

# CATernel

Generated by Doxygen 1.7.6.1

Tue Oct 23 2012 15:46:29



# Contents

<b>1</b>	<b>Class Index</b>	<b>1</b>
1.1	Class List . . . . .	1
<b>2</b>	<b>File Index</b>	<b>3</b>
2.1	File List . . . . .	3
<b>3</b>	<b>Class Documentation</b>	<b>5</b>
3.1	__attribute__ Struct Reference . . . . .	5
3.1.1	Member Data Documentation . . . . .	7
3.1.1.1	_rsrvd1 . . . . .	7
3.1.1.2	_rsrvd2 . . . . .	7
3.1.1.3	_rsrvd3 . . . . .	7
3.1.1.4	_rsrvd4 . . . . .	7
3.1.1.5	_rsrvd5 . . . . .	7
3.1.1.6	_rsrvd6 . . . . .	7
3.1.1.7	_rsrvd7 . . . . .	7
3.1.1.8	_rsrvd8 . . . . .	7
3.1.1.9	_rsrvd9 . . . . .	7
3.1.1.10	_rsrvda . . . . .	7
3.1.1.11	_rsrvdb . . . . .	7
3.1.1.12	access . . . . .	7
3.1.1.13	accessed . . . . .	7
3.1.1.14	accessible . . . . .	7
3.1.1.15	address . . . . .	7
3.1.1.16	args . . . . .	7
3.1.1.17	available . . . . .	7

3.1.1.18	base_0_15	7
3.1.1.19	base_16_23	7
3.1.1.20	base_24_31	7
3.1.1.21	cache_disable	7
3.1.1.22	cr3	7
3.1.1.23	cs	7
3.1.1.24	dirty	8
3.1.1.25	dpl	8
3.1.1.26	ds	8
3.1.1.27	eax	8
3.1.1.28	ebp	8
3.1.1.29	ebx	8
3.1.1.30	ecx	8
3.1.1.31	edi	8
3.1.1.32	edx	8
3.1.1.33	eflags	8
3.1.1.34	eip	8
3.1.1.35	es	8
3.1.1.36	esi	8
3.1.1.37	esp	8
3.1.1.38	esp0	8
3.1.1.39	esp1	8
3.1.1.40	esp2	8
3.1.1.41	fs	8
3.1.1.42	global	8
3.1.1.43	granularity	8
3.1.1.44	gs	8
3.1.1.45	ignore_this	8
3.1.1.46	iomap_base	8
3.1.1.47	ldt	8
3.1.1.48	limit_0_15	9
3.1.1.49	limit_16_19	9
3.1.1.50	offset_0_15	9
3.1.1.51	offset_16_31	9

3.1.1.52	p	9
3.1.1.53	pat	9
3.1.1.54	prelink	9
3.1.1.55	present	9
3.1.1.56	reserved	9
3.1.1.57	s	9
3.1.1.58	segment_s	9
3.1.1.59	ss	9
3.1.1.60	ss0	9
3.1.1.61	ss1	9
3.1.1.62	ss2	9
3.1.1.63	trace	9
3.1.1.64	type	9
3.1.1.65	unused	9
3.1.1.66	writable	9
3.1.1.67	write_through	9
3.2	cpu_state_t Struct Reference	10
3.2.1	Detailed Description	10
3.2.2	Member Data Documentation	10
3.2.2.1	cs	10
3.2.2.2	ds	10
3.2.2.3	eax	10
3.2.2.4	ebp_frame	10
3.2.2.5	ebx	11
3.2.2.6	ecx	11
3.2.2.7	edi	11
3.2.2.8	edx	11
3.2.2.9	eflags	11
3.2.2.10	eip	11
3.2.2.11	eip_frame	11
3.2.2.12	error_code	11
3.2.2.13	es	11
3.2.2.14	esi	11
3.2.2.15	esp	11

3.2.2.16	fs	.....	11
3.2.2.17	gs	.....	11
3.2.2.18	o_ebp	.....	11
3.2.2.19	o_esp	.....	11
3.2.2.20	ss	.....	11
3.3	cpuid_regs Struct Reference	.....	11
3.3.1	Member Data Documentation	.....	12
3.3.1.1	eax	.....	12
3.3.1.2	ebx	.....	12
3.3.1.3	ecx	.....	12
3.3.1.4	edx	.....	12
3.4	element Struct Reference	.....	12
3.5	elfhdr Struct Reference	.....	12
3.5.1	Member Data Documentation	.....	13
3.5.1.1	ehsize	.....	13
3.5.1.2	entry	.....	13
3.5.1.3	flags	.....	13
3.5.1.4	machine	.....	13
3.5.1.5	magic	.....	13
3.5.1.6	magic2	.....	13
3.5.1.7	phrnum	.....	13
3.5.1.8	phroff	.....	13
3.5.1.9	phrsize	.....	13
3.5.1.10	shrnun	.....	13
3.5.1.11	shroff	.....	13
3.5.1.12	shrsiz	.....	13
3.5.1.13	shrsttbl	.....	13
3.5.1.14	type	.....	13
3.5.1.15	version	.....	13
3.6	ext2_dir_entry Struct Reference	.....	13
3.6.1	Member Data Documentation	.....	14
3.6.1.1	inode	.....	14
3.6.1.2	name	.....	14
3.6.1.3	name_len	.....	14

3.6.1.4	rec_len . . . . .	14
3.7	ext2_group_desc Struct Reference . . . . .	14
3.7.1	Member Data Documentation . . . . .	14
3.7.1.1	bg_block_bitmap . . . . .	14
3.7.1.2	bg_free_blocks_count . . . . .	14
3.7.1.3	bg_free_inodes_count . . . . .	14
3.7.1.4	bg_inode_bitmap . . . . .	14
3.7.1.5	bg_inode_table . . . . .	14
3.7.1.6	bg_pad . . . . .	14
3.7.1.7	bg_reserved . . . . .	14
3.7