Data Documentation

5.5.1.1 access

```
u8int gdt_entry_struct::access
```

5.5.1.2 base_high

```
uint8_t gdt_entry_struct::base_high
```

5.5.1.3 base_low

```
uint16_t gdt_entry_struct::base_low
```

5.5.1.4 base_mid

```
uint8_t gdt_entry_struct::base_mid
```

5.5.1.5 flags

```
uint8_t gdt_entry_struct::flags
```

5.5.1.6 limit_low

```
uint16_t gdt_entry_struct::limit_low
```

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

- [/home/maximillian/Desktop/MAMA/include/core/tables.h](#)

5.6 header Struct Reference

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

Public Attributes

- int [size](#)
- int [index_id](#)

5.6.1 Member Data Documentation

5.6.1.1 index_id

```
int header::index_id
```

5.6.1.2 size

```
int header::size
```

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

- [/home/maximillian/Desktop/MAMA/include/mem/heap.h](#)

5.7 heap Struct Reference

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

Public Attributes

- [index_table](#) index
- [u32int](#) base
- [u32int](#) max_size
- [u32int](#) min_size

5.7.1 Member Data Documentation

5.7.1.1 base

```
u32int heap::base
```

5.7.1.2 index

```
index\_table heap::index
```

5.7.1.3 max_size

```
u32int heap::max_size
```

5.7.1.4 min_size

```
u32int heap::min_size
```

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

- [/home/maximillian/Desktop/MAMA/include/mem/heap.h](#)

5.8 idt_entry_struct Struct Reference

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

Public Attributes

- [u16int base_low](#)
- [u16int sselect](#)
- [u8int zero](#)
- [u8int flags](#)
- [u16int base_high](#)

5.8.1 Member Data Documentation

5.8.1.1 base_high

```
u16int idt_entry_struct::base_high
```

5.8.1.2 base_low

```
u16int idt_entry_struct::base_low
```

5.8.1.3 flags

```
uint idt_entry_struct::flags
```

5.8.1.4 sselect

```
uint idt_entry_struct::sselect
```

5.8.1.5 zero

```
uint idt_entry_struct::zero
```

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

- </home/maximillian/Desktop/MAMA/include/core/tables.h>

5.9 idt_struct Struct Reference

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

Public Attributes

- [uint limit](#)
- [uint base](#)

5.9.1 Member Data Documentation

5.9.1.1 base

```
uint idt_struct::base
```

5.9.1.2 limit

```
uint idt_struct::limit
```

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

- </home/maximillian/Desktop/MAMA/include/core/tables.h>

5.10 `index_entry` Struct Reference

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

Public Attributes

- `int` [size](#)
- `int` [empty](#)
- `u32int` [block](#)

5.10.1 Member Data Documentation

5.10.1.1 `block`

```
u32int index_entry::block
```

5.10.1.2 `empty`

```
int index_entry::empty
```

5.10.1.3 `size`

```
int index_entry::size
```

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

- `/home/maximillian/Desktop/MAMA/include/mem/heap.h`

5.11 `index_table` Struct Reference

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

Public Attributes

- `index_entry` [table](#) [[TABLE_SIZE](#)]
- `int` [id](#)

5.11.1 Member Data Documentation

5.11.1.1 id

```
int index_table::id
```

5.11.1.2 table

```
index_entry index_table::table[TABLE\_SIZE]
```

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

- [/home/maximillian/Desktop/MAMA/include/mem/heap.h](#)

5.12 page_dir Struct Reference

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

Public Attributes

- [page_table](#) * [tables](#) [1024]
- [u32int](#) [tables_phys](#) [1024]

5.12.1 Member Data Documentation

5.12.1.1 tables

```
page\_table* page_dir::tables[1024]
```

5.12.1.2 tables_phys

```
u32int page_dir::tables_phys[1024]
```

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

- [/home/maximillian/Desktop/MAMA/include/mem/paging.h](#)

5.13 page_entry Struct Reference

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

Public Attributes

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

5.13.1 Member Data Documentation

5.13.1.1 accessed

```
u32int page_entry::accessed
```

5.13.1.2 dirty

```
u32int page_entry::dirty
```

5.13.1.3 frameaddr

```
u32int page_entry::frameaddr
```

5.13.1.4 present

```
u32int page_entry::present
```

5.13.1.5 reserved

```
u32int page_entry::reserved
```

5.13.1.6 usermode

```
u32int page_entry::usermode
```

5.13.1.7 writeable

```
u32int page_entry::writeable
```

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

- </home/maximillian/Desktop/MAMA/include/mem/paging.h>

5.14 page_table Struct Reference

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

Public Attributes

- [page_entry pages](#) [1024]

5.14.1 Member Data Documentation

5.14.1.1 pages

```
page_entry page_table::pages[1024]
```

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

- </home/maximillian/Desktop/MAMA/include/mem/paging.h>

5.15 param Struct Reference

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

Public Attributes

- int [op_code](#)
- int [device_id](#)
- char * [buffer_ptr](#)
- int * [count_ptr](#)

5.15.1 Member Data Documentation

5.15.1.1 buffer_ptr

```
char* param::buffer_ptr
```

5.15.1.2 count_ptr

```
int* param::count_ptr
```

5.15.1.3 device_id

```
int param::device_id
```

5.15.1.4 op_code

```
int param::op_code
```

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

- [/home/maximillian/Desktop/MAMA/modules/mpx_supt.h](#)

5.16 parsed_args Struct Reference

```
#include <args.h>
```

Public Attributes

- int [flag_count](#)
- int [named_arg_count](#)
- int [unnamed_arg_count](#)
- int [unnamed_args_used_so_far](#)
- char [flags](#) [[MAX_CMD_FLAG_COUNT](#)][[MAX_CMD_ARG_NAME_LEN](#)+1]
- char [named_arg_names](#) [[MAX_CMD_NAMED_ARG_COUNT](#)][[MAX_CMD_ARG_NAME_LEN](#)+1]
- char [named_arg_values](#) [[MAX_CMD_NAMED_ARG_COUNT](#)][[MAX_CMD_ARG_VALUE_LEN](#)+1]
- char [unnamed_args](#) [[MAX_CMD_UNNAMED_ARG_COUNT](#)][[MAX_CMD_ARG_VALUE_LEN](#)+1]

5.16.1 Member Data Documentation

5.16.1.1 `flag_count`

```
int parsed_args::flag_count
```

5.16.1.2 `flags`

```
char parsed_args::flags[MAX_CMD_FLAG_COUNT][MAX_CMD_ARG_NAME_LEN+1]
```

5.16.1.3 `named_arg_count`

```
int parsed_args::named_arg_count
```

5.16.1.4 `named_arg_names`

```
char parsed_args::named_arg_names[MAX_CMD_NAMED_ARG_COUNT][MAX_CMD_ARG_NAME_LEN+1]
```

5.16.1.5 `named_arg_values`

```
char parsed_args::named_arg_values[MAX_CMD_NAMED_ARG_COUNT][MAX_CMD_ARG_VALUE_LEN+1]
```

5.16.1.6 `unnamed_arg_count`

```
int parsed_args::unnamed_arg_count
```

5.16.1.7 `unnamed_args`

```
char parsed_args::unnamed_args[MAX_CMD_UNNAMED_ARG_COUNT][MAX_CMD_ARG_VALUE_LEN+1]
```

5.16.1.8 unnamed_args_used_so_far

```
int parsed_args::unnamed_args_used_so_far
```

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

- </home/maximillian/Desktop/MAMA/term/args.h>

5.17 pcb_node_t Struct Reference

Individual PCB nodes. Each PCB is associated with one node.

```
#include <pcb.h>
```

Public Attributes

- struct [pcb_node_t](#) * [pcbn_next_pcb](#)
Pointer to the Next PCB.
- struct [pcb_node_t](#) * [pcbn_prev_pcb](#)
Pointer to the Previous PCB.
- [pcb_t](#) * [pcb](#)
Pointer to PCB.

5.17.1 Detailed Description

Individual PCB nodes. Each PCB is associated with one node.

5.17.2 Member Data Documentation

5.17.2.1 pcb

```
pcb\_t* pcb_node_t::pcb
```

Pointer to PCB.

5.17.2.2 pcbn_next_pcb

```
struct pcb\_node\_t* pcb_node_t::pcbn_next_pcb
```

Pointer to the Next PCB.

5.17.2.3 pcbn_prev_pcb

```
struct pcb_node_t* pcb_node_t::pcbn_prev_pcb
```

Pointer to the Previous PCB.

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

- /home/maximillian/Desktop/MAMA/term/pcb/[pcb.h](#)

5.18 pcb_queue Struct Reference

"Master" controller of the PCB queue

```
#include <pcb.h>
```

Public Attributes

- int [pcbq_count](#)
Number of PCB's currently in the queue.
- [pcb_node_t](#) * [pcbq_head](#)
Head of the PCB queue.
- [pcb_node_t](#) * [pcbq_tail](#)
Tail of the PCB queue.
- [pcb_queue_order_t](#) [queue_order](#)
Queue order of the Master controller.

5.18.1 Detailed Description

"Master" controller of the PCB queue

5.18.2 Member Data Documentation

5.18.2.1 pcbq_count

```
int pcb_queue::pcbq_count
```

Number of PCB's currently in the queue.

5.18.2.2 pcbq_head

```
pcb_node_t* pcb_queue::pcbq_head
```

Head of the PCB queue.

5.18.2.3 pcbq_tail

```
pcb_node_t* pcb_queue::pcbq_tail
```

Tail of the PCB queue.

5.18.2.4 queue_order

```
pcb_queue_order_t pcb_queue::queue_order
```

Queue order of the Master controller.

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

- [/home/maximillian/Desktop/MAMA/term/pcb/pcb.h](#)

5.19 pcb_t Struct Reference

Process Control Block Structure.

```
#include <pcb.h>
```

Public Attributes

- char [pcb_name](#) [32]
PCB Name.
- int [pcb_process_class](#)
Process Class.
- int [pcb_priority](#)
Priority of PCB.
- [p_state_t](#) [pcb_process_state](#)
State of the PCB.
- unsigned char * [pcb_stack_top](#)
Top of the Stack. Set equal to the stack base + size of the stack.
- unsigned char * [pcb_stack_bottom](#)
Beginning of the Stack.

5.19.1 Detailed Description

Process Control Block Structure.

5.19.2 Member Data Documentation

5.19.2.1 pcb_name

```
char pcb_t::pcb_name[32]
```

PCB Name.

5.19.2.2 pcb_priority

```
int pcb_t::pcb_priority
```

Priority of PCB.

5.19.2.3 pcb_process_class

```
int pcb_t::pcb_process_class
```

Process Class.

5.19.2.4 pcb_process_state

```
p_state_t pcb_t::pcb_process_state
```

State of the PCB.

5.19.2.5 pcb_stack_bottom

```
unsigned char* pcb_t::pcb_stack_bottom
```

Beginning of the Stack.

5.19.2.6 pcb_stack_top

```
unsigned char* pcb_t::pcb_stack_top
```

Top of the Stack. Set equal to the stack base + size of the stack.

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

- </home/maximillian/Desktop/MAMA/term/pcb/pcb.h>

Chapter 6

File Documentation

6.1 /home/maximillian/Desktop/MAMA/include/core/asm.h File Reference

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

6.2 asm.h

[Go to the documentation of this file.](#)

```
1 #ifndef _ASM_H
2 #define _ASM_H
3
4 #include <system.h>
5 #include <tables.h>
6
7 #endif
```

6.3 /home/maximillian/Desktop/MAMA/include/core/comhand.h File Reference

Functions

- int [comhand](#) ()

6.3.1 Function Documentation

6.3.1.1 comhand()

```
int comhand ( )
```

6.4 comhand.h

[Go to the documentation of this file.](#)

```
1 #ifndef _COMHAND_H
2 #define _COMHAND_H
3
4 int comhand();
5
6 #endif
```

6.5 /home/maximillian/Desktop/MAMA/include/core/interrupts.h File Reference

Functions

- void [init_irq](#) (void)
- void [init_pic](#) (void)

6.5.1 Function Documentation

6.5.1.1 init_irq()

```
void init_irq (
    void )
```

6.5.1.2 init_pic()

```
void init_pic (
    void )
```

6.6 interrupts.h

[Go to the documentation of this file.](#)

```
1 #ifndef _INTERRUPTS_H
2 #define _INTERRUPTS_H
3
4 /*
5  Procedure...: init_irq
6  Description...: Installs the initial interrupt handlers for
7                  the first 32 irq lines. Most do a panic for now.
8 */
9 void init_irq(void);
10
11 /*
12  Procedure...: init_pic
13  Description...: Initializes the programmable interrupt controllers
14                  and performs the necessary remapping of IRQs. Leaves interrupts
15                  turned off.
16 */
17 void init_pic(void);
18
19 #endif
```

6.7 /home/maximillian/Desktop/MAMA/include/core/io.h File Reference

Macros

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

6.7.1 Macro Definition Documentation

6.7.1.1 inb

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

Value:

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

6.7.1.2 outb

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

6.8 io.h

[Go to the documentation of this file.](#)

```
1 #ifndef _IO_H
2 #define _IO_H
3
4 /*
5  Procedure...: outb
6  Description...: Write a byte of data to a port.
7  */
8 #define outb(port, data) \  
9  asm volatile ("outb %%al,%%dx" : : "a" (data), "d" (port))
10
11 /*
12  Procedure...: inb
13  Description...: Read a byte of data from a port.
14  */
15 #define inb(port) ({  
16     unsigned char r;  
17     asm volatile ("inb %%dx,%%al": "=a" (r): "d" (port)); \  
18     r;  
19 })
20
21 #endif
```

6.9 /home/maximillian/Desktop/MAMA/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)`
Serially poll characters from command line.

6.9.1 Macro Definition Documentation

6.9.1.1 COM1

```
#define COM1 0x3f8
```

6.9.1.2 COM2

```
#define COM2 0x2f8
```

6.9.1.3 COM3

```
#define COM3 0x3e8
```

6.9.1.4 COM4

```
#define COM4 0x2e8
```

6.9.2 Function Documentation

6.9.2.1 init_serial()

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

6.9.2.2 polling()

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

Serially poll characters from command line.

Polls input from keyboard and interprets each character individually as it is entered from the keyboard.

Parameters

<i>buffer</i>	Space allocated for single line on the command line
<i>count</i>	Size of the space allocated

Returns

Returns 0 upon success, -1 upon error

6.9.2.3 serial_print()

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

6.9.2.4 serial_println()

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

6.9.2.5 set_serial_in()

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

6.9.2.6 set_serial_out()

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

6.10 serial.h

[Go to the documentation of this file.](#)

```
1 #ifndef _SERIAL_H
2 #define _SERIAL_H
3
4 #define COM1 0x3f8
5 #define COM2 0x2f8
6 #define COM3 0x3e8
7 #define COM4 0x2e8
8
9 /*
10  Procedure...: init_serial
11  Description...: Initializes devices for user interaction, logging, ...
12 */
13 int init_serial(int device);
14
15 /*
16  Procedure...: serial_println
17  Description...: Writes a message to the active serial output device.
18                  Appends a newline character.
19 */
20 int serial_println(const char *msg);
21
22 /*
23  Procedure...: serial_print
24  Description...: Writes a message to the active serial output device.
25 */
26 int serial_print(const char *msg);
27
28 /*
29  Procedure...: set_serial_out
30  Description...: Sets serial_port_out to the given device address.
31                  All serial output, such as that from serial_println, will be
32                  directed to this device.
33 */
34 int set_serial_out(int device);
35
36 /*
37  Procedure...: set_serial_in
38  Description...: Sets serial_port_in to the given device address.
39                  All serial input, such as console input via a virtual machine,
40                  QEMU/Bochs/etc, will be directed to this device.
41 */
42 int set_serial_in(int device);
43
44 /*
45  Procedure: Polling
46  Description: Gathers keyboard input via the serial port using
47               the technique of polling
48 */
49
50 int *polling(char *buffer, int *count);
51
52 #endif
```

6.11 /home/maximillian/Desktop/MAMA/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

6.11.1 Function Documentation

6.11.1.1 [__attribute__\(\)](#)

```
struct gdt\_entry\_struct \_\_attribute\_\_ (  
    (packed) )
```

6.11.1.2 gdt_init_entry()

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

6.11.1.3 idt_set_gate()

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

6.11.1.4 init_gdt()

```
void init_gdt ( )
```

6.11.1.5 init_idt()

```
void init_idt ( )
```

6.11.2 Variable Documentation

6.11.2.1 access

`u8int` access

6.11.2.2 base

`u32int` base

6.11.2.3 base_high

`u8int` base_high

6.11.2.4 base_low

`ul6int` base_low

6.11.2.5 base_mid

`u8int` base_mid

6.11.2.6 flags

`u8int` flags

6.11.2.7 limit

`ul6int` limit

6.11.2.8 limit_low

`ul6int` limit_low

6.11.2.9 sselect

`ul6int` sselect

6.11.2.10 zero

`u8int` zero

6.12 tables.h

[Go to the documentation of this file.](#)

```

1 #ifndef _TABLES_H
2 #define _TABLES_H
3
4 #include "system.h"
5
6 typedef struct idt_entry_struct
7 {
8     u16int base_low; //offset bits 0..15
9     u16int sselect; //stack selector in gdt or ldt
10    u8int zero; //this stays zero; unused
11    u8int flags; //attributes
12    u16int base_high; //offset bits 16..31
13 }
14 __attribute__((packed)) idt_entry;
15
16 typedef struct idt_struct
17 {
18     u16int limit;
19     u32int base;
20 }
21 __attribute__((packed)) idt_descriptor;
22
23 typedef struct gdt_descriptor_struct
24 {
25     u16int limit;
26     u32int base;
27 }
28 __attribute__((packed)) gdt_descriptor;
29
30 typedef struct gdt_entry_struct
31 {
32     u16int limit_low; //first 16 bits of limit
33     u16int base_low; //first 16 bits of base
34     u8int base_mid; //bits 16-23 of base
35     u8int access; //next 8 bits; access flags
36     u8int flags; //page granularity, size
37     u8int base_high; //last 8 bits of the base
38 }
39 __attribute__((packed)) gdt_entry;
40
41
42 void idt_set_gate(u8int idx, u32int base, u16int sel, u8int flags);
43 void gdt_init_entry(int idx, u32int base, u32int limit, u8int access,
44                    u8int flags);
45
46 void init_idt();
47 void init_gdt();
48
49 #endif

```

6.13 /home/maximillian/Desktop/MAMA/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)

6.13.1 Macro Definition Documentation

6.13.1.1 KHEAP_BASE

```
#define KHEAP_BASE 0xD000000
```

6.13.1.2 KHEAP_MIN

```
#define KHEAP_MIN 0x10000
```

6.13.1.3 KHEAP_SIZE

```
#define KHEAP_SIZE 0x1000000
```

6.13.1.4 TABLE_SIZE

```
#define TABLE_SIZE 0x1000
```

6.13.2 Function Documentation

6.13.2.1 _kmalloc()

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

6.13.2.2 alloc()

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

6.13.2.3 init_kheap()

```
void init_kheap ( )
```

6.13.2.4 kfree()

```
u32int kfree ( )
```

6.13.2.5 kmalloc()

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

6.13.2.6 make_heap()

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

6.14 heap.h

[Go to the documentation of this file.](#)

```

1  #ifndef _HEAP_H
2  #define _HEAP_H
3
4  /* Kernel heap */
5  #define TABLE_SIZE 0x1000
6  #define KHEAP_BASE 0xD000000
7  #define KHEAP_MIN 0x10000
8  #define KHEAP_SIZE 0x1000000
9
10 /* Heap allocation header */
11 typedef struct {
12     int size;
13     int index_id;
14 } header;
15
16 typedef struct {
17     header head;
18 } footer;
19
20 typedef struct {
21     int size;
22     int empty;
23     u32int block;
24 } index_entry;
25
26 /* Kernel heap index table */
27 typedef struct {
28     index_entry table[TABLE_SIZE];
29     int id;
30 } index_table;
31
32 /* Heap structure */
33 typedef struct {
34     index_table index;
35     u32int base;
36     u32int max_size;
37     u32int min_size;
38 } heap;
39
40 /*
41  Procedure...: _kmalloc
42  Description...: Base-level kernel memory allocation routine. Used to
43                  provide page alignment and access physical addresses of allocations.
44                  Called by kmalloc with align=0, physical_address=0.
45 */
46 u32int _kmalloc(u32int size, int align, u32int *phys_addr);
47
48 /*
49  Procedure...: kmalloc
50  Description...: Standard kernel memory allocation routine. Use this unless you
51                  need to specify alignment or obtain a physical address. Calls _kmalloc.
52 */
53 u32int kmalloc(u32int size);
54
55 /*
56  Procedure...: kfree
57  Description...: Free kernel memory.
58 */
59 u32int kfree();
60
61 /*
62  Procedure...: init_kheap
63  Description...: Initialize the kernel heap, and set it as the current heap.
64 */
65 void init_kheap();
66
67 /*
68  Procedure...: alloc
69  Description...: Allocate some memory using the given heap. Can specify page-alignment.
70 */
71 u32int alloc(u32int size, heap *hp, int align);
72
73 /*
74  Procedure...: make_heap
75  Description...: Create a new heap.
76  Parameters...: base - physical start address of the heap
77                  max - maximum size the heap may grow to
78                  min - minimum/initial size
79 */
80 heap* make_heap(u32int base, u32int max, u32int min);
81
82 #endif

```

6.15 /home/maximillian/Desktop/MAMA/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)

6.15.1 Macro Definition Documentation

6.15.1.1 PAGE_SIZE

```
#define PAGE_SIZE 0x1000
```

6.15.2 Function Documentation

6.15.2.1 clear_bit()

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

6.15.2.2 first_free()

```
u32int first_free ( )
```

6.15.2.3 get_bit()

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

6.15.2.4 get_page()

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

6.15.2.5 init_paging()

```
void init_paging ( )
```

6.15.2.6 load_page_dir()

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

6.15.2.7 new_frame()

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

6.15.2.8 set_bit()

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

6.16 paging.h

[Go to the documentation of this file.](#)

```

1 #ifndef _PAGING_H
2 #define _PAGING_H
3
4 #include <system.h>
5
6 #define PAGE_SIZE 0x1000
7
8 /*
9  Page entry structure
10  Describes a single page in memory
11 */
12 typedef struct {
13     u32int present : 1;
14     u32int writeable : 1;
15     u32int usermode : 1;
16     u32int accessed : 1;
17     u32int dirty : 1;
18     u32int reserved : 7;
19     u32int frameaddr : 20;
20 } page_entry;
21
22 /*
23  Page table structure
24  Contains 1024 pages/frames
25 */
26 typedef struct {
27     page_entry pages[1024];
28 } page_table;
29
30 /*
31  Page directory structure
32  Limited to 1024 tables for now
33 */
34 typedef struct {
35     page_table *tables[1024];
36     u32int tables_phys[1024];
37 } page_dir;
38
39 /*
40  Procedure...: set_bit
41  Description...: Marks a page frame bit as in use (1).
42 */
43 void set_bit(u32int addr);
44
45 /*
46  Procedure...: clear_bit
47  Description...: Marks a page frame bit as free (0).
48 */
49 void clear_bit(u32int addr);
50
51 /*
52  Procedure...: get_bit
53  Description...: Checks if page frame is in use.
54 */
55 u32int get_bit(u32int addr);
56
57 /*
58  Procedure...: first_free
59  Description...: Finds the first free page frame.
60 */
61 u32int first_free();
62
63 /*
64  Procedure...: init_paging
65  Description...: Initializes the kernel page directory and
66                  initial kernel heap area. Performs identity mapping of
67                  the kernel frames such that the virtual addresses are
68                  equivalent to the physical addresses.
69 */
70 void init_paging();
71
72 /*
73  Procedure...: load_page_dir
74  Description...: Sets a page directory as the current
75                  directory and enables paging via the cr0 register.
76                  The cr3 register enables address translation from
77                  linear to physical addresses.
78                  http://en.wikipedia.org/wiki/Control_register#Control_registers_in_x86_series
79 */
80 void load_page_dir(page_dir *new_page_dir);
81
82 */

```



```

83 Procedure...: get_page
84 Description...: Finds and returns a page, allocating a new
85 page table if necessary.
86 */
87 page_entry* get_page(u32int addr, page_dir *dir, int make_table);
88
89 /*
90 Procedure...: new_frame
91 Description...: Marks a frame as in use in the frame bitmap,
92 sets up the page, and saves the frame index in the page.
93 */
94 void new_frame(page_entry* page);
95
96 #endif

```

6.17 /home/maximillian/Desktop/MAMA/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)
 - char * [itoa](#) (int i)
- Converts 32-bit integer to an array of 8-bit characters.*

6.17.1 Function Documentation

6.17.1.1 atoi()

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

6.17.1.2 isspace()

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

6.17.1.3 itoa()

```
char * itoa (
    int i )
```

Converts 32-bit integer to an array of 8-bit characters.

Converts an integer data type by breaking it down into its individual digits. Digits are stored individually into a character array.

Parameters

<i>i</i>	Integer that will be converted into ascii
----------	---

Returns

Returns a pointer to the start of the array of character bytes

6.17.1.4 memset()

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

6.17.1.5 strcat()

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

6.17.1.6 strcmp()

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

6.17.1.7 strcpy()

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

6.17.1.8 strlen()

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

6.17.1.9 strtok()

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

6.18 string.h

[Go to the documentation of this file.](#)

```
1 #ifndef _STRING_H
2 #define _STRING_H
3
4 #include <system.h>
5
6 /*
7  Procedure...: isspace
8  Description...: Determine if a character is whitespace.
9  Params...: c-character to check
10 */
11 int isspace(const char *c);
12
13 /*
14  Procedure...: memset
15  Description...: Set a region of memory.
16  Params...: s-destination, c-byte to write, n-count
17 */
18 void* memset(void *s, int c, size_t n);
19
20 /*
21  Procedure...: strcpy
22  Description...: Copy one string to another.
23  Params...: s1-destination, s2-source
24 */
25 char* strcpy(char *s1, const char *s2);
26
27 /*
28  Procedure...: strcat
29  Description...: Concatenate the contents of one string onto another.
30  Params...: s1-destination, s2-source
31 */
32 char* strcat(char *s1, const char *s2);
33
34 /*
35  Procedure...: strlen
36  Description...: Returns the length of a string.
37  Params...: s-input string
38 */
39 int strlen(const char *s);
40
41 /*
42  Procedure...: strcmp
43  Description...: String comparison
44  Params...: s1-string 1, s2-string 2
45 */
46 int strcmp(const char *s1, const char *s2);
47
48 /*
49  Procedure...: strtok
50  Description...: Split string into tokens
51  Params...: s1-string, s2-delimiter
52 */
53 char* strtok(char *s1, const char *s2);
54
55 /*
56  Procedure...: atoi
57  Description...: Convert an ASCII string to an integer
58  Params...: const char *s -- String
59 */
60 int atoi(const char *s);
61
62 char *itoa(int i);
63
64 #endif
```

6.19 /home/maximillian/Desktop/MAMA/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)

6.19.1 Macro Definition Documentation

6.19.1.1 [asm](#)

```
#define asm __asm__
```

6.19.1.2 [cli](#)

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

6.19.1.3 GDT_CS_ID

```
#define GDT_CS_ID 0x01
```

6.19.1.4 GDT_DS_ID

```
#define GDT_DS_ID 0x02
```

6.19.1.5 hlt

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

6.19.1.6 iret

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

6.19.1.7 no_warn

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

6.19.1.8 nop

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

6.19.1.9 NULL

```
#define NULL 0
```

6.19.1.10 sti

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

6.19.1.11 volatile

```
#define volatile __volatile__
```

6.19.2 Typedef Documentation

6.19.2.1 size_t

```
typedef unsigned int size_t
```

6.19.2.2 u16int

```
typedef unsigned short u16int
```

6.19.2.3 u32int

```
typedef unsigned long u32int
```

6.19.2.4 u8int

```
typedef unsigned char u8int
```

6.19.3 Function Documentation

6.19.3.1 klogv()

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

6.19.3.2 kpanic()

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

6.20 system.h

[Go to the documentation of this file.](#)

```
1 #ifndef _SYSTEM_H
2 #define _SYSTEM_H
3
4 #define NULL 0
5
6 // Suppress 'unused parameter' warnings/errors
7 #define no_warn(p) if (p) while (1) break
8
9 // Allows compilation with gcc -std=c89
10 // May also help avoid naming conflicts
11 #define asm __asm__
12 #define volatile __volatile__
13
14 #define sti()    asm volatile ("sti::") //turn irqs off
15 #define cli()    asm volatile ("cli::") //turn irqs on
16 #define nop()    asm volatile ("nop::") //skip cycle
17 #define hlt()    asm volatile ("hlt::") //halt
18 #define iret()   asm volatile ("iret::") //interrupt return
19
20 #define GDT_CS_ID 0x01 //kernel code segment ID
21 #define GDT_DS_ID 0x02 //kernel data segment ID
22
23 /* System Types */
24 typedef unsigned int    size_t;
25 typedef unsigned char   u8int;
26 typedef unsigned short  u16int;
27 typedef unsigned long   u32int;
28
29 /* Time */
30 typedef struct {
31     int sec;
32     int min;
33     int hour;
34     int day_w;
35     int day_m;
36     int day_y;
37     int mon;
38     int year;
39 } date_time;
40
41 /* Test if interrupts are on */
42 static inline int irq_on()
43 {
44     int f;
45     asm volatile ("pushf\n\t"
46                  "popl %0"
47                  : "=g"(f));
48     return f & (1 << 9);
49 }
50
51 void klogv(const char *msg);
52 void kpanic(const char *msg);
53
54 #endif
```

6.21 /home/maximillian/Desktop/MAMA/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 `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]

6.21.1 Macro Definition Documentation

6.21.1.1 ICW1

```
#define ICW1 0x11
```

6.21.1.2 ICW4

```
#define ICW4 0x01
```

6.21.1.3 io_wait

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

6.21.1.4 PIC1

```
#define PIC1 0x20
```

6.21.1.5 PIC2

```
#define PIC2 0xA0
```

6.21.2 Function Documentation

6.21.2.1 bounds()

```
void bounds ( )
```

6.21.2.2 breakpoint()

```
void breakpoint ( )
```

6.21.2.3 coprocessor()

```
void coprocessor ( )
```

6.21.2.4 coprocessor_segment()

```
void coprocessor_segment ( )
```

6.21.2.5 debug()

```
void debug ( )
```

6.21.2.6 device_not_available()

```
void device_not_available ( )
```

6.21.2.7 divide_error()

```
void divide_error ( )
```

6.21.2.8 do_bounds()

```
void do_bounds ( )
```

6.21.2.9 do_breakpoint()

```
void do_breakpoint ( )
```

6.21.2.10 do_coprocessor()

```
void do_coprocessor ( )
```

6.21.2.11 do_coprocessor_segment()

```
void do_coprocessor_segment ( )
```

6.21.2.12 do_debug()

```
void do_debug ( )
```

6.21.2.13 do_device_not_available()

```
void do_device_not_available ( )
```

6.21.2.14 do_divide_error()

```
void do_divide_error ( )
```

6.21.2.15 do_double_fault()

```
void do_double_fault ( )
```

6.21.2.16 do_general_protection()

```
void do_general_protection ( )
```

6.21.2.17 do_invalid_op()

```
void do_invalid_op ( )
```

6.21.2.18 do_invalid_tss()

```
void do_invalid_tss ( )
```

6.21.2.19 do_isr()

```
void do_isr ( )
```

6.21.2.20 do_nmi()

```
void do_nmi ( )
```

6.21.2.21 do_overflow()

```
void do_overflow ( )
```

6.21.2.22 do_page_fault()

```
void do_page_fault ( )
```

6.21.2.23 do_reserved()

```
void do_reserved ( )
```

6.21.2.24 do_segment_not_present()

```
void do_segment_not_present ( )
```

6.21.2.25 do_stack_segment()

```
void do_stack_segment ( )
```

6.21.2.26 double_fault()

```
void double_fault ( )
```

6.21.2.27 general_protection()

```
void general_protection ( )
```

6.21.2.28 init_irq()

```
void init_irq (
    void )
```

6.21.2.29 init_pic()

```
void init_pic (
    void )
```

6.21.2.30 invalid_op()

```
void invalid_op ( )
```

6.21.2.31 invalid_tss()

```
void invalid_tss ( )
```

6.21.2.32 isr0()

```
void isr0 ( )
```

6.21.2.33 nmi()

```
void nmi ( )
```

6.21.2.34 overflow()

```
void overflow ( )
```

6.21.2.35 page_fault()

```
void page_fault ( )
```

6.21.2.36 reserved()

```
void reserved ( )
```

6.21.2.37 rtc_isr()

```
void rtc_isr ( )
```

6.21.2.38 segment_not_present()

```
void segment_not_present ( )
```

6.21.2.39 stack_segment()

```
void stack_segment ( )
```

6.21.3 Variable Documentation

6.21.3.1 idt_entries

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

6.22 /home/maximillian/Desktop/MAMA/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 <term/commhand.c>
```

Functions

- void [kmain](#) (void)

6.22.1 Function Documentation

6.22.1.1 kmain()

```
void kmain (
    void )
```

6.23 /home/maximillian/Desktop/MAMA/kernel/core/serial.c File Reference

```
#include <stdint.h>
#include <string.h>
#include <core/io.h>
#include <core/serial.h>
#include <term/history.h>
#include <term/visuals/syntax_highlight.h>
#include <term/visuals/syntax_highlight.c>
```

Macros

- `#define NO_ERROR 0`
- `#define DELETE 0b00001`
- `#define LEFT_ARROW 0b00010`
- `#define RIGHT_ARROW 0b00100`
- `#define UP_ARROW 0b01000`
- `#define DOWN_ARROW 0b10000`

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)`
- `unsigned int consume_special ()`
- `int * polling (char *buffer, int *count)`
Serially poll characters from command line.

Variables

- `int serial_port_out = 0`
- `int serial_port_in = 0`

6.23.1 Macro Definition Documentation

6.23.1.1 DELETE

```
#define DELETE 0b00001
```

6.23.1.2 DOWN_ARROW

```
#define DOWN_ARROW 0b10000
```

6.23.1.3 LEFT_ARROW

```
#define LEFT_ARROW 0b00010
```


6.23.1.4 NO_ERROR

```
#define NO_ERROR 0
```

6.23.1.5 RIGHT_ARROW

```
#define RIGHT_ARROW 0b00100
```

6.23.1.6 UP_ARROW

```
#define UP_ARROW 0b01000
```

6.23.2 Function Documentation

6.23.2.1 consume_special()

```
unsigned int consume_special ( )
```

6.23.2.2 init_serial()

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

6.23.2.3 polling()

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

Serially poll characters from command line.

Polls input from keyboard and interprets each character individually as it is entered from the keyboard.

Parameters

<i>buffer</i>	Space allocated for single line on the command line
<i>count</i>	Size of the space allocated

Returns

Returns 0 upon success, -1 upon error

6.23.2.4 serial_print()

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

6.23.2.5 serial_println()

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

6.23.2.6 set_serial_in()

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

6.23.2.7 set_serial_out()

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

6.23.3 Variable Documentation**6.23.3.1 serial_port_in**

```
int serial_port_in = 0
```

6.23.3.2 serial_port_out

```
int serial_port_out = 0
```

6.24 /home/maximillian/Desktop/MAMA/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)

6.24.1 Function Documentation

6.24.1.1 klogv()

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

6.24.1.2 kpanic()

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

6.25 /home/maximillian/Desktop/MAMA/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]

6.25.1 Function Documentation

6.25.1.1 gdt_init_entry()

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

6.25.1.2 idt_set_gate()

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

6.25.1.3 init_gdt()

```
void init_gdt ( )
```

6.25.1.4 init_idt()

```
void init_idt ( )
```

6.25.1.5 write_gdt_ptr()

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

6.25.1.6 write_idt_ptr()

```
void write_idt_ptr (
    u32int )
```

6.25.2 Variable Documentation

6.25.2.1 gdt_entries

```
gdt_entry gdt_entries[5]
```

6.25.2.2 gdt_ptr

```
gdt_descriptor gdt_ptr
```

6.25.2.3 idt_entries

```
idt_entry idt_entries[256]
```

6.25.2.4 idt_ptr

```
idt_descriptor idt_ptr
```

6.26 /home/maximillian/Desktop/MAMA/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`

6.26.1 Function Documentation

6.26.1.1 _kmalloc()

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

6.26.1.2 alloc()

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

6.26.1.3 kmalloc()

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

6.26.1.4 make_heap()

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

6.26.2 Variable Documentation

6.26.2.1 __end

```
void __end
```

6.26.2.2 _end

```
void _end
```

6.26.2.3 curr_heap

```
heap* curr_heap = 0
```

6.26.2.4 end

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

6.26.2.5 kdir

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

6.26.2.6 kheap

```
heap* kheap = 0
```

6.26.2.7 phys_alloc_addr

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

6.27 /home/maximillian/Desktop/MAMA/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](#)

6.27.1 Function Documentation

6.27.1.1 clear_bit()

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


6.27.1.2 find_free()

```
u32int find_free ( )
```

6.27.1.3 get_bit()

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

6.27.1.4 get_page()

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

6.27.1.5 init_paging()

```
void init_paging ( )
```

6.27.1.6 load_page_dir()

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

6.27.1.7 new_frame()

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

6.27.1.8 set_bit()

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

6.27.2 Variable Documentation

6.27.2.1 cdir

```
page_dir* cdir = 0
```

6.27.2.2 frames

```
u32int* frames
```

6.27.2.3 kdir

```
page_dir* kdir = 0
```

6.27.2.4 kheap

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

6.27.2.5 mem_size

```
u32int mem_size = 0x4000000
```

6.27.2.6 nframes

```
u32int nframes
```

6.27.2.7 page_size

```
u32int page_size = 0x1000
```

6.27.2.8 phys_alloc_addr

```
u32int phys_alloc_addr [extern]
```

6.28 /home/maximillian/Desktop/MAMA/lib/out.c File Reference

```
#include <modules/mpx_supt.h>
#include <stdarg.h>
```

Functions

- int `print` (char *str, int len)
- int `putc` (char c)
- int `println` (char *str, int len)
- void `printf` (char *str,...)
- int `read` (char *buf, int len)

6.28.1 Function Documentation

6.28.1.1 print()

```
int print (
    char * str,
    int len )
```

6.28.1.2 putc()

```
int putc (
    char c )
```

6.28.1.3 printf()

```
void printf (
    char * str,
    ... )
```

6.28.1.4 println()

```
int println (
    char * str,
    int len )
```

6.28.1.5 read()

```
int read (
    char * buf,
    int len )
```

6.29 /home/maximillian/Desktop/MAMA/lib/out.h File Reference

Functions

- int [cmd_help](#) (char *command)
Prints help message for command.
- void [gettimeHelp](#) ()
Help page for [gettime\(\)](#) method.
- void [settimeHelp](#) ()
Help page for [settime\(\)](#) method.
- void [getdateHelp](#) ()
Help page for the [getdate\(\)](#) method.
- void [setdateHelp](#) ()
Help page for the [setdate\(\)](#) method.
- void [helpHelp](#) ()
Help page for the help command.
- void [shutdownHelp](#) ()
Help page for the shutdown command.
- void [helpList](#) ()
Displays a list of common system commands.
- void [versionHelp](#) ()
Help page for the version command.
- void [createpcbHelp](#) ()
Help page for createpcb.
- void [deletepcbHelp](#) ()
Help page for deletepcb.
- void [showpcbHelp](#) ()
Help page for showpcb.
- void [showallpcbHelp](#) ()
Help page for showallpcb.
- void [showreadypcbHelp](#) ()
Help page for showreadypcb.
- void [showblockedpcbHelp](#) ()
Help page for showblockedpcb.
- void [blockHelp](#) ()

- *Help page for block.*
- void [unblockHelp](#) ()
- *Help page for unblock.*
- void [setpriorityHelp](#) ()
- *Help page for setpriority.*
- void [resumeHelp](#) ()
- *Help page for resume.*
- void [suspendHelp](#) ()
- *Help page for suspend.*
- int [print](#) (char *, int)
- int [putc](#) (char)
- int [println](#) (char *, int)
- void [printf](#) (char *,...)
- int [read](#) (char *, int)

6.29.1 Function Documentation

6.29.1.1 blockHelp()

```
void blockHelp ( )
```

Help page for block.

Displays the block help page

6.29.1.2 cmd_help()

```
int cmd_help (
    char * command )
```

Prints help message for command.

Prints out a help message and basic syntax for a specific command

Parameters

<i>command</i>	Command which the user needs basic information and syntax for
----------------	---

Returns

1 upon success, -1 upon error

6.29.1.3 createpcbHelp()

```
void createpcbHelp ( )
```

Help page for createpcb.

Displays the createpcb help page

6.29.1.4 deletepcbHelp()

```
void deletepcbHelp ( )
```

Help page for deletepcb.

Displays the deletepcb help page

6.29.1.5 getdateHelp()

```
void getdateHelp ( )
```

Help page for the [getdate\(\)](#) method.

Prints out the name, usage, return and description for the [getdate\(\)](#) method.

6.29.1.6 gettimeHelp()

```
void gettimeHelp ( )
```

Help page for [gettime\(\)](#) method.

Prints out the name, usage, return and description for the [gettime\(\)](#) method.

6.29.1.7 helpHelp()

```
void helpHelp ( )
```

Help page for the help command.

Prints out the name, usage, return and description for the help command.

6.29.1.8 helpList()

```
void helpList ( )
```

Displays a list of common system commands.

Displays a list of common system commands for the user.

6.29.1.9 print()

```
int print (
    char * str,
    int len )
```

6.29.1.10 printc()

```
int printc (
    char c )
```

6.29.1.11 printf()

```
void printf (
    char * str,
    ... )
```

6.29.1.12 println()

```
int println (
    char * str,
    int len )
```

6.29.1.13 read()

```
int read (
    char * buf,
    int len )
```

6.29.1.14 resumeHelp()

```
void resumeHelp ( )
```

Help page for resume.

Displays the resume help page

6.29.1.15 setdateHelp()

```
void setdateHelp ( )
```

Help page for the [setdate\(\)](#) method.

Prints out the name, usage, and description for the [setdate\(\)](#) method.

6.29.1.16 setpriorityHelp()

```
void setpriorityHelp ( )
```

Help page for setpriority.

Displays the setpriority help page

6.29.1.17 settimeHelp()

```
void settimeHelp ( )
```

Help page for [settime\(\)](#) method.

Prints out the name, usage, and description for the [settime\(\)](#) method.

6.29.1.18 showallpcbHelp()

```
void showallpcbHelp ( )
```

Help page for showallpcb.

Displays the showallpcb help page

6.29.1.19 showblockedpcbHelp()

```
void showblockedpcbHelp ( )
```

Help page for showblockedpcb.

Displays the showblockedpcb help page

6.29.1.20 showpcbHelp()

```
void showpcbHelp ( )
```

Help page for showpcb.

Displays the showpcb help page

6.29.1.21 showreadypcbHelp()

```
void showreadypcbHelp ( )
```

Help page for showreadypcb.

Displays the showreadypcb help page

6.29.1.22 shutdownHelp()

```
void shutdownHelp ( )
```

Help page for the shutdown command.

Prints out the name, usage, and description for the shutdown system command.

6.29.1.23 suspendHelp()

```
void suspendHelp ( )
```

Help page for suspend.

Displays the suspend help page

6.29.1.24 unblockHelp()

```
void unblockHelp ( )
```

Help page for unblock.

Displays te unblock help page

6.29.1.25 versionHelp()

```
void versionHelp ( )
```

Help page for the version command.

Displays the current version of the system.

6.30 out.h

[Go to the documentation of this file.](#)

```

1 #ifndef OUT_H
2 #define OUT_H
3
14 int cmd_help(char * command);
15
21 void gettimeHelp();
22
28 void settimeHelp();
29
35 void getdateHelp();
36
42 void setdateHelp();
43
49 void helpHelp();
50
57 void shutdownHelp();
58
64 void helpList();
65
72 void versionHelp();
73
80 void createpcbHelp();
81
87 void deletepcbHelp();
88
94 void showpcbHelp();
95
101 void showallpcbHelp();
102
108 void showreadypcbHelp();
109
115 void showblockedpcbHelp();
116
122 void blockHelp();
123
129 void unblockHelp();
130
136 void setpriorityHelp();
137
143 void resumeHelp();
144
150 void suspendHelp();
151
152 int print(char *, int);
153 int printc(char);
154 int println(char *, int);
155 void printf(char *, ...);
156 int read(char *, int);
157
158 #endif

```

6.31 /home/maximillian/Desktop/MAMA/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)
- char * [itoa](#) (int value)

Converts 32-bit integer to an array of 8-bit characters.
- 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)

6.31.1 Function Documentation

6.31.1.1 atoi()

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

6.31.1.2 isspace()

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

6.31.1.3 itoa()

```
char * itoa (
    int i )
```

Converts 32-bit integer to an array of 8-bit characters.

Converts an integer data type by breaking it down into its individual digits. Digits are stored individually into a character array.

Parameters

<i>i</i>	Integer that will be converted into ascii
----------	---

Returns

Returns a pointer to the start of the array of character bytes

6.31.1.4 memset()

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

6.31.1.5 strcat()

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

6.31.1.6 strcmp()

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

6.31.1.7 strcpy()

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

6.31.1.8 strlen()

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

6.31.1.9 strtok()

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

6.32 /home/maximillian/Desktop/MAMA/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` ()

Variables

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

6.32.1 Function Documentation

6.32.1.1 `idle()`

```
void idle ( )
```

6.32.1.2 `mpx_init()`

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

6.32.1.3 `sys_alloc_mem()`

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

6.32.1.4 `sys_free_mem()`

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

6.32.1.5 sys_req()

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

6.32.1.6 sys_set_free()

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

6.32.1.7 sys_set_malloc()

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

6.32.2 Variable Documentation

6.32.2.1 current_module

```
int current_module = -1
```

6.32.2.2 params

```
param params
```

6.32.2.3 student_free

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

6.32.2.4 student_malloc

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

6.33 /home/maximillian/Desktop/MAMA/modules/mpx_supt.h File Reference

```
#include <system.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](#) ()

6.33.1 Macro Definition Documentation

6.33.1.1 COM_PORT

```
#define COM_PORT 222
```

6.33.1.2 DEFAULT_DEVICE

```
#define DEFAULT_DEVICE 111
```

6.33.1.3 EXIT

```
#define EXIT 0
```

6.33.1.4 FALSE

```
#define FALSE 0
```

6.33.1.5 IDLE

```
#define IDLE 1
```

6.33.1.6 INVALID_BUFFER

```
#define INVALID_BUFFER 1000
```

6.33.1.7 INVALID_COUNT

```
#define INVALID_COUNT 2000
```


6.33.1.8 INVALID_OPERATION

```
#define INVALID_OPERATION 4
```

6.33.1.9 IO_MODULE

```
#define IO_MODULE 10
```

6.33.1.10 MEM_MODULE

```
#define MEM_MODULE 11
```

6.33.1.11 MODULE_F

```
#define MODULE_F 9
```

6.33.1.12 MODULE_R1

```
#define MODULE_R1 0
```

6.33.1.13 MODULE_R2

```
#define MODULE_R2 1
```

6.33.1.14 MODULE_R3

```
#define MODULE_R3 2
```

6.33.1.15 MODULE_R4

```
#define MODULE_R4 4
```

6.33.1.16 MODULE_R5

```
#define MODULE_R5 8
```

6.33.1.17 READ

```
#define READ 2
```

6.33.1.18 TRUE

```
#define TRUE 1
```

6.33.1.19 WRITE

```
#define WRITE 3
```

6.33.2 Function Documentation

6.33.2.1 idle()

```
void idle ( )
```

6.33.2.2 mpx_init()

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

6.33.2.3 sys_alloc_mem()

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

6.33.2.4 sys_free_mem()

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

6.33.2.5 sys_req()

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

6.33.2.6 sys_set_free()

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

6.33.2.7 sys_set_malloc()

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

6.34 mpx_supt.h

[Go to the documentation of this file.](#)

```
1 #ifndef _MPX_SUPT_H
2 #define _MPX_SUPT_H
3
4 #include <system.h>
5
6 #define EXIT 0
7 #define IDLE 1
8 #define READ 2
9 #define WRITE 3
10 #define INVALID_OPERATION 4
11
12 #define TRUE 1
13 #define FALSE 0
14
15 #define MODULE_R1 0
16 #define MODULE_R2 1
17 #define MODULE_R3 2
18 #define MODULE_R4 4
19 #define MODULE_R5 8
20 #define MODULE_F 9
21 #define IO_MODULE 10
22 #define MEM_MODULE 11
23
24 // error codes
25 #define INVALID_BUFFER 1000
26 #define INVALID_COUNT 2000
27
```

```

28 #define DEFAULT_DEVICE 111
29 #define COM_PORT 222
30
31 typedef struct {
32     int op_code;
33     int device_id;
34     char *buffer_ptr;
35     int *count_ptr;
36 } param;
37
38 /*
39 Procedure...: sys_req
40 Description...: Generate interrupt 60H
41 Params...: int op_code one of (IDLE, EXIT, READ, WRITE)
42 */
43 int sys_req( int op_code, int device_id, char *buffer_ptr,
44             int *count_ptr );
45
46 /*
47 Procedure...: mpx_init
48 Description...: Initialize MPX support software
49 Params...: int cur_mod (symbolic constants MODULE_R1, MODULE_R2, etc
50 */
51 void mpx_init(int cur_mod);
52
53 /*
54 Procedure...: sys_set_malloc
55 Description...: Sets the memory allocation function for sys_alloc_mem
56 Params...: Function pointer
57 */
58 void sys_set_malloc(u32int (*func)(u32int));
59
60 /*
61 Procedure...: sys_set_free
62 Description...: Sets the memory free function for sys_free_mem
63 Params...: s1-destination, s2-source
64 */
65 void sys_set_free(int (*func)(void *));
66
67
68
69 /*
70 Procedure...: sys_alloc_mem
71 Description...: Allocates a block of memory (similar to malloc)
72 Params...: Number of bytes to allocate
73 */
74 void *sys_alloc_mem(u32int size);
75
76 /*
77 Procedure...: sys_free_mem
78 Description...: Frees memory
79 Params...: Pointer to block of memory to free
80 */
81 int sys_free_mem(void *ptr);
82
83 /*
84 Procedure...: idle
85 Description...: The idle process
86 Params...: None
87 */
88 void idle();
89
90 #endif

```

6.35 /home/maximillian/Desktop/MAMA/README.md File Reference

6.36 /home/maximillian/Desktop/MAMA/term/args.c File Reference

```

#include "commhand.h"
#include "utils.h"
#include "args.h"
#include "syntax.h"
#include <lib/out.h>
#include <include/string.h>

```

Macros

- `#define MAX_PARSE_STACK_SIZE 2`

Functions

- `int get_token (char **, char *, int)`
- `int stack_empty ()`
- `enum SyntaxState stack_peek ()`
- `void stack_push (enum SyntaxState)`
- `void stack_pop ()`
- `parsed_args * parse_args (char *arg_str)`
- `int named_arg (parsed_args *args, char *arg_name, char **arg_val)`
- `int flag (parsed_args *args, char *flag_name)`
- `int next_unnamed_arg (parsed_args *args, char **arg_val)`

Variables

- `enum SyntaxState parse_stack [MAX_PARSE_STACK_SIZE]`
- `int stack_size = 0`
- `enum SyntaxState last_state`
- `enum SyntaxState cur_state`

6.36.1 Macro Definition Documentation

6.36.1.1 MAX_PARSE_STACK_SIZE

```
#define MAX_PARSE_STACK_SIZE 2
```

6.36.2 Function Documentation

6.36.2.1 flag()

```
int flag (  
    parsed_args * args,  
    char * flag_name )
```

6.36.2.2 `get_token()`

```
int get_token (
    char ** arg_str,
    char * token,
    int max_token_len )
```

6.36.2.3 `named_arg()`

```
int named_arg (
    parsed\_args * args,
    char * arg_name,
    char ** arg_val )
```

6.36.2.4 `next_unnamed_arg()`

```
int next_unnamed_arg (
    parsed\_args * args,
    char ** arg_val )
```

6.36.2.5 `parse_args()`

```
parsed\_args * parse_args (
    char * arg_str )
```

6.36.2.6 `stack_empty()`

```
int stack_empty ( )
```

6.36.2.7 `stack_peek()`

```
enum SyntaxState stack_peek ( )
```

6.36.2.8 `stack_pop()`

```
void stack_pop ( )
```

6.36.2.9 `stack_push()`

```
void stack_push (
    enum SyntaxState state )
```

6.36.3 Variable Documentation

6.36.3.1 `cur_state`

```
enum SyntaxState cur_state
```

6.36.3.2 `last_state`

```
enum SyntaxState last_state
```

6.36.3.3 `parse_stack`

```
enum SyntaxState parse_stack[MAX\_PARSE\_STACK\_SIZE]
```

6.36.3.4 `stack_size`

```
int stack_size = 0
```

6.37 /home/maximillian/Desktop/MAMA/term/args.h File Reference

Classes

- struct [parsed_args](#)

Typedefs

- typedef struct [parsed_args](#) [parsed_args](#)

Functions

- [parsed_args](#) * [parse_args](#) (char *)

6.37.1 Typedef Documentation

6.37.1.1 [parsed_args](#)

```
typedef struct parsed\_args parsed\_args
```

6.37.2 Function Documentation

6.37.2.1 [parse_args\(\)](#)

```
parsed\_args * parse\_args (
    char * arg_str )
```

6.38 args.h

[Go to the documentation of this file.](#)

```
1 #ifndef ARGS_H
2 #define ARGS_H
3
4 typedef struct parsed\_args {
5     int flag_count;
6     int named_arg_count;
7     int unnamed_arg_count;
8     int unnamed_args_used_so_far;
9
10     char flags[MAX_CMD_FLAG_COUNT][MAX_CMD_ARG_NAME_LEN + 1];
11     char named_arg_names[MAX_CMD_NAMED_ARG_COUNT][MAX_CMD_ARG_NAME_LEN + 1];
12     char named_arg_values[MAX_CMD_NAMED_ARG_COUNT][MAX_CMD_ARG_VALUE_LEN + 1];
13     char unnamed_args[MAX_CMD_UNNAMED_ARG_COUNT][MAX_CMD_ARG_VALUE_LEN + 1];
14 } parsed\_args;
15
16 parsed\_args *parse\_args(char *);
17
18 #endif
```

6.39 /home/maximillian/Desktop/MAMA/term/ascii/mama.c File Reference

```
#include "mama.h"
```


Functions

- void `mama` ()
mama ascii art

6.39.1 Function Documentation

6.39.1.1 mama()

```
void mama ( )
```

mama ascii art

One of the intro ascii art.

6.40 /home/maximillian/Desktop/MAMA/term/ascii/mama.h File Reference

Functions

- void `mama` ()
mama ascii art

6.40.1 Function Documentation

6.40.1.1 mama()

```
void mama ( )
```

mama ascii art

One of the intro ascii art.

6.41 mama.h

[Go to the documentation of this file.](#)

```
1  
7 void mama ();
```

6.42 /home/maximillian/Desktop/MAMA/term/cmds/argtest.c File Reference

```
#include "../args.h"
#include "../args.c"
#include <lib/out.h>
```

Functions

- int [cmd_argtest](#) (char *arg_str)

6.42.1 Function Documentation

6.42.1.1 cmd_argtest()

```
int cmd_argtest (
    char * arg_str )
```

6.43 /home/maximillian/Desktop/MAMA/term/cmds/echo.c File Reference

```
#include <lib/out.h>
```

Functions

- int [cmd_echo](#) (char *arg_str)

6.43.1 Function Documentation

6.43.1.1 cmd_echo()

```
int cmd_echo (
    char * arg_str )
```

6.44 /home/maximillian/Desktop/MAMA/help.c File Reference

```
#include <lib/out.h>
```

Functions

- int [cmd_help](#) (char *command)
Prints help message for command.
- void [helpList](#) ()
Displays a list of common system commands.
- void [shutdownHelp](#) ()
Help page for the shutdown command.
- void [helpHelp](#) ()
Help page for the help command.
- void [setdateHelp](#) ()
Help page for the [setdate\(\)](#) method.
- void [getdateHelp](#) ()
Help page for the [getdate\(\)](#) method.
- void [gettimeHelp](#) ()
Help page for [gettime\(\)](#) method.
- void [settimeHelp](#) ()
Help page for [settime\(\)](#) method.
- void [versionOs](#) ()

6.44.1 Function Documentation

6.44.1.1 cmd_help()

```
int cmd_help (
    char * command )
```

Prints help message for command.

Prints out a help message and basic syntax for a specific command

Parameters

<i>command</i>	Command which the user needs basic information and syntax for
----------------	---

Returns

1 upon success, -1 upon error

6.44.1.2 getdateHelp()

```
void getdateHelp ( )
```

Help page for the [getdate\(\)](#) method.

Prints out the name, usage, return and description for the [getdate\(\)](#) method.

6.44.1.3 `gettimeHelp()`

```
void gettimeHelp ( )
```

Help page for [`gettime\(\)`](#) method.

Prints out the name, usage, return and description for the [`gettime\(\)`](#) method.

6.44.1.4 `helpHelp()`

```
void helpHelp ( )
```

Help page for the help command.

Prints out the name, usage, return and description for the help command.

6.44.1.5 `helpList()`

```
void helpList ( )
```

Displays a list of common system commands.

Displays a list of common system commands for the user.

6.44.1.6 `setdateHelp()`

```
void setdateHelp ( )
```

Help page for the [`setdate\(\)`](#) method.

Prints out the name, usage, and description for the [`setdate\(\)`](#) method.

6.44.1.7 `settimeHelp()`

```
void settimeHelp ( )
```

Help page for [`settime\(\)`](#) method.

Prints out the name, usage, and description for the [`settime\(\)`](#) method.

6.44.1.8 `shutdownHelp()`

```
void shutdownHelp ( )
```

Help page for the shutdown command.

Prints out the name, usage, and description for the shutdown system command.

6.44.1.9 versionOs()

```
void versionOs ( )
```

6.45 /home/maximillian/Desktop/MAMA/term/cmds/help.c File Reference

```
#include <lib/out.h>
```

Functions

- int [cmd_help](#) (char *command)
Prints help message for command.
- void [versionHelp](#) ()
Help page for the version command.
- void [helpList](#) ()
Displays a list of common system commands.
- void [shutdownHelp](#) ()
Help page for the shutdown command.
- void [helpHelp](#) ()
Help page for the help command.
- void [setdateHelp](#) ()
Help page for the [setdate\(\)](#) method.
- void [getdateHelp](#) ()
Help page for the [getdate\(\)](#) method.
- void [gettimeHelp](#) ()
Help page for [gettime\(\)](#) method.
- void [settimeHelp](#) ()
Help page for [settime\(\)](#) method.
- void [createpcbHelp](#) ()
Help page for createpcb.
- void [deletepcbHelp](#) ()
Help page for deletepcb.
- void [showpcbHelp](#) ()
Help page for showpcb.
- void [showallpcbHelp](#) ()
Help page for showallpcb.
- void [showreadypcbHelp](#) ()
Help page for showreadypcb.
- void [showblockedpcbHelp](#) ()
Help page for showblockedpcb.
- void [blockHelp](#) ()
Help page for block.
- void [unblockHelp](#) ()
Help page for unblock.
- void [setpriorityHelp](#) ()
Help page for setpriority.
- void [resumeHelp](#) ()
Help page for resume.
- void [suspendHelp](#) ()
Help page for suspend.

6.45.1 Function Documentation

6.45.1.1 blockHelp()

```
void blockHelp ( )
```

Help page for block.

Displays the block help page

6.45.1.2 cmd_help()

```
int cmd_help (
    char * command )
```

Prints help message for command.

Prints out a help message and basic syntax for a specific command

Parameters

<i>command</i>	Command which the user needs basic information and syntax for
----------------	---

Returns

1 upon success, -1 upon error

6.45.1.3 createpcbHelp()

```
void createpcbHelp ( )
```

Help page for createpcb.

Displays the createpcb help page

6.45.1.4 deletepcbHelp()

```
void deletepcbHelp ( )
```

Help page for deletepcb.

Displays the deletepcb help page

6.45.1.5 getdateHelp()

```
void getdateHelp ( )
```

Help page for the [getdate\(\)](#) method.

Prints out the name, usage, return and description for the [getdate\(\)](#) method.

6.45.1.6 gettimeHelp()

```
void gettimeHelp ( )
```

Help page for [gettime\(\)](#) method.

Prints out the name, usage, return and description for the [gettime\(\)](#) method.

6.45.1.7 helpHelp()

```
void helpHelp ( )
```

Help page for the help command.

Prints out the name, usage, return and description for the help command.

6.45.1.8 helpList()

```
void helpList ( )
```

Displays a list of common system commands.

Displays a list of common system commands for the user.

6.45.1.9 resumeHelp()

```
void resumeHelp ( )
```

Help page for resume.

Displays the resume help page

6.45.1.10 setdateHelp()

```
void setdateHelp ( )
```

Help page for the [setdate\(\)](#) method.

Prints out the name, usage, and description for the [setdate\(\)](#) method.

6.45.1.11 `setpriorityHelp()`

```
void setpriorityHelp ( )
```

Help page for `setpriority`.

Displays the `setpriority` help page

6.45.1.12 `settimeHelp()`

```
void settimeHelp ( )
```

Help page for [`settime\(\)`](#) method.

Prints out the name, usage, and description for the [`settime\(\)`](#) method.

6.45.1.13 `showallpcbHelp()`

```
void showallpcbHelp ( )
```

Help page for `showallpcb`.

Displays the `showallpcb` help page

6.45.1.14 `showblockedpcbHelp()`

```
void showblockedpcbHelp ( )
```

Help page for `showblockedpcb`.

Displays the `showblockedpcb` help page

6.45.1.15 `showpcbHelp()`

```
void showpcbHelp ( )
```

Help page for `showpcb`.

Displays the `showpcb` help page

6.45.1.16 `showreadypcbHelp()`

```
void showreadypcbHelp ( )
```

Help page for `showreadypcb`.

Displays the `showreadypcb` help page

6.45.1.17 shutdownHelp()

```
void shutdownHelp ( )
```

Help page for the shutdown command.

Prints out the name, usage, and description for the shutdown system command.

6.45.1.18 suspendHelp()

```
void suspendHelp ( )
```

Help page for suspend.

Displays the suspend help page

6.45.1.19 unblockHelp()

```
void unblockHelp ( )
```

Help page for unblock.

Displays te unblock help page

6.45.1.20 versionHelp()

```
void versionHelp ( )
```

Help page for the version command.

Displays the current version of the system.

6.46 /home/maximillian/Desktop/MAMA/term/cmds/shutdown.c File Reference

```
#include <lib/out.h>
```

Functions

- int [cmd_shutdown](#) (char *arg_str)
Handler for calls to the shutdown command.

6.46.1 Function Documentation

6.46.1.1 cmd_shutdown()

```
int cmd_shutdown (
    char * arg_str )
```

Handler for calls to the shutdown command.

Prompts for user confirmation before shutting the system down.

Parameters

<i>arg_str</i>	The arguments passed to the shutdown command. Unused by the handler.
----------------	--

Returns

The exit code of the command, indicating whether or not the user confirmed the request to shutdown the system. Returns 0 if the user confirmed the request, 1 otherwise.

6.47 /home/maximillian/Desktop/MAMA/term/cmds/version.c File Reference

```
#include <lib/out.h>
```

Functions

- int [cmd_version](#) (char *arg_str)
Handler for the version command.

6.47.1 Function Documentation

6.47.1.1 cmd_version()

```
int cmd_version (  
    char * arg_str )
```

Handler for the version command.

Prints the current version of the operating system.

Parameters

<i>arg_str</i>	The arguments passed to the version command. Unused by the handler.
----------------	---

Returns

The exit code of the command, always 0.

6.48 /home/maximillian/Desktop/MAMA/term/commands.h File Reference

```
#include "cmds/help.c"
```

```
#include "cmds/shutdown.c"
#include "cmds/echo.c"
#include "cmds/version.c"
#include "cmds/argtest.c"
#include "cmds/pcb.c"
```

6.49 commands.h

[Go to the documentation of this file.](#)

```
1 #ifndef COMMANDS_H
2 #define COMMANDS_H
3
4 #include "cmds/help.c"
5 #include "cmds/shutdown.c"
6 #include "cmds/echo.c"
7 #include "cmds/version.c"
8 #include "cmds/argtest.c"
9 #include "cmds/pcb.c"
10
11 #endif
```

6.50 /home/maximillian/Desktop/MAMA/term/commhand.c File Reference

```
#include <include/string.h>
#include <modules/mpx_supt.h>
#include "visuals/colorize.c"
#include "history.c"
#include "commhand.h"
#include "commands.h"
#include "visuals/syntax_highlight.h"
#include "visuals/hints.h"
#include <lib/out.c>
#include "dnt/dnt.c"
#include "utils.h"
#include "ascii/mama.c"
#include "pcb/pcb.c"
```

Classes

- struct [cmd_mapping](#)

Typedefs

- typedef int(* [cmd_func_t](#)) (char *)
- typedef struct [cmd_mapping cmd_mapping](#)

Functions

- int `is_name_char` (char)
Returns whether or not the specified character is a valid character in an identifier, such as a command or argument name.
- void `extract_cmd_name` (char *, char *, int *, int *)
- `cmd_func_t` `fetch_cmd_handler` (char *)
- int `commhand` ()
Displays command line and interprets inputted commands.

Variables

- const `cmd_mapping` `cmd_mappings` []

6.50.1 Typedef Documentation

6.50.1.1 `cmd_func_t`

```
typedef int (* cmd_func_t) (char *)
```

6.50.1.2 `cmd_mapping`

```
typedef struct cmd_mapping cmd_mapping
```

6.50.2 Function Documentation

6.50.2.1 `commhand()`

```
int commhand ( )
```

Displays command line and interprets inputted commands.

Parses through the input that was polled from the command line and interprets the command that was inputted (typically the first word)

Returns

Returns 0 upon success, -1 upon error

6.50.2.2 extract_cmd_name()

```
void extract_cmd_name (
    char * cmd_str,
    char * cmd_name,
    int * cmd_name_len,
    int * args_start_index )
```

6.50.2.3 fetch_cmd_handler()

```
cmd_func_t fetch_cmd_handler (
    char * cmd_name )
```

6.50.2.4 is_name_char()

```
int is_name_char (
    char c )
```

Returns whether or not the specified character is a valid character in an identifier, such as a command or argument name.

Parameters

<code>c</code>	The character to test.
----------------	------------------------

Returns

True if the specified character `c` is valid in an identifier, false otherwise.

6.50.3 Variable Documentation

6.50.3.1 cmd_mappings

```
const cmd_mapping cmd_mappings[]
```

6.51 /home/maximillian/Desktop/MAMA/term/commhand.h File Reference

Macros

- #define `MAX_CMD_STRING_LEN` 100

- `#define MAX_CMD_NAME_LEN 30`
- `#define MAX_CMD_HIST_LEN 20`
- `#define MAX_CMD_ARG_NAME_LEN 30`
- `#define MAX_CMD_ARG_VALUE_LEN 40`
- `#define MAX_CMD_FLAG_COUNT 10`
- `#define MAX_CMD_NAMED_ARG_COUNT 10`
- `#define MAX_CMD_UNNAMED_ARG_COUNT 10`

Functions

- `int commhand ()`
Displays command line and interprets inputted commands.

6.51.1 Macro Definition Documentation

6.51.1.1 MAX_CMD_ARG_NAME_LEN

```
#define MAX_CMD_ARG_NAME_LEN 30
```

6.51.1.2 MAX_CMD_ARG_VALUE_LEN

```
#define MAX_CMD_ARG_VALUE_LEN 40
```

6.51.1.3 MAX_CMD_FLAG_COUNT

```
#define MAX_CMD_FLAG_COUNT 10
```

6.51.1.4 MAX_CMD_HIST_LEN

```
#define MAX_CMD_HIST_LEN 20
```

6.51.1.5 MAX_CMD_NAME_LEN

```
#define MAX_CMD_NAME_LEN 30
```

6.51.1.6 MAX_CMD_NAMED_ARG_COUNT

```
#define MAX_CMD_NAMED_ARG_COUNT 10
```

6.51.1.7 MAX_CMD_STRING_LEN

```
#define MAX_CMD_STRING_LEN 100
```

6.51.1.8 MAX_CMD_UNNAMED_ARG_COUNT

```
#define MAX_CMD_UNNAMED_ARG_COUNT 10
```

6.51.2 Function Documentation**6.51.2.1 commhand()**

```
int commhand ( )
```

Displays command line and interprets inputted commands.

Parses through the input that was polled from the command line and interprets the command that was inputted (typically the first word)

Returns

Returns 0 upon success, -1 upon error

6.52 commhand.h

[Go to the documentation of this file.](#)

```
1 /* the logic for each command the user has to run is contained in a separate file in term/cmds
2  * each file should contain a function to run this command and possibly any helper functions the command
   needs to run
3  * include each of these files below - make sure to add an #include directive if you write a new command
4  */
5 #ifndef COMMHAND_H
6 #define COMMHAND_H
7
8 #define MAX_CMD_STRING_LEN 100
9 #define MAX_CMD_NAME_LEN 30
10 #define MAX_CMD_HIST_LEN 20
11 #define MAX_CMD_ARG_NAME_LEN 30
12 #define MAX_CMD_ARG_VALUE_LEN 40
13 #define MAX_CMD_FLAG_COUNT 10
14 #define MAX_CMD_NAMED_ARG_COUNT 10
15 #define MAX_CMD_UNNAMED_ARG_COUNT 10
16
17 int commhand();
18 #endif
```

6.53 /home/maximillian/Desktop/MAMA/term/dnt/dnt.c File Reference

```
#include "dnt.h"
```

Functions

- int [setdate](#) (char *date)
Sets the date of the system.
- int [setDateInMemory](#) (int month, int day, int year)
Sets the date in memory.
- int [getdate](#) (char *p)
Gets the date of the system.
- int [settime](#) (char *time)
Sets the time of the system.
- void [setTimeInMemory](#) (int hour, int minute, int second)
Sets the time into memory.
- int [gettime](#) (char *p)
Gets the system time.
- unsigned char [ltoBCD](#) (unsigned int value)
Converts 32-bit integer to 8-bit BCD.
- unsigned int [BCDtol](#) (unsigned char value)
Converts 8-bit BCD to 32-bit integer.
- char * [intToMonth](#) (int value)
Converts masked int into string month.
- char * [intToDayOfWeek](#) (int value)
Converts integer to string day of the week.
- int [daysInMonth](#) (int month, int year)
Calculates the number of days in a month.

6.53.1 Function Documentation

6.53.1.1 BCDtol()

```
unsigned int BCDtoI (
    unsigned char value )
```

Converts 8-bit BCD to 32-bit integer.

Converts an 8-bit BCD unsigned char to a 32-bit unsigned integer.

Parameters

<i>value</i>	8-bit BCD value that will be converted to 32-bit int
--------------	--

Returns

Returns 32-bit unsigned int

6.53.1.2 daysInMonth()

```
int daysInMonth (
    int month,
    int year )
```

Calculates the number of days in a month.

Calculates the number of days in the month based upon which month it is. If year is divisible by four then it is a leap year and will add 1 day for February for a total of 29 days. Otherwise, February will be 28 days.

Parameters

<i>month</i>	The month in the year (January = 1...December = 12)
<i>year</i>	The year that was being set

Returns

Returns the number of days in the month

6.53.1.3 getdate()

```
int getdate (
    char * p )
```

Gets the date of the system.

Returns a string that represents the current date of the system. This is in the format DayOfWeek, Month Day, Year
Ex: Wednesday, August 25, 2021

Parameters

<i>p</i>	Empty parameter that is required to call this method. Does not do anything.
----------	---

Returns

Returns 1 upon success, -1 upon error

6.53.1.4 `gettextime()`

```
int gettextime (
    char * p )
```

Gets the system time.

Gets the system time from memory by reading from the corresponding memory address. Time will be writtin to the interface in the syntax of Hour:Minute:Second Ex: 10:06:23

Parameters

<i>Empty</i>	parameter that does not do anything. Required in order to call from commhand
--------------	--

Returns

Returns 1 upon success, -1 upon error

6.53.1.5 `intToDayOfWeek()`

```
char * intToDayOfWeek (
    int value )
```

Converts integer to string day of the week.

Converts a masked integer into an unmasked string day of the week. The days of the week are Sunday to Saturday and are 1 to 7 respectivley. 1 = Sunday 2 = Monday ... 7 = Saturday

Parameters

<i>value</i>	The masked integer value of month
--------------	-----------------------------------

Returns

Returns the unasked string value of month

6.53.1.6 `intToMonth()`

```
char * intToMonth (
    int value )
```

Converts masked int into string month.

Converts integer to a string month.

Converts the masked integer value of month and converts it into an unmasked string of month

Parameters

<i>value</i>	Masked integer month
--------------	----------------------

Returns

Returns unmasked string of month

6.53.1.7 ItoBCD()

```
unsigned char ItoBCD (
    unsigned int value )
```

Converts 32-bit integer to 8-bit BCD.

Uses basic arithmetic and bit shifting to convert from 32-bit integer to 8-bit BCD.

Parameters

<i>value</i>	The 32-bit integer that is converted to BCD
--------------	---

Returns

8-bit BCD number as an unsigned char

6.53.1.8 setdate()

```
int setdate (
    char * date )
```

Sets the date of the system.

Parses the parameter to setdate, breaking the parameter into month, day and year before passing it to setDateInMemory. The basic syntax is month.day.year

Parameters

<i>date</i>	The parameter that is passed with setdate. This string is parsed and each segment is converted to a 32-bit int.
-------------	---

Returns

Returns 1 upon success, -1 upon error

6.53.1.9 setDateInMemory()

```
int setDateInMemory (
    int month,
    int day,
    int year )
```

Sets the date in memory.

Sets the date in memory by assigning the values to the appropriate places in memory. This method is called by the setdate method.

Parameters

<i>month</i>	The month (1 = January ... 12 = December)
<i>day</i>	The day in the month. Can be between 0 and 32
<i>year</i>	The current year. This is a 2-digit number

Returns

Returns 1 upon success, -1 upon error

6.53.1.10 settime()

```
int settime (
    char * time )
```

Sets the time of the system.

Takes the parameter which will be parsed into 32-bit int (later converted to BCD) and sets it into memory. The syntax is Hour.Minute.Second Ex: 10.23.00

Parameters

<i>The</i>	parameter passed with the settime call
------------	--

Returns

Returns 1 upon success, -1 upon error

6.53.1.11 setTimeInMemory()

```
void setTimeInMemory (
    int hour,
    int minute,
    int second )
```

Sets the time into memory.

This method is called by the settime method. Writes the data into memory. First converts all parameter from 32-bit int to 8-bit BCD and then writes to the appropriate address.

Parameters

<i>hour</i>	32-bit int hour
<i>minute</i>	32-bit int minute
<i>second</i>	32-bit int second

6.54 /home/maximillian/Desktop/MAMA/term/dnt/dnt.h File Reference

Macros

- #define `MAX_HOURS` 23
The largest value that the user can set their hours to.
- #define `MAX_MINUTES` 59
The largest value that the user can set their minutes to.
- #define `MAX_SECONDS` 59
The largest value that the user can set their seconds to.
- #define `MAX_YEAR` 99
The largest value that the user can set their year to.
- #define `MAX_MONTH` 12
The largest value that the user can set their month to.
- #define `MAX_DAY` 31
The largest value that the user can set their day to.
- #define `MIN_YEAR` 10
Minimum year that can be set in memory.
- #define `MIN_MONTH` 1
Minimum month that can be set in memory.
- #define `MIN_DAY` 1
Minimum day that can be set in memory.
- #define `EPOCH_YEAR` 1970
Unix Epoch year.
- #define `EPOCH_FIRST_DAY_OF_YEAR` 1
Unix Epoch first day of the year.
- #define `EPOCH_FIRST_MONTH_OF_YEAR` 1
Unix Epoch first month of the year.
- #define `EPOCH_FIRST_DAY_OF_WEEK_OF_YEAR` 5
Unix Epoch first day of the week in the year.
- #define `DAYS_IN_YEAR` 365
Number of days in a normal year.
- #define `DAYS_IN_LEAP_YEAR` 366
Number of days in a leap year.
- #define `MIN` 0
Minimum value that can be set for hours, minutes, and seconds.

Functions

- int `setdate` (char *date)
Sets the date of the system.
- int `setDateInMemory` (int month, int day, int year)
Sets the date in memory.
- int `getdate` (char *p)
Gets the date of the system.
- int `settime` (char *time)
Sets the time of the system.
- void `setTimeInMemory` (int hour, int minute, int second)
Sets the time into memory.
- int `gettime` (char *p)
Gets the system time.
- unsigned char `ltoBCD` (unsigned int value)
Converts 32-bit integer to 8-bit BCD.
- unsigned int `BCDtol` (unsigned char value)
Converts 8-bit BCD to 32-bit integer.
- char * `intToMonth` (int value)
Converts integer to a string month.
- char * `intToDayOfWeek` (int value)
Converts integer to string day of the week.
- int `daysInMonth` (int month, int year)
Calculates the number of days in a month.

6.54.1 Macro Definition Documentation

6.54.1.1 DAYS_IN_LEAP_YEAR

```
#define DAYS_IN_LEAP_YEAR 366
```

Number of days in a leap year.

6.54.1.2 DAYS_IN_YEAR

```
#define DAYS_IN_YEAR 365
```

Number of days in a normal year.

6.54.1.3 EPOCH_FIRST_DAY_OF_WEEK_OF_YEAR

```
#define EPOCH_FIRST_DAY_OF_WEEK_OF_YEAR 5
```

Unix Epoch first day of the week in the year.

6.54.1.4 EPOCH_FIRST_DAY_OF_YEAR

```
#define EPOCH_FIRST_DAY_OF_YEAR 1
```

Unix Epoch first day of the year.

6.54.1.5 EPOCH_FIRST_MONTH_OF_YEAR

```
#define EPOCH_FIRST_MONTH_OF_YEAR 1
```

Unix Epoch first month of the year.

6.54.1.6 EPOCH_YEAR

```
#define EPOCH_YEAR 1970
```

Unix Epoch year.

6.54.1.7 MAX_DAY

```
#define MAX_DAY 31
```

The largest value that the user can set their day to.

6.54.1.8 MAX_HOURS

```
#define MAX_HOURS 23
```

The largest value that the user can set their hours to.

6.54.1.9 MAX_MINUTES

```
#define MAX_MINUTES 59
```

The largest value that the user can set their minutes to.

6.54.1.10 MAX_MONTH

```
#define MAX_MONTH 12
```

The largest value that the user can set their month to.

6.54.1.11 MAX_SECONDS

```
#define MAX_SECONDS 59
```

The largest value that the user can set their seconds to.

6.54.1.12 MAX_YEAR

```
#define MAX_YEAR 99
```

The largest value that the user can set their year to.

6.54.1.13 MIN

```
#define MIN 0
```

Minimum value that can be set for hours, minutes, and seconds.

6.54.1.14 MIN_DAY

```
#define MIN_DAY 1
```

Minimum day that can be set in memory.

6.54.1.15 MIN_MONTH

```
#define MIN_MONTH 1
```

Minimum month that can be set in memory.

6.54.1.16 MIN_YEAR

```
#define MIN_YEAR 10
```

Minimum year that can be set in memory.

6.54.2 Function Documentation

6.54.2.1 BCDtoI()

```
unsigned int BCDtoI (  
    unsigned char value )
```

Converts 8-bit BCD to 32-bit integer.

Converts an 8-bit BCD unsigned char to a 32-bit unsigned integer.

Parameters

<i>value</i>	8-bit BCD value that will be converted to 32-bit int
--------------	--

Returns

Returns 32-bit unsigned int

6.54.2.2 daysInMonth()

```
int daysInMonth (  
    int month,  
    int year )
```

Calculates the number of days in a month.

Calculates the number of days in the month based upon which month it is. If year is divisible by four then it is a leap year and will add 1 day for February for a total of 29 days. Otherwise, February will be 28 days.

Parameters

<i>month</i>	The month in the year (January = 1...December = 12)
<i>year</i>	The year that was being set

Returns

Returns the number of days in the month

6.54.2.3 getdate()

```
int getdate (
    char * p )
```

Gets the date of the system.

Returns a string that represents the current date of the system. This is in the format DayOfWeek, Month Day, Year
Ex: Wednesday, August 25, 2021

Parameters

<i>p</i>	Empty parameter that is required to call this method. Does not do anything.
----------	---

Returns

Returns 1 upon success, -1 upon error

6.54.2.4 gettime()

```
int gettime (
    char * p )
```

Gets the system time.

Gets the system time from memory by reading from the corresponding memory address. Time will be written to the interface in the syntax of Hour:Minute:Second Ex: 10:06:23

Parameters

<i>Empty</i>	parameter that does not do anything. Required in order to call from command
--------------	---

Returns

Returns 1 upon success, -1 upon error

6.54.2.5 intToDayOfWeek()

```
char * intToDayOfWeek (
    int value )
```

Converts integer to string day of the week.

Converts a masked integer into an unmasked string day of the week. The days of the week are Sunday to Saturday and are 1 to 7 respectively. 1 = Sunday 2 = Monday ... 7 = Saturday

Parameters

<i>value</i>	The masked integer value of month
--------------	-----------------------------------

Returns

Returns the unmasked string value of month

6.54.2.6 intToMonth()

```
char * intToMonth (
    int value )
```

Converts integer to a string month.

Converts masked int into string month.

Converts a masked integer into an unmasked string month. The months are January to December and are 1 to 12 respectively. 1 = January 2 = February ... 12 = December

Parameters

<i>value</i>	The masked month
--------------	------------------

Returns

Returns unmasked string month

Converts the masked integer value of month and converts it into an unmasked string of month

Parameters

<i>value</i>	Masked integer month
--------------	----------------------

Returns

Returns unmasked string of month

Converts integer to a string month.

Converts the masked integer value of month and converts it into an unmasked string of month

Parameters

<i>value</i>	Masked integer month
--------------	----------------------

Returns

Returns unmasked string of month

6.54.2.7 ItoBCD()

```
unsigned char ItoBCD (
    unsigned int value )
```

Converts 32-bit integer to 8-bit BCD.

Uses basic arithmetic and bit shifting to convert from 32-bit integer to 8-bit BCD.

Parameters

<i>value</i>	The 32-bit integer that is converted to BCD
--------------	---

Returns

8-bit BCD number as an unsigned char

6.54.2.8 setdate()

```
int setdate (
    char * date )
```

Sets the date of the system.

Parses the parameter to setdate, breaking the parameter into month, day and year before passing it to setDateInMemory. The basic syntax is month.day.year

Parameters

<i>date</i>	The parameter that is passed with setdate. This string is parsed and each segment is converted to a 32-bit int.
-------------	---

Returns

Returns 1 upon success, -1 upon error

6.54.2.9 setDateInMemory()

```
int setDateInMemory (
    int month,
    int day,
    int year )
```

Sets the date in memory.

Sets the date in memory by assigning the values to the appropriate places in memory. This method is called by the setdate method.

Parameters

<i>month</i>	The month (1 = January ... 12 = December)
<i>day</i>	The day in the month. Can be between 0 and 32
<i>year</i>	The current year. This is a 2-digit number

Returns

Returns 1 upon success, -1 upon error

6.54.2.10 settime()

```
int settime (
    char * time )
```

Sets the time of the system.

Takes the parameter which will be parsed into 32-bit int (later converted to BCD) and sets it into memory. The syntax is Hour.Minute.Second Ex: 10.23.00

Parameters

<i>The</i>	parameter passed with the settime call
------------	--

Returns

Returns 1 upon success, -1 upon error

6.54.2.11 setTimeInMemory()

```
void setTimeInMemory (
    int hour,
    int minute,
    int second )
```

Sets the time into memory.

This method is called by the settime method. Writes the data into memory. First converts all parameter from 32-bit int to 8-bit BCD and then writes to the appropriate address.

Parameters

<i>hour</i>	32-bit int hour
<i>minute</i>	32-bit int minute
<i>second</i>	32-bit int second

6.55 dnt.h

[Go to the documentation of this file.](#)

```
1
2 #define MAX_HOURS 23
4 #define MAX_MINUTES 59
6 #define MAX_SECONDS 59
7
9 #define MAX_YEAR 99
11 #define MAX_MONTH 12
13 #define MAX_DAY 31
14
16 #define MIN_YEAR 10
18 #define MIN_MONTH 1
20 #define MIN_DAY 1
21
23 #define EPOCH_YEAR 1970
25 #define EPOCH_FIRST_DAY_OF_YEAR 1
27 #define EPOCH_FIRST_MONTH_OF_YEAR 1
29 #define EPOCH_FIRST_DAY_OF_WEEK_OF_YEAR 5
31 #define DAYS_IN_YEAR 365
33 #define DAYS_IN_LEAP_YEAR 366
34
36 #define MIN 0
37
51 int setdate(char * date);
52
66 int setDateInMemory(int month,int day,int year);
67
80 int getdate(char * p);
81
94 int settime(char * time);
95
106 void setTimeInMemory(int hour, int minute, int second);
107
120 int gettime(char * p);
121
132 unsigned char ItoBCD(unsigned int value);
133
144 unsigned int BCDtoI(unsigned char value);
145
```

```

161 char * intToMonth(int value);
162
178 char * intToDayOfWeek(int value);
179
190 char * intToMonth(int value);
191
206 int daysInMonth(int month, int year);

```

6.56 /home/maximillian/Desktop/MAMA/term/history.c File Reference

```

#include "commhand.h"
#include "visuals/cursor.c"
#include "visuals/syntax_highlight.h"
#include <lib/out.h>

```

Functions

- int `circular_next_index` (int)

Whether or not the most recent entry in the user's command history has been discarded by calling `hist_discard_↔last_frame`.
- int `circular_prev_index` (int i)

Returns the index immediately preceding the specified index in `cmd_hist`, an array-based circular queue containing entries in the user's command history.
- void `write_hist_to_buf` (char *buf, int *index, int *len)

Writes the history entry pointed to by `cmd_hist_current_index` to the specified buffer and prints the new buffer to the terminal.
- void `hist_rewind` (char *internal_buf, int *internal_index, int *internal_buf_len)

Moves backwards 1 entry in the user's command history.
- void `hist_forward` (char *internal_buf, int *internal_index, int *internal_buf_len)

Moves forwards 1 entry in the user's command history.
- void `hist_discard_last_frame` ()

Removes the most recent command input from the user from the user's command history.
- char * `hist_next_frame` ()

Requests a buffer to write user input to that will become the most recent entry in the user's command history.

6.56.1 Function Documentation

6.56.1.1 `circular_next_index()`

```

int circular_next_index (
    int i )

```

Whether or not the most recent entry in the user's command history has been discarded by calling `hist_discard_↔last_frame`.

Returns the index immediately following the specified index in `cmd_hist`, an array-based circular queue containing entries in the user's command history.

Parameters

<i>i</i>	An index in cmd_hist.
----------	-----------------------

Returns

The index of the slot immediately following the slot at index *i* in cmd_hist.

6.56.1.2 circular_prev_index()

```
int circular_prev_index (
    int i )
```

Returns the index immediately preceding the specified index in cmd_hist, an array-based circular queue containing entries in the user's command history.

Parameters

<i>i</i>	An index in cmd_hist.
----------	-----------------------

Returns

The index of the slot immediately preceding the slot at index *i* in cmd_hist.

6.56.1.3 hist_discard_last_frame()

```
void hist_discard_last_frame ( )
```

Removes the most recent command input from the user from the user's command history.

6.56.1.4 hist_forward()

```
void hist_forward (
    char * internal_buf,
    int * internal_index,
    int * internal_buf_len )
```

Moves forwards 1 entry in the user's command history.

Parameters

<i>internal_buf</i>	The buffer managed by low-level read operations containing user input from the terminal. Contents will be overwritten with the next entry in the user's command history.
<i>internal_index</i>	A pointer to the position of the cursor, managed by low-level read operations. The cursor position will be adjusted to point to the end of the line.
<i>internal_buf_len</i>	A pointer to the length of the buffer containing user input to the terminal. Will be adjusted to contain the length of the history entry being written to the buffer.

6.56.1.5 hist_next_frame()

```
char * hist_next_frame ( )
```

Requests a buffer to write user input to that will become the most recent entry in the user's command history.

Returns

A pointer to the first slot in a character buffer representing the next entry in the user's command history.

6.56.1.6 hist_rewind()

```
void hist_rewind (
    char * internal_buf,
    int * internal_index,
    int * internal_buf_len )
```

Moves backwards 1 entry in the user's command history.

Parameters

<i>internal_buf</i>	The buffer managed by low-level read operations containing user input from the terminal. Contents will be overwritten with the previous entry in the user's command history.
<i>internal_index</i>	A pointer to the position of the cursor, managed by low-level read operations. The cursor position will be adjusted to point to the end of the line.
<i>internal_buf_len</i>	A pointer to the length of the buffer containing user input to the terminal. Will be adjusted to contain the length of the history entry being written to the buffer.

6.56.1.7 write_hist_to_buf()

```
void write_hist_to_buf (
    char * buf,
```

```
int * index,
int * len )
```

Writes the history entry pointed to by `cmd_hist_current_index` to the specified buffer and prints the new buffer to the terminal.

Used internally by `hist_rewind` and `hist_forward`.

Parameters

<i>buf</i>	The buffer to write the current history entry to.
<i>index</i>	A pointer to the position of the cursor in the user's terminal.
<i>len</i>	A pointer to the length of the buffer.

6.57 /home/maximillian/Desktop/MAMA/term/history.h File Reference

Functions

- void [hist_rewind](#) (char *, int *, int *)
Moves backwards 1 entry in the user's command history.
- void [hist_forward](#) (char *, int *, int *)
Moves forwards 1 entry in the user's command history.
- char * [hist_next_frame](#) ()
Requests a buffer to write user input to that will become the most recent entry in the user's command history.

6.57.1 Function Documentation

6.57.1.1 hist_forward()

```
void hist_forward (
    char * internal_buf,
    int * internal_index,
    int * internal_buf_len )
```

Moves forwards 1 entry in the user's command history.

Parameters

<i>internal_buf</i>	The buffer managed by low-level read operations containing user input from the terminal. Contents will be overwritten with the next entry in the user's command history.
<i>internal_index</i>	A pointer to the position of the cursor, managed by low-level read operations. The cursor position will be adjusted to point to the end of the line.
<i>internal_buf_len</i>	A pointer to the length of the buffer containing user input to the terminal. Will be adjusted to contain the length of the history entry being written to the buffer.

6.57.1.2 hist_next_frame()

```
char * hist_next_frame ( )
```

Requests a buffer to write user input to that will become the most recent entry in the user's command history.

Returns

A pointer to the first slot in a character buffer representing the next entry in the user's command history.

6.57.1.3 hist_rewind()

```
void hist_rewind (
    char * internal_buf,
    int * internal_index,
    int * internal_buf_len )
```

Moves backwards 1 entry in the user's command history.

Parameters

<i>internal_buf</i>	The buffer managed by low-level read operations containing user input from the terminal. Contents will be overwritten with the previous entry in the user's command history.
<i>internal_index</i>	A pointer to the position of the cursor, managed by low-level read operations. The cursor position will be adjusted to point to the end of the line.
<i>internal_buf_len</i>	A pointer to the length of the buffer containing user input to the terminal. Will be adjusted to contain the length of the history entry being written to the buffer.

6.58 history.h

[Go to the documentation of this file.](#)

```
1 #ifndef HISTORY_H
2 #define HISTORY_H
3
4 void hist_rewind(char *, int *, int *);
5 void hist_forward(char *, int *, int *);
6 char *hist_next_frame();
7
8 #endif
```

6.59 /home/maximillian/Desktop/MAMA/term/cmds/pcb.c File Reference**6.60 /home/maximillian/Desktop/MAMA/term/pcb/pcb.c File Reference**

```
#include "pcb.h"
#include <modules/mpx_supt.h>
```

```
#include <include/string.h>
#include <lib/out.h>
#include <term/utils.h>
#include <term/args.h>
```

Functions

- void `initPCB` ()
Initialize PCB Queue.
- `pcb_t * allocatePCB` ()
Allocate memory for a new PCB.
- int `freePCB` (`pcb_t *pcb`)
Free's memory associated with PCB.
- `pcb_t * setupPCB` (char *name, int process_class, int priority)
Creates a PCB.
- `pcb_t * findPCB` (char *name)
Searches for PCB.
- int `insertPCB` (`pcb_t *pcb`)
Insert PCB into queue.
- int `removePCB` (`pcb_t *pcb`)
Removes PCB from Queue.
- int `createPCB` (char *args)
Create a PCB.
- int `setPriority` (char *args)
Set a new priority to a PCB.
- int `showPCB` (char *args)
Show informatino of PCB.
- int `showReady` (char *p)
Show PCBs in ready queue.
- int `showBlocked` (char *args)
Show PCBs in blocked queue.
- int `showAll` (char *args)
Show all PCBs.
- int `suspendPCB` (char *args)
Set PCB state to suspended.
- int `resumePCB` (char *args)
Set PCB state to resume.
- int `deletePCB` (char *args)
Delete PCB.
- int `blockPCB` (char *args)
Set PCB state to be blocked.
- int `unblockPCB` (char *name)
Set PCB state to unblocked.

Variables

- `pcb_queue_t p_queue`
- `pcb_queue_t f_queue`
- `pcb_queue_t * priority_queue = &p_queue`
- `pcb_queue_t * fifo_queue = &f_queue`

6.60.1 Function Documentation

6.60.1.1 allocatePCB()

```
pcb_t * allocatePCB ( )
```

Allocate memory for a new PCB.

Allocates memory for a new PCB in the stack and performs actions to initialize PCB

Returns

Pointer to newly created PCB, NULL otherwise

6.60.1.2 blockPCB()

```
int blockPCB (
    char * name )
```

Set PCB state to be blocked.

Find the PCB name in queue and sets its state to blocked and reinserts it into the appropriate queue.

Parameters

<i>name</i>	Name of PCB to block
-------------	----------------------

6.60.1.3 createPCB()

```
int createPCB (
    char * user_input )
```

Create a PCB.

Creates a new, unique PCB in memory.

Parameters

<i>name</i>	Give name of the PCB
<i>process_class</i>	The type of process class that will be used
<i>priority</i>	Priority of the PCB

Returns

Returns 1 upon success, Otherwise 0

Parse the user input

Error Handling

6.60.1.4 deletePCB()

```
int deletePCB (
    char * name )
```

Delete PCB.

Will remove a PCB from the appropriate queue and free all associated memory. Will find the PCB in the queue, unlink it and free it.

Parameters

<i>name</i>	Name of the PCB to delete
-------------	---------------------------

Returns

Return 0 upon success, 1 upon failure

6.60.1.5 findPCB()

```
pcb_t * findPCB (
    char * name )
```

Searches for PCB.

Given a PCB name, will search all queues for a process.

Parameters

<i>name</i>	Name of the PCB being searched
-------------	--------------------------------

Returns

Returns pointer to PCB upon success, NULL if PCB was not found

6.60.1.6 freePCB()

```
int freePCB (
    pcb_t * freed_pcb )
```

Free's memory associated with PCB.

Free's the memory associated with the PCB such as the stack and the PCB itself

Parameters

<i>freed_pcb</i>	Pointer to the PCB being freed
------------------	--------------------------------

Returns

Returns 1 upon success, 0 upon error

6.60.1.7 initPCB()

```
void initPCB ( )
```

Initialize PCB Queue.

Initialize the PCB queue's by assigning values for the two queues that exist. This method is called upon startup in the commhand

6.60.1.8 insertPCB()

```
int insertPCB (
    pcb_t * pcb )
```

Insert PCB into queue.

Inserts a PCB into the appropriate queue

Parameters

<i>pcb</i>	Pointer to the PCB being inserted
------------	-----------------------------------

Returns

0 on success, 1 on error

6.60.1.9 removePCB()

```
int removePCB (
    pcb_t * pcb )
```

Removes PCB from Queue.

Removes specified PCB from queue it is stored in.

Parameters

<i>pcb</i>	Pointer to the PCB being removed
------------	----------------------------------

Returns

Returns 1 upon success, 0 upon error

6.60.1.10 resumePCB()

```
int resumePCB (  
    char * name )
```

Set PCB state to resume.

Places a PCB into a not suspended state and reinserts into the appropriate queue

Parameters

<i>name</i>	Name of PCB to resume
-------------	-----------------------

Returns

Returns 0 upon success, 1 upon error

6.60.1.11 setPriority()

```
int setPriority (  
    char * args )
```

Set a new priority to a PCB.

Sets a PCB's priority and reinserts the process into the correct place in the correct queue

Parameters

<i>args</i>	Name of the PCB and new priority (PCB_NAME.PRIORITY)
-------------	--

Returns

Returns 0 upon success, 1 upon error

Parse the user input

Error Handling

6.60.1.12 setupPCB()

```
pcb_t * setupPCB (
    char * name,
    int process_class,
    int priority )
```

Creates a PCB.

Allocates and fill memory associated with the PCB being created. This is accomplished by calling `allocatePCB()` to initialize the memory and the fills the data with the parameters.

Parameters

<i>name</i>	Name of the PCB
<i>process_class</i>	Type of process being created
<i>priority</i>	The priority of the PCB being created

Returns

Returns pointer to PCB upon success, NULL otherwise

6.60.1.13 showAll()

```
int showAll (
    char * args )
```

Show all PCBs.

Display information for each PCB in the ready and blocked queue. The information that is displayed is: Process Name, Class, State, Suspended Status, Priority.

@params args Empty params

Returns

Returns 0 upon success, 1 upon error

6.60.1.14 showBlocked()

```
int showBlocked (
    char * args )
```

Show PCBs in blocked queue.

Display information for each PCB in the blocked queue. The information that is displayed is: Process Name, Class, State, Suspended Status, Priority.

6.60.1.15 showPCB()

```
int showPCB (
    char * name )
```

Show informatino of PCB.

Display information of the PCB. The information that is displayed is: Process Name, Class, State, Suspended Status and Priority

Parameters

<i>name</i>	Name of PCB to have its information displayed
-------------	---

6.60.1.16 showReady()

```
int showReady (
    char * p )
```

Show PCBs in ready queue.

Display information for each PCB in the ready queue. The information that is displayed is: Process Name, Class, State, Suspended Status, Priority

Parameters

<i>p</i>	Empty parameters.
----------	-------------------

Returns

0 upon success, 1 upon failure

6.60.1.17 suspendPCB()

```
int suspendPCB (
    char * name )
```

Set PCB state to suspended.

Places a PCB state into suspended and reinserts into appropriate queue

Parameters

<i>name</i>	Name of PCB to suspend
-------------	------------------------

Returns

Returns 0 upon success, 1 upon error

6.60.1.18 unblockPCB()

```
int unblockPCB (
    char * name )
```

Set PCB state to unblocked.

Sets PCB state into unblocked and reinserts it into the appropriate queue

Parameters

<i>name</i>	Name of the PCB to unblock
-------------	----------------------------

Returns

Returns 0 upon success, 1 upon error

6.60.2 Variable Documentation

6.60.2.1 f_queue

```
pcb_queue_t f_queue
```

6.60.2.2 fifo_queue

```
pcb_queue_t* fifo_queue = &f_queue
```

6.60.2.3 p_queue

```
pcb_queue_t p_queue
```

6.60.2.4 priority_queue

```
pcb_queue_t* priority_queue = &p_queue
```

6.61 /home/maximillian/Desktop/MAMA/term/pcb/pcb.h File Reference

Classes

- struct `pcb_t`
Process Control Block Structure.
- struct `pcb_node_t`
Individual PCB nodes. Each PCB is associated with one node.
- struct `pcb_queue`
"Master" controller of the PCB queue

Macros

- `#define MAX_STACK_SIZE 1024`
The maximum size the stack can be. May change.
- `#define MAX_PRIORITY 9`
Maximum priority a PCB can be given.
- `#define MIN_PRIORITY 0`
Minimum priority a PCB can be given.
- `#define MAX_NAME_SIZE 32`
Maximum name size that can be given to a pcb.

Typedefs

- `typedef struct pcb_node_t pcb_node_t`
Individual PCB nodes. Each PCB is associated with one node.
- `typedef struct pcb_queue pcb_queue_t`
"Master" controller of the PCB queue

Enumerations

- `enum pcb_queue_order_t { PRIORITY , FIFO }`
Type of Queue Ordering.
- `enum p_state_t {
RUNNING , READY , BLOCKED , SUSPENDED_READY ,
SUSPENDED_BLOCKED }`
Types of process states.

Functions

- `void initPCB ()`
Initialize PCB Queue.
- `pcb_t * allocatePCB ()`
Allocate memory for a new PCB.
- `int freePCB (pcb_t *freed_pcb)`
Free's memory associated with PCB.
- `pcb_t * setupPCB (char *name, int process_class, int priority)`
Creates a PCB.
- `pcb_t * findPCB (char *name)`
Searches for PCB.
- `int insertPCB (pcb_t *pcb)`
Insert PCB into queue.
- `int removePCB (pcb_t *pcb)`
Removes PCB from Queue.
- `int createPCB (char *user_input)`
Create a PCB.
- `int deletePCB (char *name)`
Delete PCB.
- `int blockPCB (char *name)`
Set PCB state to be blocked.

- int `unblockPCB` (char *name)
Set PCB state to unblocked.
- int `suspendPCB` (char *name)
Set PCB state to suspended.
- int `resumePCB` (char *name)
Set PCB state to resume.
- int `setPriority` (char *args)
Set a new priority to a PCB.
- int `showPCB` (char *name)
Show informatino of PCB.
- int `showReady` (char *p)
Show PCBs in ready queue.
- int `showBlocked` (char *args)
Show PCBs in blocked queue.
- int `showAll` (char *args)
Show all PCBs.

6.61.1 Macro Definition Documentation

6.61.1.1 MAX_NAME_SIZE

```
#define MAX_NAME_SIZE 32
```

Maximum name size that can be given to a pcb.

6.61.1.2 MAX_PRIORITY

```
#define MAX_PRIORITY 9
```

Maximum priority a PCB can be given.

6.61.1.3 MAX_STACK_SIZE

```
#define MAX_STACK_SIZE 1024
```

The maximum size the stack can be. May change.

6.61.1.4 MIN_PRIORITY

```
#define MIN_PRIORITY 0
```

Minimum priority a PCB can be given.

6.61.2 Typedef Documentation

6.61.2.1 pcb_node_t

```
typedef struct pcb_node_t pcb_node_t
```

Individual PCB nodes. Each PCB is associated with one node.

6.61.2.2 pcb_queue_t

```
typedef struct pcb_queue pcb_queue_t
```

"Master" controller of the PCB queue

6.61.3 Enumeration Type Documentation

6.61.3.1 p_state_t

```
enum p_state_t
```

Types of process states.

Enumerator

RUNNING	Running State.
READY	Ready State.
BLOCKED	Blocked State.
SUSPENDED_READY	Suspended Ready State.
SUSPENDED_BLOCKED	Suspended Blocked State.

6.61.3.2 pcb_queue_order_t

enum `pcb_queue_order_t`

Type of Queue Ordering.

Enumerator

PRIORITY	Priority Queue (Ready)
FIFO	FIFO Queue (Blocked)

6.61.4 Function Documentation

6.61.4.1 allocatePCB()

```
pcb_t * allocatePCB ( )
```

Allocate memory for a new PCB.

Allocates memory for a new PCB in the stack and performs actions to initialize PCB

Returns

Pointer to newly created PCB, NULL otherwise

6.61.4.2 blockPCB()

```
int blockPCB (
    char * name )
```

Set PCB state to be blocked.

Find the PCB name in queue and sets its state to blocked and reinserts it into the appropriate queue.

Parameters

<i>name</i>	Name of PCB to block
-------------	----------------------

6.61.4.3 createPCB()

```
int createPCB (
```

```
char * user_input )
```

Create a PCB.

Creates a new, unique PCB in memory.

Parameters

<i>name</i>	Give name of the PCB
<i>process_class</i>	The type of process class that will be used
<i>priority</i>	Priority of the PCB

Returns

Returns 1 upon success, Otherwise 0

Parse the user input

Error Handling

6.61.4.4 deletePCB()

```
int deletePCB (  
    char * name )
```

Delete PCB.

Will remove a PCB from the appropriate queue and free all associated memory. Will find the PCB in the queue, unlink it and free it.

Parameters

<i>name</i>	Name of the PCB to delete
-------------	---------------------------

Returns

Return 0 upon success, 1 upon failure

6.61.4.5 findPCB()

```
pcb_t * findPCB (  
    char * name )
```

Searches for PCB.

Given a PCB name, will search all queues for a process.

Parameters

<i>name</i>	Name of the PCB being searched
-------------	--------------------------------

Returns

Returns pointer to PCB upon success, NULL if PCB was not found

6.61.4.6 freePCB()

```
int freePCB (
    pcb_t * freed_pcb )
```

Free's memory associated with PCB.

Free's the memory associated with the PCB such as the stack and the PCB itself

Parameters

<i>freed_pcb</i>	Pointer to the PCB being freed
------------------	--------------------------------

Returns

Returns 1 upon success, 0 upon error

6.61.4.7 initPCB()

```
void initPCB ( )
```

Initialize PCB Queue.

Initialize the PCB queue's by assigning values for the two queues that exist. This method is called upon startup in the commhand

6.61.4.8 insertPCB()

```
int insertPCB (
    pcb_t * pcb )
```

Insert PCB into queue.

Inserts a PCB into the appropriate queue

Parameters

<i>pcb</i>	Pointer to the PCB being inserted
------------	-----------------------------------

Returns

0 on success, 1 on error

6.61.4.9 removePCB()

```
int removePCB (
    pcb_t * pcb )
```

Removes PCB from Queue.

Removes specified PCB from queue it is stored in.

Parameters

<i>pcb</i>	Pointer to the PCB being removed
------------	----------------------------------

Returns

Returns 1 upon success, 0 upon error

6.61.4.10 resumePCB()

```
int resumePCB (
    char * name )
```

Set PCB state to resume.

Places a PCB into a not suspended state and reinserts into the appropriate queue

Parameters

<i>name</i>	Name of PCB to resume
-------------	-----------------------

Returns

Returns 0 upon success, 1 upon error

6.61.4.11 setPriority()

```
int setPriority (
    char * args )
```

Set a new priority to a PCB.

Sets a PCB's priority and reinserts the process into the correct place in the correct queue

Parameters

<i>args</i>	Name of the PCB and new priority (PCB_NAME.PRIORITY)
-------------	--

Returns

Returns 0 upon success, 1 upon error

Parse the user input

Error Handling

6.61.4.12 setupPCB()

```
pcb_t * setupPCB (
    char * name,
    int process_class,
    int priority )
```

Creates a PCB.

Allocates and fill memory associated with the PCB being created. This is accomplished by calling [allocatePCB\(\)](#) to initialize the memory and the fills the data with the parameters.

Parameters

<i>name</i>	Name of the PCB
<i>process_class</i>	Type of process being created
<i>priority</i>	The priority of the PCB being created

Returns

Returns pointer to PCB upon success, NULL otherwise

6.61.4.13 showAll()

```
int showAll (
    char * args )
```

Show all PCBs.

Display information for each PCB in the ready and blocked queue. The information that is displayed is: Process Name, Class, State, Suspended Status, Priority.

@params args Empty params

Returns

Returns 0 upon success, 1 upon error

6.61.4.14 showBlocked()

```
int showBlocked (
    char * args )
```

Show PCBs in blocked queue.

Display information for each PCB in the blocked queue. The information that is displayed is: Process Name, Class, State, Suspended Status, Priority.

6.61.4.15 showPCB()

```
int showPCB (
    char * name )
```

Show informatino of PCB.

Display information of the PCB. The information that is displayed is: Process Name, Class, State, Suspended Status and Priority

Parameters

<i>name</i>	Name of PCB to have its information displayed
-------------	---

6.61.4.16 showReady()

```
int showReady (
    char * p )
```

Show PCBs in ready queue.

Display information for each PCB in the ready queue. The information that is displayed is: Process Name, Class, State, Suspended Status, Priority

Parameters

<i>p</i>	Empty parameters.
----------	-------------------

Returns

0 upon success, 1 upon failure

6.61.4.17 suspendPCB()

```
int suspendPCB (  
    char * name )
```

Set PCB state to suspended.

Places a PCB state into suspended and reinserts into appropriate queue

Parameters

<i>name</i>	Name of PCB to suspend
-------------	------------------------

Returns

Returns 0 upon success, 1 upon error

6.61.4.18 unblockPCB()

```
int unblockPCB (  
    char * name )
```

Set PCB state to unblocked.

Sets PCB state into unblocked and reinserts it into the appropriate queue

Parameters

<i>name</i>	Name of the PCB to unblock
-------------	----------------------------

Returns

Returns 0 upon success, 1 upon error

6.62 pcb.h

[Go to the documentation of this file.](#)

```

1 #ifndef PCB_H
2 #define PCB_H
3
4
5 #define MAX_STACK_SIZE 1024
6
7
8 #define MAX_PRIORITY 9
9 #define MIN_PRIORITY 0
10
11
12 #define MAX_NAME_SIZE 32
13
14
15 /***** Structures *****/
16 /***** Structures *****/
17 /***** Structures *****/
18
19 typedef enum {
20     PRIORITY,
21
22     FIFO
23 } pcb_queue_order_t;
24
25 typedef enum {
26     RUNNING,
27
28     READY,
29
30     BLOCKED,
31
32     SUSPENDED_READY,
33
34     SUSPENDED_BLOCKED
35 } p_state_t;
36
37 typedef struct {
38     char pcb_name[32];          // Can change size in the future
39
40     int pcb_process_class;      // I've decided that process class will be an int. SYS_PROCESS = 0,
41     APPLICATION = 1
42
43     int pcb_priority;
44
45     p_state_t pcb_process_state;
46
47     unsigned char * pcb_stack_top;
48
49     unsigned char * pcb_stack_bottom;
50 } pcb_t;
51
52 typedef struct pcb_node_t {
53     struct pcb_node_t *pcbn_next_pcb;
54
55     struct pcb_node_t *pcbn_prev_pcb;
56
57     pcb_t *pcb;
58 } pcb_node_t;
59
60 typedef struct pcb_queue {
61     int pcbq_count;
62
63     pcb_node_t *pcbq_head;
64
65     pcb_node_t *pcbq_tail;
66
67     pcb_queue_order_t queue_order;
68 } pcb_queue_t;
69
70 /***** Function Headers *****/
71 /***** Function Headers *****/
72 /***** Function Headers *****/
73
74 void initPCB();
75
76 pcb_t * allocatePCB();

```

```
119
129 int freePCB(pcb_t * freed_pcb);
130
145 pcb_t * setupPCB(char * name, int process_class, int priority);
146
157 pcb_t * findPCB(char * name);
158
168 int insertPCB(pcb_t * pcb);
169
180 int removePCB(pcb_t * pcb);
181
182
194 int createPCB(char * user_input);
195
208 int deletePCB(char * name);
209
219 int blockPCB(char * name);
220
231 int unblockPCB(char * name);
232
243 int suspendPCB(char * name);
244
255 int resumePCB(char * name);
256
267 int setPriority(char * args);
268
278 int showPCB(char * name);
279
291 int showReady(char * p);
292
301 int showBlocked(char * args);
302
314 int showAll(char * args);
315
316 #endif
```

6.63 /home/maximillian/Desktop/MAMA/term/syntax.c File Reference

```
#include "syntax.h"
#include "utils.h"
```

Functions

- int [changes_state](#) (char, enum [SyntaxState](#), enum [SyntaxState](#) *)
- enum [SyntaxState](#) [get_state](#) (char c, enum [SyntaxState](#) cur_state)

6.63.1 Function Documentation

6.63.1.1 changes_state()

```
int changes_state (
    char c,
    enum SyntaxState cur_state,
    enum SyntaxState * next_state )
```

6.63.1.2 `get_state()`

```
enum SyntaxState get_state (
    char c,
    enum SyntaxState cur_state )
```

6.64 `/home/maximillian/Desktop/MAMA/term/syntax.h` File Reference

Enumerations

- enum `SyntaxState` {
`CMD_NAME_OR_LEADING_WHITESPACE` , `CMD_NAME` , `PARAM_NAME` , `PARAM_VALUE` ,
`DOUBLE_QUOTE_STRING` , `DOUBLE_QUOTE_STRING_END_QUOTE` , `SINGLE_QUOTE_STRING` ,
`SINGLE_QUOTE_STRING_END_QUOTE` ,
`END_OF_INPUT` , `DEFAULT` }

Functions

- enum `SyntaxState` `get_state` (char, enum `SyntaxState`)
- int `changes_state` (char, enum `SyntaxState`, enum `SyntaxState` *)

6.64.1 Enumeration Type Documentation

6.64.1.1 `SyntaxState`

```
enum SyntaxState
```

Enumerator

<code>CMD_NAME_OR_LEADING_WHITESPACE</code>	
<code>CMD_NAME</code>	
<code>PARAM_NAME</code>	
<code>PARAM_VALUE</code>	
<code>DOUBLE_QUOTE_STRING</code>	
<code>DOUBLE_QUOTE_STRING_END_QUOTE</code>	
<code>SINGLE_QUOTE_STRING</code>	
<code>SINGLE_QUOTE_STRING_END_QUOTE</code>	
<code>END_OF_INPUT</code>	
<code>DEFAULT</code>	

6.64.2 Function Documentation

6.64.2.1 changes_state()

```
int changes_state (
    char c,
    enum SyntaxState,
    enum SyntaxState * next_state )
```

6.64.2.2 get_state()

```
enum SyntaxState get_state (
    char c,
    enum SyntaxState )
```

6.65 syntax.h

[Go to the documentation of this file.](#)

```
1 #ifndef SYNTAX_H
2 #define SYNTAX_H
3
4 enum SyntaxState {
5     CMD_NAME_OR_LEADING_WHITESPACE,
6     CMD_NAME,
7     PARAM_NAME,
8     PARAM_VALUE,
9     DOUBLE_QUOTE_STRING,
10    DOUBLE_QUOTE_STRING_END_QUOTE,
11    SINGLE_QUOTE_STRING,
12    SINGLE_QUOTE_STRING_END_QUOTE,
13    END_OF_INPUT,
14    DEFAULT
15 };
16
17 enum SyntaxState get_state(char, enum SyntaxState);
18 int changes_state(char, enum SyntaxState, enum SyntaxState *);
19
20 #endif
```

6.66 /home/maximillian/Desktop/MAMA/term/utils.c File Reference

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

Functions

- int [is_name_char](#) (char c)
Returns whether or not the specified character is a valid character in an identifier, such as a command or argument name.
- void [skip_ws](#) (char **c)
Moves the specified pointer to a character buffer forward until it points to the next non-whitespace character.

6.66.1 Function Documentation

6.66.1.1 `is_name_char()`

```
int is_name_char (
    char c )
```

Returns whether or not the specified character is a valid character in an identifier, such as a command or argument name.

Parameters

<code>c</code>	The character to test.
----------------	------------------------

Returns

True if the specified character `c` is valid in an identifier, false otherwise.

6.66.1.2 `skip_ws()`

```
void skip_ws (
    char ** c )
```

Moves the specified pointer to a character buffer forward until it points to the next non-whitespace character.

Parameters

<code>c</code>	A pointer to a pointer to an entry in a character buffer. Will be modified to point to the next non-whitespace character in the buffer.
----------------	---

6.67 `/home/maximillian/Desktop/MAMA/term/utils.h` File Reference

Functions

- int [is_name_char](#) (char)
Returns whether or not the specified character is a valid character in an identifier, such as a command or argument name.
- void [skip_ws](#) (char **)
Moves the specified pointer to a character buffer forward until it points to the next non-whitespace character.

6.67.1 Function Documentation

6.67.1.1 is_name_char()

```
int is_name_char (
    char c )
```

Returns whether or not the specified character is a valid character in an identifier, such as a command or argument name.

Parameters

<code>c</code>	The character to test.
----------------	------------------------

Returns

True if the specified character `c` is valid in an identifier, false otherwise.

6.67.1.2 skip_ws()

```
void skip_ws (
    char ** c )
```

Moves the specified pointer to a character buffer forward until it points to the next non-whitespace character.

Parameters

<code>c</code>	A pointer to a pointer to an entry in a character buffer. Will be modified to point to the next non-whitespace character in the buffer.
----------------	---

6.68 utils.h

[Go to the documentation of this file.](#)

```
1 #ifndef UTILS_H
2 #define UTILS_H
3
4 int is_name_char(char);
5 void skip_ws(char **);
6
7 #endif
```

6.69 /home/maximillian/Desktop/MAMA/term/visuals/colorize.c File Reference

```
#include <lib/out.h>
```

Macros

- `#define START_SEQ "\e["`

Enumerations

- `enum Color {`
`BLACK, RED, GREEN, YELLOW,`
`BLUE, MAGENTA, CYAN, WHITE,`
`BLACK, RED, GREEN, YELLOW,`
`BLUE, MAGENTA, CYAN, WHITE }`

Functions

- void `print_color_code` (enum `Color` color)
Description: Prints part of the escape sequence needed to switch the foreground or background color to the specified color.
- void `display_fg_color` (enum `Color` color)
Switches the text color in the terminal so that subsequent text written to the screen will be the specified color.
- void `display_bg_color` (enum `Color` color)
Switches the background color in the terminal so that subsequent text written to the screen will appear on a backdrop of the specified color.
- void `display_reset` ()
Resets any formatting so that subsequent text written to the screen will use the default appearance.
- void `display_italicize` ()
Description: Causes subsequent text written to the screen to be displayed in italics.

6.69.1 Macro Definition Documentation

6.69.1.1 START_SEQ

```
#define START_SEQ "\e["
```

6.69.2 Enumeration Type Documentation

6.69.2.1 Color

```
enum Color
```

Enumerator

BLACK	
RED	
GREEN	
YELLOW	
BLUE	
MAGENTA	
CYAN	
WHITE	
BLACK	
RED	
GREEN	
YELLOW	
BLUE	
MAGENTA	
CYAN	
WHITE	

6.69.3 Function Documentation

6.69.3.1 display_bg_color()

```
void display_bg_color (
    enum Color color )
```

Switches the background color in the terminal so that subsequent text written to the screen will appear on a backdrop of the specified color.

Parameters

<i>color</i>	The color to switch to.
--------------	-------------------------

6.69.3.2 display_fg_color()

```
void display_fg_color (
    enum Color color )
```

Switches the text color in the terminal so that subsequent text written to the screen will be the specified color.

Parameters

<i>color</i>	The color to switch to.
--------------	-------------------------

6.69.3.3 display_italicize()

```
void display_italicize ( )
```

Description: Causes subsequent text written to the screen to be displayed in italics.

6.69.3.4 display_reset()

```
void display_reset ( )
```

Resets any formatting so that subsequent text written to the screen will use the default appearance.

6.69.3.5 print_color_code()

```
void print_color_code (
    enum Color color )
```

Description: Prints part of the escape sequence needed to switch the foreground or background color to the specified color.

Used internally by display_fg_color and display_bg_color.

Parameters

<i>color</i>	The color being switched to.
--------------	------------------------------

6.70 /home/maximillian/Desktop/MAMA/term/visuals/colorize.h File Reference

Enumerations

- enum Color {
BLACK, RED, GREEN, YELLOW,
BLUE, MAGENTA, CYAN, WHITE,
BLACK, RED, GREEN, YELLOW,
BLUE, MAGENTA, CYAN, WHITE }

Functions

- void display_fg_color (enum Color)
Switches the text color in the terminal so that subsequent text written to the screen will be the specified color.
- void display_bg_color (enum Color)
Switches the background color in the terminal so that subsequent text written to the screen will appear on a backdrop of the specified color.
- void display_italicize ()
Description: Causes subsequent text written to the screen to be displayed in italics.
- void display_reset ()
Resets any formatting so that subsequent text written to the screen will use the default appearance.

6.70.1 Enumeration Type Documentation

6.70.1.1 Color

```
enum Color
```

Enumerator

BLACK	
RED	
GREEN	
YELLOW	
BLUE	
MAGENTA	
CYAN	
WHITE	
BLACK	
RED	
GREEN	
YELLOW	
BLUE	
MAGENTA	
CYAN	
WHITE	

6.70.2 Function Documentation

6.70.2.1 display_bg_color()

```
void display_bg_color (
    enum Color color )
```

Switches the background color in the terminal so that subsequent text written to the screen will appear on a backdrop of the specified color.

Parameters

<i>color</i>	The color to switch to.
--------------	-------------------------

6.70.2.2 display_fg_color()

```
void display_fg_color (
    enum Color color )
```

Switches the text color in the terminal so that subsequent text written to the screen will be the specified color.

Parameters

<i>color</i>	The color to switch to.
--------------	-------------------------

6.70.2.3 `display_italicize()`

```
void display_italicize ( )
```

Description: Causes subsequent text written to the screen to be displayed in italics.

6.70.2.4 `display_reset()`

```
void display_reset ( )
```

Resets any formatting so that subsequent text written to the screen will use the default appearance.

6.71 `colorize.h`

[Go to the documentation of this file.](#)

```
1 #ifndef COLORIZE_H
2 #define COLORIZE_H
3
4 enum Color {
5     BLACK,
6     RED,
7     GREEN,
8     YELLOW,
9     BLUE,
10    MAGENTA,
11    CYAN,
12    WHITE
13 };
14
15 void display_fg_color(enum Color);
16 void display_bg_color(enum Color);
17 void display_italicize();
18 void display_reset();
19
20 #endif
```

6.72 `/home/maximillian/Desktop/MAMA/term/visuals/cursor.c` File Reference

```
#include <lib/out.h>
```

Functions

- void `cursor_left` (int steps)
Moves the visual cursor to the left a specified number of steps.
- void `cursor_right` (int steps)
Moves the visual cursor to the right a specified number of steps.
- void `cursor_down` (int steps)
Moves the visual cursor down a specified number of steps.
- void `cursor_up` (int steps)
Moves the visual cursor up a specified number of steps.
- void `cursor_return` ()
Moves the visual cursor to the beginning of the line.

6.72.1 Function Documentation

6.72.1.1 `cursor_down()`

```
void cursor_down (
    int steps )
```

Moves the visual cursor down a specified number of steps.

Parameters

<i>steps</i>	The number of steps to move the cursor down.
--------------	--

6.72.1.2 `cursor_left()`

```
void cursor_left (
    int steps )
```

Moves the visual cursor to the left a specified number of steps.

Parameters

<i>steps</i>	The number of steps to move the cursor to the left.
--------------	---

6.72.1.3 `cursor_return()`

```
void cursor_return ( )
```

Moves the visual cursor to the beginning of the line.

6.72.1.4 `cursor_right()`

```
void cursor_right (
    int steps )
```

Moves the visual cursor to the right a specified number of steps.

Parameters

<i>steps</i>	The number of steps to move the cursor to the right.
--------------	--

6.72.1.5 cursor_up()

```
void cursor_up (  
    int steps )
```

Moves the visual cursor up a specified number of steps.

Parameters

<i>steps</i>	The number of steps to move the cursor up.
--------------	--

6.73 /home/maximillian/Desktop/MAMA/term/visuals/cursor.h File Reference

Functions

- void [cursor_left](#) (int)
Moves the visual cursor to the left a specified number of steps.
- void [cursor_right](#) (int)
Moves the visual cursor to the right a specified number of steps.
- void [cursor_up](#) (int)
Moves the visual cursor up a specified number of steps.
- void [cursor_down](#) (int)
Moves the visual cursor down a specified number of steps.
- void [cursor_return](#) ()
Moves the visual cursor to the beginning of the line.

6.73.1 Function Documentation

6.73.1.1 cursor_down()

```
void cursor_down (  
    int steps )
```

Moves the visual cursor down a specified number of steps.

Parameters

<i>steps</i>	The number of steps to move the cursor down.
--------------	--

6.73.1.2 cursor_left()

```
void cursor_left (
    int steps )
```

Moves the visual cursor to the left a specified number of steps.

Parameters

<i>steps</i>	The number of steps to move the cursor to the left.
--------------	---

6.73.1.3 cursor_return()

```
void cursor_return ( )
```

Moves the visual cursor to the beginning of the line.

6.73.1.4 cursor_right()

```
void cursor_right (
    int steps )
```

Moves the visual cursor to the right a specified number of steps.

Parameters

<i>steps</i>	The number of steps to move the cursor to the right.
--------------	--

6.73.1.5 cursor_up()

```
void cursor_up (
    int steps )
```

Moves the visual cursor up a specified number of steps.

Parameters

<code>steps</code>	The number of steps to move the cursor up.
--------------------	--

6.74 cursor.h

[Go to the documentation of this file.](#)

```
1 #ifndef CURSOR_H
2 #define CURSOR_H
3
4 void cursor_left(int);
5 void cursor_right(int);
6 void cursor_up(int);
7 void cursor_down(int);
8 void cursor_return();
9
10 #endif
```

6.75 /home/maximillian/Desktop/MAMA/term/visuals/hints.c File Reference

```
#include <lib/out.h>
#include "cursor.h"
```

Functions

- void `hint_under_prompt` (char *str, int len, int ret_index)
Writes a line of text under the user's prompt in the terminal.

6.75.1 Function Documentation

6.75.1.1 hint_under_prompt()

```
void hint_under_prompt (
    char * str,
    int len,
    int ret_index )
```

Writes a line of text under the user's prompt in the terminal.

Recommended for providing hints or warnings to the user as they type.

Parameters

<code>str</code>	The text to write under the user's prompt.
<code>len</code>	The length of the text to write under the user's prompt.
<code>ret_index</code>	The position to return the user's cursor to after writing the text.

6.76 /home/maximillian/Desktop/MAMA/term/visuals/hints.h File Reference

Functions

- void [hint_under_prompt](#) (char *, int, int)
Writes a line of text under the user's prompt in the terminal.

6.76.1 Function Documentation

6.76.1.1 [hint_under_prompt\(\)](#)

```
void hint_under_prompt (
    char * str,
    int len,
    int ret_index )
```

Writes a line of text under the user's prompt in the terminal.

Recommended for providing hints or warnings to the user as they type.

Parameters

<i>str</i>	The text to write under the user's prompt.
<i>len</i>	The length of the text to write under the user's prompt.
<i>ret_index</i>	The position to return the user's cursor to after writing the text.

6.77 hints.h

[Go to the documentation of this file.](#)

```
1 #ifndef HINTS_H
2 #define HINTS_H
3
4 void hint_under_prompt(char *, int, int);
5
6 #endif
```

6.78 /home/maximillian/Desktop/MAMA/term/visuals/syntax_highlight.c File Reference

```
#include "../syntax.h"
#include "../syntax.c"
#include "syntax_highlight.h"
#include "../commhand.h"
```

```
#include "colorize.h"
#include "hints.c"
#include "../utils.c"
#include <include/string.h>
```

Functions

- void `switch_to` (enum `SyntaxState`, int, int)
Whether or not syntax highlighting is enabled as the user types.
- void `color_for` (enum `SyntaxState` state)
Prints the ANSI color code for the specified syntax state.
- void `get_state_at` (int index, int *index_of_state_in_record)
Retrieves the index in the states and switch_indexes data structures corresponding to the specified cursor index.
- void `syntax_init` ()
Initializes internal data structures needed for syntax highlighting.
- void `syntax_enable_highlighting` ()
Enables syntax highlighting as the user types.
- void `syntax_disable_highlighting` ()
Disables syntax highlighting as the user types.
- void `syntax_handle_char` (char c, int index)
Adjusts the terminal color assuming the specified character will immediately be written to the screen at the specified index.

Variables

- enum `SyntaxState` states [`MAX_SYNTAX_SWITCHES`]
- int `switch_indexes` [`MAX_SYNTAX_SWITCHES`]
Array of all the states the cursor has been in as the user has typed. Entries correspond to entries in switch_indexes.
- int `newest_switch`
Array of indexes the cursor was at when the corresponding syntax state in states was switched to.
- int `enabled` = 0
The largest and most recent valid index in states and switch_indexes.

6.78.1 Function Documentation

6.78.1.1 `color_for()`

```
void color_for (
    enum SyntaxState state )
```

Prints the ANSI color code for the specified syntax state.

Used internally by `syntax_handle_char`.

Parameters

<i>state</i>	The syntax state for which to print the correct color code to the terminal for.
--------------	---

6.78.1.2 get_state_at()

```
void get_state_at (
    int index,
    int * index_of_state_in_record )
```

Retrieves the index in the states and switch_indexes data structures corresponding to the specified cursor index.

Used internally by syntax_handle_char.

Parameters

<i>index</i>	The index of the cursor.
<i>index_of_state_in_record</i>	A pointer to the index in the states and switch_indexes data structures corresponding to the specified cursor index. Will be updated to point to the correct index in the data structures.

6.78.1.3 switch_to()

```
void switch_to (
    enum SyntaxState state,
    int index,
    int record_index )
```

Whether or not syntax highlighting is enabled as the user types.

Switches to the specified syntax state.

Used internally by syntax_handle_char.

Parameters

<i>state</i>	The syntax state being switched to.
<i>index</i>	The index in the user's input at which this switch occurs.
<i>record_index</i>	The index in the internal data structures states and switch_indexes at which to write this switch to.

6.78.1.4 syntax_disable_highlighting()

```
void syntax_disable_highlighting ( )
```


Disables syntax highlighting as the user types.

6.78.1.5 `syntax_enable_highlighting()`

```
void syntax_enable_highlighting ( )
```

Enables syntax highlighting as the user types.

6.78.1.6 `syntax_handle_char()`

```
void syntax_handle_char (
    char c,
    int index )
```

Adjusts the terminal color assuming the specified character will immediately be written to the screen at the specified index.

Parameters

<i>c</i>	The next character that will be output to the screen.
<i>index</i>	The index of the cursor.

6.78.1.7 `syntax_init()`

```
void syntax_init ( )
```

Initializes internal data structures needed for syntax highlighting.

6.78.2 Variable Documentation

6.78.2.1 `enabled`

```
int enabled = 0
```

The largest and most recent valid index in `states` and `switch_indexes`.

6.78.2.2 newest_switch

```
int newest_switch
```

Array of indexes the cursor was at when the corresponding syntax state in states was switched to.

6.78.2.3 states

```
enum SyntaxState states[MAX_SYNTAX_SWITCHES]
```

6.78.2.4 switch_indexes

```
int switch_indexes[MAX_SYNTAX_SWITCHES]
```

Array of all the states the cursor has been in as the user has typed. Entries correspond to entries in switch_indexes.

6.79 /home/maximillian/Desktop/MAMA/term/visuals/syntax_highlight.h File Reference

Macros

- #define MAX_SYNTAX_SWITCHES 40
- #define SYNTAX_COLOR_CMD_NAME CYAN
- #define SYNTAX_COLOR_PARAM_NAME MAGENTA
- #define SYNTAX_COLOR_PARAM_VALUE WHITE
- #define SYNTAX_COLOR_DOUBLE_QUOTE_STRING YELLOW
- #define SYNTAX_COLOR_SINGLE_QUOTE_STRING YELLOW
- #define SYNTAX_COLOR_DEFAULT WHITE

Functions

- void `syntax_init` ()
Initializes internal data structures needed for syntax highlighting.
- void `syntax_enable_highlighting` ()
Enables syntax highlighting as the user types.
- void `syntax_disable_highlighting` ()
Disables syntax highlighting as the user types.
- void `syntax_handle_char` (char, int)
Adjusts the terminal color assuming the specified character will immediately be written to the screen at the specified index.

6.79.1 Macro Definition Documentation

6.79.1.1 MAX_SYNTAX_SWITCHES

```
#define MAX_SYNTAX_SWITCHES 40
```

6.79.1.2 SYNTAX_COLOR_CMD_NAME

```
#define SYNTAX_COLOR_CMD_NAME CYAN
```

6.79.1.3 SYNTAX_COLOR_DEFAULT

```
#define SYNTAX_COLOR_DEFAULT WHITE
```

6.79.1.4 SYNTAX_COLOR_DOUBLE_QUOTE_STRING

```
#define SYNTAX_COLOR_DOUBLE_QUOTE_STRING YELLOW
```

6.79.1.5 SYNTAX_COLOR_PARAM_NAME

```
#define SYNTAX_COLOR_PARAM_NAME MAGENTA
```

6.79.1.6 SYNTAX_COLOR_PARAM_VALUE

```
#define SYNTAX_COLOR_PARAM_VALUE WHITE
```

6.79.1.7 SYNTAX_COLOR_SINGLE_QUOTE_STRING

```
#define SYNTAX_COLOR_SINGLE_QUOTE_STRING YELLOW
```

6.79.2 Function Documentation

6.79.2.1 `syntax_disable_highlighting()`

```
void syntax_disable_highlighting ( )
```

Disables syntax highlighting as the user types.

6.79.2.2 `syntax_enable_highlighting()`

```
void syntax_enable_highlighting ( )
```

Enables syntax highlighting as the user types.

6.79.2.3 `syntax_handle_char()`

```
void syntax_handle_char (
    char c,
    int index )
```

Adjusts the terminal color assuming the specified character will immediately be written to the screen at the specified index.

Parameters

<i>c</i>	The next character that will be output to the screen.
<i>index</i>	The index of the cursor.

6.79.2.4 `syntax_init()`

```
void syntax_init ( )
```

Initializes internal data structures needed for syntax highlighting.

6.80 `syntax_highlight.h`

[Go to the documentation of this file.](#)

```
1 #ifndef SYNTAX_HIGHLIGHT_H
```

```
2 #define SYNTAX_HIGHLIGHT_H
3
4 #define MAX_SYNTAX_SWITCHES 40
5
6 #define SYNTAX_COLOR_CMD_NAME CYAN
7 #define SYNTAX_COLOR_PARAM_NAME MAGENTA
8 #define SYNTAX_COLOR_PARAM_VALUE WHITE
9 #define SYNTAX_COLOR_DOUBLE_QUOTE_STRING YELLOW
10 #define SYNTAX_COLOR_SINGLE_QUOTE_STRING YELLOW
11 #define SYNTAX_COLOR_DEFAULT WHITE
12
13 void syntax_init();
14 void syntax_enable_highlighting();
15 void syntax_disable_highlighting();
16 void syntax_handle_char(char, int);
17
18 #endif
```

6.81 /home/maximillian/Desktop/MAMA/WhoDidWhat.md File Reference

Index

/home/maximillian/Desktop/MAMA/README.md, [86](#)
/home/maximillian/Desktop/MAMA/WhoDidWhat.md, [167](#)
/home/maximillian/Desktop/MAMA/help.c, [92](#)
/home/maximillian/Desktop/MAMA/include/core/asm.h, [27](#)
/home/maximillian/Desktop/MAMA/include/core/comhand.h, [27](#), [28](#)
/home/maximillian/Desktop/MAMA/include/core/interrupts.h, [28](#)
/home/maximillian/Desktop/MAMA/include/core/io.h, [29](#)
/home/maximillian/Desktop/MAMA/include/core/serial.h, [30](#), [32](#)
/home/maximillian/Desktop/MAMA/include/core/tables.h, [33](#), [36](#)
/home/maximillian/Desktop/MAMA/include/mem/heap.h, [36](#), [39](#)
/home/maximillian/Desktop/MAMA/include/mem/paging.h, [40](#), [42](#)
/home/maximillian/Desktop/MAMA/include/string.h, [43](#), [45](#)
/home/maximillian/Desktop/MAMA/include/system.h, [46](#), [49](#)
/home/maximillian/Desktop/MAMA/kernel/core/interrupts.c, [49](#)
/home/maximillian/Desktop/MAMA/kernel/core/kmain.c, [57](#)
/home/maximillian/Desktop/MAMA/kernel/core/serial.c, [57](#)
/home/maximillian/Desktop/MAMA/kernel/core/system.c, [61](#)
/home/maximillian/Desktop/MAMA/kernel/core/tables.c, [61](#)
/home/maximillian/Desktop/MAMA/kernel/mem/heap.c, [63](#)
/home/maximillian/Desktop/MAMA/kernel/mem/paging.c, [66](#)
/home/maximillian/Desktop/MAMA/lib/out.c, [69](#)
/home/maximillian/Desktop/MAMA/lib/out.h, [70](#), [76](#)
/home/maximillian/Desktop/MAMA/lib/string.c, [76](#)
/home/maximillian/Desktop/MAMA/modules/mpx_supt.c, [78](#)
/home/maximillian/Desktop/MAMA/modules/mpx_supt.h, [81](#), [85](#)
/home/maximillian/Desktop/MAMA/term/args.c, [86](#)
/home/maximillian/Desktop/MAMA/term/args.h, [89](#), [90](#)
/home/maximillian/Desktop/MAMA/term/ascii/mama.c, [90](#)
/home/maximillian/Desktop/MAMA/term/ascii/mama.h, [91](#)
/home/maximillian/Desktop/MAMA/term/cmds/argtest.c, [92](#)
/home/maximillian/Desktop/MAMA/term/cmds/echo.c, [92](#)
/home/maximillian/Desktop/MAMA/term/cmds/help.c, [95](#)
/home/maximillian/Desktop/MAMA/term/cmds/pcb.c, [125](#)
/home/maximillian/Desktop/MAMA/term/cmds/shutdown.c, [99](#)
/home/maximillian/Desktop/MAMA/term/cmds/version.c, [100](#)
/home/maximillian/Desktop/MAMA/term/commands.h, [100](#), [101](#)
/home/maximillian/Desktop/MAMA/term/commhand.c, [101](#)
/home/maximillian/Desktop/MAMA/term/commhand.h, [103](#), [105](#)
/home/maximillian/Desktop/MAMA/term/dnt/dnt.c, [106](#)
/home/maximillian/Desktop/MAMA/term/dnt/dnt.h, [111](#), [120](#)
/home/maximillian/Desktop/MAMA/term/history.c, [121](#)
/home/maximillian/Desktop/MAMA/term/history.h, [124](#), [125](#)
/home/maximillian/Desktop/MAMA/term/pcb/pcb.c, [125](#)
/home/maximillian/Desktop/MAMA/term/pcb/pcb.h, [133](#), [144](#)
/home/maximillian/Desktop/MAMA/term/syntax.c, [145](#)
/home/maximillian/Desktop/MAMA/term/syntax.h, [146](#), [147](#)
/home/maximillian/Desktop/MAMA/term/utils.c, [147](#)
/home/maximillian/Desktop/MAMA/term/utils.h, [148](#), [150](#)
/home/maximillian/Desktop/MAMA/term/visuals/colorize.c, [150](#)
/home/maximillian/Desktop/MAMA/term/visuals/colorize.h, [153](#), [155](#)
/home/maximillian/Desktop/MAMA/term/visuals/cursor.c, [155](#)
/home/maximillian/Desktop/MAMA/term/visuals/cursor.h, [157](#), [159](#)
/home/maximillian/Desktop/MAMA/term/visuals/hints.c, [159](#)
/home/maximillian/Desktop/MAMA/term/visuals/hints.h, [160](#)
/home/maximillian/Desktop/MAMA/term/visuals/syntax_highlight.c, [160](#)

/home/maximillian/Desktop/MAMA/term/visuals/syntax_highlight
 164, 166
 __attribute__
 tables.h, 33
 __end
 heap.c, 65
 _end
 heap.c, 65
 _kmalloc
 heap.c, 64
 heap.h, 37
 access
 gdt_entry_struct, 12
 tables.h, 34
 accessed
 page_entry, 19
 alloc
 heap.c, 64
 heap.h, 37
 allocatePCB
 pcb.c, 127
 pcb.h, 137
 args.c
 cur_state, 89
 flag, 87
 get_token, 87
 last_state, 89
 MAX_PARSE_STACK_SIZE, 87
 named_arg, 88
 next_unnamed_arg, 88
 parse_args, 88
 parse_stack, 89
 stack_empty, 88
 stack_peek, 88
 stack_pop, 88
 stack_push, 89
 stack_size, 89
 args.h
 parse_args, 90
 parsed_args, 90
 argtest.c
 cmd_argtest, 92
 asm
 system.h, 46
 atoi
 string.c, 77
 string.h, 43
 base
 gdt_descriptor_struct, 12
 heap, 14
 idt_struct, 16
 tables.h, 34
 base_high
 gdt_entry_struct, 12
 idt_entry_struct, 15
 tables.h, 34
 base_low
 gdt_entry_struct, 13
 idt_entry_struct, 15
 tables.h, 35
 base_mid
 gdt_entry_struct, 13
 tables.h, 35
 BCDtol
 dnt.c, 106
 dnt.h, 115
 BLACK
 colorize.c, 151
 colorize.h, 154
 block
 index_entry, 17
 BLOCKED
 pcb.h, 136
 blockHelp
 help.c, 96
 out.h, 71
 blockPCB
 pcb.c, 127
 pcb.h, 137
 BLUE
 colorize.c, 151
 colorize.h, 154
 bounds
 interrupts.c, 51
 breakpoint
 interrupts.c, 51
 buffer_ptr
 param, 21
 cdir
 paging.c, 68
 changes_state
 syntax.c, 145
 syntax.h, 146
 circular_next_index
 history.c, 121
 circular_prev_index
 history.c, 122
 clear_bit
 paging.c, 66
 paging.h, 40
 cli
 system.h, 46
 cmd_argtest
 argtest.c, 92
 cmd_echo
 echo.c, 92
 cmd_func_t
 commhand.c, 102
 cmd_handler
 cmd_mapping, 9
 cmd_help
 help.c, 93, 96
 out.h, 71
 cmd_mapping, 9
 cmd_handler, 9

- cmd_name, 9
- commhand.c, 102
- cmd_mappings
 - commhand.c, 103
- CMD_NAME
 - syntax.h, 146
- cmd_name
 - cmd_mapping, 9
- CMD_NAME_OR_LEADING_WHITESPACE
 - syntax.h, 146
- cmd_shutdown
 - shutdown.c, 99
- cmd_version
 - version.c, 100
- Color
 - colorize.c, 151
 - colorize.h, 153
- color_for
 - syntax_highlight.c, 161
- colorize.c
 - BLACK, 151
 - BLUE, 151
 - Color, 151
 - CYAN, 151
 - display_bg_color, 152
 - display_fg_color, 152
 - display_italicize, 152
 - display_reset, 152
 - GREEN, 151
 - MAGENTA, 151
 - print_color_code, 152
 - RED, 151
 - START_SEQ, 151
 - WHITE, 151
 - YELLOW, 151
- colorize.h
 - BLACK, 154
 - BLUE, 154
 - Color, 153
 - CYAN, 154
 - display_bg_color, 154
 - display_fg_color, 154
 - display_italicize, 155
 - display_reset, 155
 - GREEN, 154
 - MAGENTA, 154
 - RED, 154
 - WHITE, 154
 - YELLOW, 154
- COM1
 - serial.h, 30
- COM2
 - serial.h, 30
- COM3
 - serial.h, 30
- COM4
 - serial.h, 30
- COM_PORT
 - mpx_supt.h, 82
- comhand
 - comhand.h, 27
- comhand.h
 - comhand, 27
- commhand
 - commhand.c, 102
 - commhand.h, 105
- commhand.c
 - cmd_func_t, 102
 - cmd_mapping, 102
 - cmd_mappings, 103
 - commhand, 102
 - extract_cmd_name, 102
 - fetch_cmd_handler, 103
 - is_name_char, 103
- commhand.h
 - commhand, 105
 - MAX_CMD_ARG_NAME_LEN, 104
 - MAX_CMD_ARG_VALUE_LEN, 104
 - MAX_CMD_FLAG_COUNT, 104
 - MAX_CMD_HIST_LEN, 104
 - MAX_CMD_NAME_LEN, 104
 - MAX_CMD_NAMED_ARG_COUNT, 104
 - MAX_CMD_STRING_LEN, 105
 - MAX_CMD_UNNAMED_ARG_COUNT, 105
- consume_special
 - serial.c, 59
- coprocessor
 - interrupts.c, 52
- coprocessor_segment
 - interrupts.c, 52
- count_ptr
 - param, 21
- createPCB
 - pcb.c, 127
 - pcb.h, 137
- createpcbHelp
 - help.c, 96
 - out.h, 71
- cur_state
 - args.c, 89
- curr_heap
 - heap.c, 65
- current_module
 - mpx_supt.c, 80
- cursor.c
 - cursor_down, 156
 - cursor_left, 156
 - cursor_return, 156
 - cursor_right, 156
 - cursor_up, 157
- cursor.h
 - cursor_down, 157
 - cursor_left, 158
 - cursor_return, 158
 - cursor_right, 158
 - cursor_up, 158

- cursor_down
 - cursor.c, 156
 - cursor.h, 157
- cursor_left
 - cursor.c, 156
 - cursor.h, 158
- cursor_return
 - cursor.c, 156
 - cursor.h, 158
- cursor_right
 - cursor.c, 156
 - cursor.h, 158
- cursor_up
 - cursor.c, 157
 - cursor.h, 158
- CYAN
 - colorize.c, 151
 - colorize.h, 154
- date_time, 9
 - day_m, 10
 - day_w, 10
 - day_y, 10
 - hour, 10
 - min, 10
 - mon, 10
 - sec, 11
 - year, 11
- day_m
 - date_time, 10
- day_w
 - date_time, 10
- day_y
 - date_time, 10
- DAYS_IN_LEAP_YEAR
 - dnt.h, 112
- DAYS_IN_YEAR
 - dnt.h, 112
- daysInMonth
 - dnt.c, 107
 - dnt.h, 115
- debug
 - interrupts.c, 52
- DEFAULT
 - syntax.h, 146
- DEFAULT_DEVICE
 - mpx_supt.h, 82
- DELETE
 - serial.c, 58
- deletePCB
 - pcb.c, 128
 - pcb.h, 138
- deletepcbHelp
 - help.c, 96
 - out.h, 72
- device_id
 - param, 21
- device_not_available
 - interrupts.c, 52
- dirty
 - page_entry, 19
- display_bg_color
 - colorize.c, 152
 - colorize.h, 154
- display_fg_color
 - colorize.c, 152
 - colorize.h, 154
- display_italicize
 - colorize.c, 152
 - colorize.h, 155
- display_reset
 - colorize.c, 152
 - colorize.h, 155
- divide_error
 - interrupts.c, 52
- dnt.c
 - BCDtol, 106
 - daysInMonth, 107
 - getdate, 107
 - gettime, 107
 - intToDayOfWeek, 108
 - intToMonth, 108
 - ltoBCD, 109
 - setdate, 109
 - setDateInMemory, 109
 - settime, 110
 - setTimeInMemory, 110
- dnt.h
 - BCDtol, 115
 - DAYS_IN_LEAP_YEAR, 112
 - DAYS_IN_YEAR, 112
 - daysInMonth, 115
 - EPOCH_FIRST_DAY_OF_WEEK_OF_YEAR, 112
 - EPOCH_FIRST_DAY_OF_YEAR, 113
 - EPOCH_FIRST_MONTH_OF_YEAR, 113
 - EPOCH_YEAR, 113
 - getdate, 116
 - gettime, 116
 - intToDayOfWeek, 117
 - intToMonth, 117
 - ltoBCD, 118
 - MAX_DAY, 113
 - MAX_HOURS, 113
 - MAX_MINUTES, 113
 - MAX_MONTH, 114
 - MAX_SECONDS, 114
 - MAX_YEAR, 114
 - MIN, 114
 - MIN_DAY, 114
 - MIN_MONTH, 114
 - MIN_YEAR, 115
 - setdate, 118
 - setDateInMemory, 119
 - settime, 119
 - setTimeInMemory, 120
- do_bounds
 - interrupts.c, 52

- do_breakpoint
 - interrupts.c, [52](#)
- do_coprocessor
 - interrupts.c, [52](#)
- do_coprocessor_segment
 - interrupts.c, [53](#)
- do_debug
 - interrupts.c, [53](#)
- do_device_not_available
 - interrupts.c, [53](#)
- do_divide_error
 - interrupts.c, [53](#)
- do_double_fault
 - interrupts.c, [53](#)
- do_general_protection
 - interrupts.c, [53](#)
- do_invalid_op
 - interrupts.c, [53](#)
- do_invalid_tss
 - interrupts.c, [53](#)
- do_isr
 - interrupts.c, [54](#)
- do_nmi
 - interrupts.c, [54](#)
- do_overflow
 - interrupts.c, [54](#)
- do_page_fault
 - interrupts.c, [54](#)
- do_reserved
 - interrupts.c, [54](#)
- do_segment_not_present
 - interrupts.c, [54](#)
- do_stack_segment
 - interrupts.c, [54](#)
- double_fault
 - interrupts.c, [54](#)
- DOUBLE_QUOTE_STRING
 - syntax.h, [146](#)
- DOUBLE_QUOTE_STRING_END_QUOTE
 - syntax.h, [146](#)
- DOWN_ARROW
 - serial.c, [58](#)
- echo.c
 - cmd_echo, [92](#)
- empty
 - index_entry, [17](#)
- enabled
 - syntax_highlight.c, [163](#)
- end
 - heap.c, [65](#)
- END_OF_INPUT
 - syntax.h, [146](#)
- EPOCH_FIRST_DAY_OF_WEEK_OF_YEAR
 - dnt.h, [112](#)
- EPOCH_FIRST_DAY_OF_YEAR
 - dnt.h, [113](#)
- EPOCH_FIRST_MONTH_OF_YEAR
 - dnt.h, [113](#)
- EPOCH_YEAR
 - dnt.h, [113](#)
- EXIT
 - mpx_supt.h, [82](#)
- extract_cmd_name
 - commhand.c, [102](#)
- f_queue
 - pcb.c, [133](#)
- FALSE
 - mpx_supt.h, [82](#)
- fetch_cmd_handler
 - commhand.c, [103](#)
- FIFO
 - pcb.h, [137](#)
- fifo_queue
 - pcb.c, [133](#)
- find_free
 - paging.c, [66](#)
- findPCB
 - pcb.c, [128](#)
 - pcb.h, [138](#)
- first_free
 - paging.h, [40](#)
- flag
 - args.c, [87](#)
- flag_count
 - parsed_args, [22](#)
- flags
 - gdt_entry_struct, [13](#)
 - idt_entry_struct, [15](#)
 - parsed_args, [22](#)
 - tables.h, [35](#)
- footer, [11](#)
 - head, [11](#)
- frameaddr
 - page_entry, [19](#)
- frames
 - paging.c, [68](#)
- freePCB
 - pcb.c, [128](#)
 - pcb.h, [139](#)
- GDT_CS_ID
 - system.h, [46](#)
- gdt_descriptor_struct, [11](#)
 - base, [12](#)
 - limit, [12](#)
- GDT_DS_ID
 - system.h, [47](#)
- gdt_entries
 - tables.c, [63](#)
- gdt_entry_struct, [12](#)
 - access, [12](#)
 - base_high, [12](#)
 - base_low, [13](#)
 - base_mid, [13](#)
 - flags, [13](#)
 - limit_low, [13](#)

- gdt_init_entry
 - tables.c, 62
 - tables.h, 33
- gdt_ptr
 - tables.c, 63
- general_protection
 - interrupts.c, 55
- get_bit
 - paging.c, 67
 - paging.h, 41
- get_page
 - paging.c, 67
 - paging.h, 41
- get_state
 - syntax.c, 145
 - syntax.h, 147
- get_state_at
 - syntax_highlight.c, 162
- get_token
 - args.c, 87
- getdate
 - dnt.c, 107
 - dnt.h, 116
- getdateHelp
 - help.c, 93, 96
 - out.h, 72
- gettime
 - dnt.c, 107
 - dnt.h, 116
- gettimeHelp
 - help.c, 93, 97
 - out.h, 72
- GREEN
 - colorize.c, 151
 - colorize.h, 154
- head
 - footer, 11
- header, 13
 - index_id, 13
 - size, 14
- heap, 14
 - base, 14
 - index, 14
 - max_size, 14
 - min_size, 15
- heap.c
 - __end, 65
 - _end, 65
 - _kmalloc, 64
 - alloc, 64
 - curr_heap, 65
 - end, 65
 - kdir, 65
 - kheap, 65
 - kmalloc, 64
 - make_heap, 64
 - phys_alloc_addr, 65
- heap.h
 - _kmalloc, 37
 - alloc, 37
 - init_kheap, 38
 - kfree, 38
 - KHEAP_BASE, 37
 - KHEAP_MIN, 37
 - KHEAP_SIZE, 37
 - kmalloc, 38
 - make_heap, 38
 - TABLE_SIZE, 37
- help.c
 - blockHelp, 96
 - cmd_help, 93, 96
 - createpcbHelp, 96
 - deletepcbHelp, 96
 - getdateHelp, 93, 96
 - gettimeHelp, 93, 97
 - helpHelp, 94, 97
 - helpList, 94, 97
 - resumeHelp, 97
 - setdateHelp, 94, 97
 - setpriorityHelp, 97
 - settimeHelp, 94, 98
 - showallpcbHelp, 98
 - showblockedpcbHelp, 98
 - showpcbHelp, 98
 - showreadypcbHelp, 98
 - shutdownHelp, 94, 98
 - suspendHelp, 99
 - unblockHelp, 99
 - versionHelp, 99
 - versionOs, 94
- helpHelp
 - help.c, 94, 97
 - out.h, 72
- helpList
 - help.c, 94, 97
 - out.h, 72
- hint_under_prompt
 - hints.c, 159
 - hints.h, 160
- hints.c
 - hint_under_prompt, 159
- hints.h
 - hint_under_prompt, 160
- hist_discard_last_frame
 - history.c, 122
- hist_forward
 - history.c, 122
 - history.h, 124
- hist_next_frame
 - history.c, 123
 - history.h, 125
- hist_rewind
 - history.c, 123
 - history.h, 125
- history.c
 - circular_next_index, 121

- circular_prev_index, [122](#)
 - hist_discard_last_frame, [122](#)
 - hist_forward, [122](#)
 - hist_next_frame, [123](#)
 - hist_rewind, [123](#)
 - write_hist_to_buf, [123](#)
- history.h
 - hist_forward, [124](#)
 - hist_next_frame, [125](#)
 - hist_rewind, [125](#)
- hlt
 - system.h, [47](#)
- hour
 - date_time, [10](#)
- ICW1
 - interrupts.c, [51](#)
- ICW4
 - interrupts.c, [51](#)
- id
 - index_table, [18](#)
- IDLE
 - mpx_supt.h, [82](#)
- idle
 - mpx_supt.c, [79](#)
 - mpx_supt.h, [84](#)
- idt_entries
 - interrupts.c, [56](#)
 - tables.c, [63](#)
- idt_entry_struct, [15](#)
 - base_high, [15](#)
 - base_low, [15](#)
 - flags, [15](#)
 - sselect, [16](#)
 - zero, [16](#)
- idt_ptr
 - tables.c, [63](#)
- idt_set_gate
 - tables.c, [62](#)
 - tables.h, [34](#)
- idt_struct, [16](#)
 - base, [16](#)
 - limit, [16](#)
- inb
 - io.h, [29](#)
- index
 - heap, [14](#)
- index_entry, [17](#)
 - block, [17](#)
 - empty, [17](#)
 - size, [17](#)
- index_id
 - header, [13](#)
- index_table, [17](#)
 - id, [18](#)
 - table, [18](#)
- init_gdt
 - tables.c, [62](#)
 - tables.h, [34](#)
- init_idt
 - tables.c, [62](#)
 - tables.h, [34](#)
- init_irq
 - interrupts.c, [55](#)
 - interrupts.h, [28](#)
- init_kheap
 - heap.h, [38](#)
- init_paging
 - paging.c, [67](#)
 - paging.h, [41](#)
- init_pic
 - interrupts.c, [55](#)
 - interrupts.h, [28](#)
- init_serial
 - serial.c, [59](#)
 - serial.h, [31](#)
- initPCB
 - pcb.c, [129](#)
 - pcb.h, [139](#)
- insertPCB
 - pcb.c, [129](#)
 - pcb.h, [139](#)
- interrupts.c
 - bounds, [51](#)
 - breakpoint, [51](#)
 - coprocessor, [52](#)
 - coprocessor_segment, [52](#)
 - debug, [52](#)
 - device_not_available, [52](#)
 - divide_error, [52](#)
 - do_bounds, [52](#)
 - do_breakpoint, [52](#)
 - do_coprocessor, [52](#)
 - do_coprocessor_segment, [53](#)
 - do_debug, [53](#)
 - do_device_not_available, [53](#)
 - do_divide_error, [53](#)
 - do_double_fault, [53](#)
 - do_general_protection, [53](#)
 - do_invalid_op, [53](#)
 - do_invalid_tss, [53](#)
 - do_isr, [54](#)
 - do_nmi, [54](#)
 - do_overflow, [54](#)
 - do_page_fault, [54](#)
 - do_reserved, [54](#)
 - do_segment_not_present, [54](#)
 - do_stack_segment, [54](#)
 - double_fault, [54](#)
 - general_protection, [55](#)
 - ICW1, [51](#)
 - ICW4, [51](#)
 - idt_entries, [56](#)
 - init_irq, [55](#)
 - init_pic, [55](#)
 - invalid_op, [55](#)
 - invalid_tss, [55](#)

- io_wait, [51](#)
- isr0, [55](#)
- nmi, [55](#)
- overflow, [56](#)
- page_fault, [56](#)
- PIC1, [51](#)
- PIC2, [51](#)
- reserved, [56](#)
- rtc_isr, [56](#)
- segment_not_present, [56](#)
- stack_segment, [56](#)
- interrupts.h
 - init_irq, [28](#)
 - init_pic, [28](#)
- intToDayOfWeek
 - dnt.c, [108](#)
 - dnt.h, [117](#)
- intToMonth
 - dnt.c, [108](#)
 - dnt.h, [117](#)
- INVALID_BUFFER
 - mpx_supt.h, [82](#)
- INVALID_COUNT
 - mpx_supt.h, [82](#)
- invalid_op
 - interrupts.c, [55](#)
- INVALID_OPERATION
 - mpx_supt.h, [82](#)
- invalid_tss
 - interrupts.c, [55](#)
- io.h
 - inb, [29](#)
 - outb, [29](#)
- IO_MODULE
 - mpx_supt.h, [83](#)
- io_wait
 - interrupts.c, [51](#)
- iret
 - system.h, [47](#)
- is_name_char
 - commhand.c, [103](#)
 - utils.c, [147](#)
 - utils.h, [148](#)
- isr0
 - interrupts.c, [55](#)
- isspace
 - string.c, [77](#)
 - string.h, [43](#)
- itoa
 - string.c, [77](#)
 - string.h, [43](#)
- ltoBCD
 - dnt.c, [109](#)
 - dnt.h, [118](#)
- kdir
 - heap.c, [65](#)
 - paging.c, [68](#)
- kfree
 - heap.h, [38](#)
- kheap
 - heap.c, [65](#)
 - paging.c, [68](#)
- KHEAP_BASE
 - heap.h, [37](#)
- KHEAP_MIN
 - heap.h, [37](#)
- KHEAP_SIZE
 - heap.h, [37](#)
- klogv
 - system.c, [61](#)
 - system.h, [48](#)
- kmain
 - kmain.c, [57](#)
- kmain.c
 - kmain, [57](#)
- kmalloc
 - heap.c, [64](#)
 - heap.h, [38](#)
- kpanic
 - system.c, [61](#)
 - system.h, [48](#)
- last_state
 - args.c, [89](#)
- LEFT_ARROW
 - serial.c, [58](#)
- limit
 - gdt_descriptor_struct, [12](#)
 - idt_struct, [16](#)
 - tables.h, [35](#)
- limit_low
 - gdt_entry_struct, [13](#)
 - tables.h, [35](#)
- load_page_dir
 - paging.c, [67](#)
 - paging.h, [41](#)
- MAGENTA
 - colorize.c, [151](#)
 - colorize.h, [154](#)
- make_heap
 - heap.c, [64](#)
 - heap.h, [38](#)
- mama
 - mama.c, [91](#)
 - mama.h, [91](#)
- mama.c
 - mama, [91](#)
- mama.h
 - mama, [91](#)
- MAX_CMD_ARG_NAME_LEN
 - commhand.h, [104](#)
- MAX_CMD_ARG_VALUE_LEN
 - commhand.h, [104](#)
- MAX_CMD_FLAG_COUNT
 - commhand.h, [104](#)
- MAX_CMD_HIST_LEN

- commhand.h, [104](#)
- MAX_CMD_NAME_LEN
 - commhand.h, [104](#)
- MAX_CMD_NAMED_ARG_COUNT
 - commhand.h, [104](#)
- MAX_CMD_STRING_LEN
 - commhand.h, [105](#)
- MAX_CMD_UNNAMED_ARG_COUNT
 - commhand.h, [105](#)
- MAX_DAY
 - dnt.h, [113](#)
- MAX_HOURS
 - dnt.h, [113](#)
- MAX_MINUTES
 - dnt.h, [113](#)
- MAX_MONTH
 - dnt.h, [114](#)
- MAX_NAME_SIZE
 - pcb.h, [135](#)
- MAX_PARSE_STACK_SIZE
 - args.c, [87](#)
- MAX_PRIORITY
 - pcb.h, [135](#)
- MAX_SECONDS
 - dnt.h, [114](#)
- max_size
 - heap, [14](#)
- MAX_STACK_SIZE
 - pcb.h, [135](#)
- MAX_SYNTAX_SWITCHES
 - syntax_highlight.h, [165](#)
- MAX_YEAR
 - dnt.h, [114](#)
- MEM_MODULE
 - mpx_supt.h, [83](#)
- mem_size
 - paging.c, [68](#)
- memset
 - string.c, [77](#)
 - string.h, [44](#)
- MIN
 - dnt.h, [114](#)
- min
 - date_time, [10](#)
- MIN_DAY
 - dnt.h, [114](#)
- MIN_MONTH
 - dnt.h, [114](#)
- MIN_PRIORITY
 - pcb.h, [135](#)
- min_size
 - heap, [15](#)
- MIN_YEAR
 - dnt.h, [115](#)
- MODULE_F
 - mpx_supt.h, [83](#)
- MODULE_R1
 - mpx_supt.h, [83](#)
- MODULE_R2
 - mpx_supt.h, [83](#)
- MODULE_R3
 - mpx_supt.h, [83](#)
- MODULE_R4
 - mpx_supt.h, [83](#)
- MODULE_R5
 - mpx_supt.h, [83](#)
- mon
 - date_time, [10](#)
- mpx_init
 - mpx_supt.c, [79](#)
 - mpx_supt.h, [84](#)
- mpx_supt.c
 - current_module, [80](#)
 - idle, [79](#)
 - mpx_init, [79](#)
 - params, [80](#)
 - student_free, [80](#)
 - student_malloc, [80](#)
 - sys_alloc_mem, [79](#)
 - sys_free_mem, [79](#)
 - sys_req, [79](#)
 - sys_set_free, [80](#)
 - sys_set_malloc, [80](#)
- mpx_supt.h
 - COM_PORT, [82](#)
 - DEFAULT_DEVICE, [82](#)
 - EXIT, [82](#)
 - FALSE, [82](#)
 - IDLE, [82](#)
 - idle, [84](#)
 - INVALID_BUFFER, [82](#)
 - INVALID_COUNT, [82](#)
 - INVALID_OPERATION, [82](#)
 - IO_MODULE, [83](#)
 - MEM_MODULE, [83](#)
 - MODULE_F, [83](#)
 - MODULE_R1, [83](#)
 - MODULE_R2, [83](#)
 - MODULE_R3, [83](#)
 - MODULE_R4, [83](#)
 - MODULE_R5, [83](#)
 - mpx_init, [84](#)
 - READ, [84](#)
 - sys_alloc_mem, [84](#)
 - sys_free_mem, [84](#)
 - sys_req, [85](#)
 - sys_set_free, [85](#)
 - sys_set_malloc, [85](#)
 - TRUE, [84](#)
 - WRITE, [84](#)
- named_arg
 - args.c, [88](#)
- named_arg_count
 - parsed_args, [22](#)
- named_arg_names
 - parsed_args, [22](#)

- named_arg_values
 - parsed_args, 22
- new_frame
 - paging.c, 67
 - paging.h, 41
- newest_switch
 - syntax_highlight.c, 163
- next_unnamed_arg
 - args.c, 88
- nframes
 - paging.c, 68
- nmi
 - interrupts.c, 55
- NO_ERROR
 - serial.c, 58
- no_warn
 - system.h, 47
- nop
 - system.h, 47
- NULL
 - system.h, 47
- op_code
 - param, 21
- out.c
 - print, 69
 - putc, 69
 - printf, 69
 - println, 69
 - read, 70
- out.h
 - blockHelp, 71
 - cmd_help, 71
 - createpcbHelp, 71
 - deletepcbHelp, 72
 - getdateHelp, 72
 - gettimeHelp, 72
 - helpHelp, 72
 - helpList, 72
 - print, 72
 - putc, 73
 - printf, 73
 - println, 73
 - read, 73
 - resumeHelp, 73
 - setdateHelp, 73
 - setpriorityHelp, 74
 - settimeHelp, 74
 - showallpcbHelp, 74
 - showblockedpcbHelp, 74
 - showpcbHelp, 74
 - showreadypcbHelp, 74
 - shutdownHelp, 75
 - suspendHelp, 75
 - unblockHelp, 75
 - versionHelp, 75
- outb
 - io.h, 29
- overflow
 - interrupts.c, 56
- p_queue
 - pcb.c, 133
- p_state_t
 - pcb.h, 136
- page_dir, 18
 - tables, 18
 - tables_phys, 18
- page_entry, 19
 - accessed, 19
 - dirty, 19
 - frameaddr, 19
 - present, 19
 - reserved, 19
 - usermode, 19
 - writable, 20
- page_fault
 - interrupts.c, 56
- PAGE_SIZE
 - paging.h, 40
- page_size
 - paging.c, 68
- page_table, 20
 - pages, 20
- pages
 - page_table, 20
- paging.c
 - cdir, 68
 - clear_bit, 66
 - find_free, 66
 - frames, 68
 - get_bit, 67
 - get_page, 67
 - init_paging, 67
 - kdir, 68
 - kheap, 68
 - load_page_dir, 67
 - mem_size, 68
 - new_frame, 67
 - nframes, 68
 - page_size, 68
 - phys_alloc_addr, 68
 - set_bit, 67
- paging.h
 - clear_bit, 40
 - first_free, 40
 - get_bit, 41
 - get_page, 41
 - init_paging, 41
 - load_page_dir, 41
 - new_frame, 41
 - PAGE_SIZE, 40
 - set_bit, 41
- param, 20
 - buffer_ptr, 21
 - count_ptr, 21
 - device_id, 21
 - op_code, 21

PARAM_NAME
 syntax.h, 146

PARAM_VALUE
 syntax.h, 146

params
 mpx_supt.c, 80

parse_args
 args.c, 88
 args.h, 90

parse_stack
 args.c, 89

parsed_args, 21
 args.h, 90
 flag_count, 22
 flags, 22
 named_arg_count, 22
 named_arg_names, 22
 named_arg_values, 22
 unnamed_arg_count, 22
 unnamed_args, 22
 unnamed_args_used_so_far, 22

pcb
 pcb_node_t, 23

pcb.c
 allocatePCB, 127
 blockPCB, 127
 createPCB, 127
 deletePCB, 128
 f_queue, 133
 fifo_queue, 133
 findPCB, 128
 freePCB, 128
 initPCB, 129
 insertPCB, 129
 p_queue, 133
 priority_queue, 133
 removePCB, 129
 resumePCB, 130
 setPriority, 130
 setupPCB, 130
 showAll, 131
 showBlocked, 131
 showPCB, 131
 showReady, 132
 suspendPCB, 132
 unblockPCB, 132

pcb.h
 allocatePCB, 137
 BLOCKED, 136
 blockPCB, 137
 createPCB, 137
 deletePCB, 138
 FIFO, 137
 findPCB, 138
 freePCB, 139
 initPCB, 139
 insertPCB, 139
 MAX_NAME_SIZE, 135
 MAX_PRIORITY, 135
 MAX_STACK_SIZE, 135
 MIN_PRIORITY, 135
 p_state_t, 136
 pcb_node_t, 136
 pcb_queue_order_t, 136
 pcb_queue_t, 136
 PRIORITY, 137
 READY, 136
 removePCB, 140
 resumePCB, 140
 RUNNING, 136
 setPriority, 140
 setupPCB, 141
 showAll, 141
 showBlocked, 142
 showPCB, 142
 showReady, 142
 SUSPENDED_BLOCKED, 136
 SUSPENDED_READY, 136
 suspendPCB, 143
 unblockPCB, 143

pcb_name
 pcb_t, 26

pcb_node_t, 23
 pcb, 23
 pcb.h, 136
 pcbn_next_pcb, 23
 pcbn_prev_pcb, 23

pcb_priority
 pcb_t, 26

pcb_process_class
 pcb_t, 26

pcb_process_state
 pcb_t, 26

pcb_queue, 24
 pcbq_count, 24
 pcbq_head, 24
 pcbq_tail, 25
 queue_order, 25

pcb_queue_order_t
 pcb.h, 136

pcb_queue_t
 pcb.h, 136

pcb_stack_bottom
 pcb_t, 26

pcb_stack_top
 pcb_t, 26

pcb_t, 25
 pcb_name, 26
 pcb_priority, 26
 pcb_process_class, 26
 pcb_process_state, 26
 pcb_stack_bottom, 26
 pcb_stack_top, 26

pcbn_next_pcb
 pcb_node_t, 23

pcbn_prev_pcb

- pcb_node_t, 23
- pcbq_count
 - pcb_queue, 24
- pcbq_head
 - pcb_queue, 24
- pcbq_tail
 - pcb_queue, 25
- phys_alloc_addr
 - heap.c, 65
 - paging.c, 68
- PIC1
 - interrupts.c, 51
- PIC2
 - interrupts.c, 51
- polling
 - serial.c, 59
 - serial.h, 31
- present
 - page_entry, 19
- print
 - out.c, 69
 - out.h, 72
- print_color_code
 - colorize.c, 152
- putc
 - out.c, 69
 - out.h, 73
- printf
 - out.c, 69
 - out.h, 73
- println
 - out.c, 69
 - out.h, 73
- PRIORITY
 - pcb.h, 137
- priority_queue
 - pcb.c, 133
- queue_order
 - pcb_queue, 25
- READ
 - mpx_supt.h, 84
- read
 - out.c, 70
 - out.h, 73
- READY
 - pcb.h, 136
- RED
 - colorize.c, 151
 - colorize.h, 154
- removePCB
 - pcb.c, 129
 - pcb.h, 140
- reserved
 - interrupts.c, 56
 - page_entry, 19
- resumeHelp
 - help.c, 97
- out.h, 73
- resumePCB
 - pcb.c, 130
 - pcb.h, 140
- RIGHT_ARROW
 - serial.c, 59
- rtc_isr
 - interrupts.c, 56
- RUNNING
 - pcb.h, 136
- sec
 - date_time, 11
- segment_not_present
 - interrupts.c, 56
- serial.c
 - consume_special, 59
 - DELETE, 58
 - DOWN_ARROW, 58
 - init_serial, 59
 - LEFT_ARROW, 58
 - NO_ERROR, 58
 - polling, 59
 - RIGHT_ARROW, 59
 - serial_port_in, 60
 - serial_port_out, 60
 - serial_print, 60
 - serial_println, 60
 - set_serial_in, 60
 - set_serial_out, 60
 - UP_ARROW, 59
- serial.h
 - COM1, 30
 - COM2, 30
 - COM3, 30
 - COM4, 30
 - init_serial, 31
 - polling, 31
 - serial_print, 31
 - serial_println, 31
 - set_serial_in, 31
 - set_serial_out, 32
- serial_port_in
 - serial.c, 60
- serial_port_out
 - serial.c, 60
- serial_print
 - serial.c, 60
 - serial.h, 31
- serial_println
 - serial.c, 60
 - serial.h, 31
- set_bit
 - paging.c, 67
 - paging.h, 41
- set_serial_in
 - serial.c, 60
 - serial.h, 31
- set_serial_out

- serial.c, 60
- serial.h, 32
- setdate
 - dnt.c, 109
 - dnt.h, 118
- setdateHelp
 - help.c, 94, 97
 - out.h, 73
- setDateInMemory
 - dnt.c, 109
 - dnt.h, 119
- setPriority
 - pcb.c, 130
 - pcb.h, 140
- setpriorityHelp
 - help.c, 97
 - out.h, 74
- settime
 - dnt.c, 110
 - dnt.h, 119
- settimeHelp
 - help.c, 94, 98
 - out.h, 74
- setTimeInMemory
 - dnt.c, 110
 - dnt.h, 120
- setupPCB
 - pcb.c, 130
 - pcb.h, 141
- showAll
 - pcb.c, 131
 - pcb.h, 141
- showallpcbHelp
 - help.c, 98
 - out.h, 74
- showBlocked
 - pcb.c, 131
 - pcb.h, 142
- showblockedpcbHelp
 - help.c, 98
 - out.h, 74
- showPCB
 - pcb.c, 131
 - pcb.h, 142
- showpcbHelp
 - help.c, 98
 - out.h, 74
- showReady
 - pcb.c, 132
 - pcb.h, 142
- showreadypcbHelp
 - help.c, 98
 - out.h, 74
- shutdown.c
 - cmd_shutdown, 99
- shutdownHelp
 - help.c, 94, 98
 - out.h, 75
- SINGLE_QUOTE_STRING
 - syntax.h, 146
- SINGLE_QUOTE_STRING_END_QUOTE
 - syntax.h, 146
- size
 - header, 14
 - index_entry, 17
- size_t
 - system.h, 48
- skip_ws
 - utils.c, 148
 - utils.h, 150
- sselect
 - idt_entry_struct, 16
 - tables.h, 35
- stack_empty
 - args.c, 88
- stack_peek
 - args.c, 88
- stack_pop
 - args.c, 88
- stack_push
 - args.c, 89
- stack_segment
 - interrupts.c, 56
- stack_size
 - args.c, 89
- START_SEQ
 - colorize.c, 151
- states
 - syntax_highlight.c, 164
- sti
 - system.h, 47
- strcat
 - string.c, 77
 - string.h, 44
- strcmp
 - string.c, 78
 - string.h, 44
- strcpy
 - string.c, 78
 - string.h, 44
- string.c
 - atoi, 77
 - isspace, 77
 - itoa, 77
 - memset, 77
 - strcat, 77
 - strcmp, 78
 - strcpy, 78
 - strlen, 78
 - strtok, 78
- string.h
 - atoi, 43
 - isspace, 43
 - itoa, 43
 - memset, 44
 - strcat, 44

- strcmp, 44
- strcpy, 44
- strlen, 44
- strtok, 44
- strlen
 - string.c, 78
 - string.h, 44
- strtok
 - string.c, 78
 - string.h, 44
- student_free
 - mpx_supt.c, 80
- student_malloc
 - mpx_supt.c, 80
- SUSPENDED_BLOCKED
 - pcb.h, 136
- SUSPENDED_READY
 - pcb.h, 136
- suspendHelp
 - help.c, 99
 - out.h, 75
- suspendPCB
 - pcb.c, 132
 - pcb.h, 143
- switch_indexes
 - syntax_highlight.c, 164
- switch_to
 - syntax_highlight.c, 162
- syntax.c
 - changes_state, 145
 - get_state, 145
- syntax.h
 - changes_state, 146
 - CMD_NAME, 146
 - CMD_NAME_OR_LEADING_WHITESPACE, 146
 - DEFAULT, 146
 - DOUBLE_QUOTE_STRING, 146
 - DOUBLE_QUOTE_STRING_END_QUOTE, 146
 - END_OF_INPUT, 146
 - get_state, 147
 - PARAM_NAME, 146
 - PARAM_VALUE, 146
 - SINGLE_QUOTE_STRING, 146
 - SINGLE_QUOTE_STRING_END_QUOTE, 146
 - SyntaxState, 146
- SYNTAX_COLOR_CMD_NAME
 - syntax_highlight.h, 165
- SYNTAX_COLOR_DEFAULT
 - syntax_highlight.h, 165
- SYNTAX_COLOR_DOUBLE_QUOTE_STRING
 - syntax_highlight.h, 165
- SYNTAX_COLOR_PARAM_NAME
 - syntax_highlight.h, 165
- SYNTAX_COLOR_PARAM_VALUE
 - syntax_highlight.h, 165
- SYNTAX_COLOR_SINGLE_QUOTE_STRING
 - syntax_highlight.h, 165
- syntax_disable_highlighting
 - syntax_highlight.c, 162
 - syntax_highlight.h, 166
- syntax_enable_highlighting
 - syntax_highlight.c, 163
 - syntax_highlight.h, 166
- syntax_handle_char
 - syntax_highlight.c, 163
 - syntax_highlight.h, 166
- syntax_highlight.c
 - color_for, 161
 - enabled, 163
 - get_state_at, 162
 - newest_switch, 163
 - states, 164
 - switch_indexes, 164
 - switch_to, 162
 - syntax_disable_highlighting, 162
 - syntax_enable_highlighting, 163
 - syntax_handle_char, 163
 - syntax_init, 163
- syntax_highlight.h
 - MAX_SYNTAX_SWITCHES, 165
 - SYNTAX_COLOR_CMD_NAME, 165
 - SYNTAX_COLOR_DEFAULT, 165
 - SYNTAX_COLOR_DOUBLE_QUOTE_STRING, 165
 - SYNTAX_COLOR_PARAM_NAME, 165
 - SYNTAX_COLOR_PARAM_VALUE, 165
 - SYNTAX_COLOR_SINGLE_QUOTE_STRING, 165
 - syntax_disable_highlighting, 166
 - syntax_enable_highlighting, 166
 - syntax_handle_char, 166
 - syntax_init, 166
- syntax_init
 - syntax_highlight.c, 163
 - syntax_highlight.h, 166
- SyntaxState
 - syntax.h, 146
- sys_alloc_mem
 - mpx_supt.c, 79
 - mpx_supt.h, 84
- sys_free_mem
 - mpx_supt.c, 79
 - mpx_supt.h, 84
- sys_req
 - mpx_supt.c, 79
 - mpx_supt.h, 85
- sys_set_free
 - mpx_supt.c, 80
 - mpx_supt.h, 85
- sys_set_malloc
 - mpx_supt.c, 80
 - mpx_supt.h, 85
- system.c
 - klogv, 61
 - kpanic, 61
- system.h

- asm, 46
- cli, 46
- GDT_CS_ID, 46
- GDT_DS_ID, 47
- hlt, 47
- iret, 47
- klogv, 48
- kpanic, 48
- no_warn, 47
- nop, 47
- NULL, 47
- size_t, 48
- sti, 47
- u16int, 48
- u32int, 48
- u8int, 48
- volatile, 48
- table
 - index_table, 18
- TABLE_SIZE
 - heap.h, 37
- tables
 - page_dir, 18
- tables.c
 - gdt_entries, 63
 - gdt_init_entry, 62
 - gdt_ptr, 63
 - idt_entries, 63
 - idt_ptr, 63
 - idt_set_gate, 62
 - init_gdt, 62
 - init_idt, 62
 - write_gdt_ptr, 62
 - write_idt_ptr, 63
- tables.h
 - __attribute__, 33
 - access, 34
 - base, 34
 - base_high, 34
 - base_low, 35
 - base_mid, 35
 - flags, 35
 - gdt_init_entry, 33
 - idt_set_gate, 34
 - init_gdt, 34
 - init_idt, 34
 - limit, 35
 - limit_low, 35
 - sselect, 35
 - zero, 35
- tables_phys
 - page_dir, 18
- TRUE
 - mpx_supt.h, 84
- u16int
 - system.h, 48
- u32int
 - system.h, 48
- u8int
 - system.h, 48
- unblockHelp
 - help.c, 99
 - out.h, 75
- unblockPCB
 - pcb.c, 132
 - pcb.h, 143
- unnamed_arg_count
 - parsed_args, 22
- unnamed_args
 - parsed_args, 22
- unnamed_args_used_so_far
 - parsed_args, 22
- UP_ARROW
 - serial.c, 59
- usermode
 - page_entry, 19
- utils.c
 - is_name_char, 147
 - skip_ws, 148
- utils.h
 - is_name_char, 148
 - skip_ws, 150
- version.c
 - cmd_version, 100
- versionHelp
 - help.c, 99
 - out.h, 75
- versionOs
 - help.c, 94
- volatile
 - system.h, 48
- WHITE
 - colorize.c, 151
 - colorize.h, 154
- WRITE
 - mpx_supt.h, 84
- write_gdt_ptr
 - tables.c, 62
- write_hist_to_buf
 - history.c, 123
- write_idt_ptr
 - tables.c, 63
- writeable
 - page_entry, 20
- year
 - date_time, 11
- YELLOW
 - colorize.c, 151
 - colorize.h, 154
- zero
 - idt_entry_struct, 16
 - tables.h, 35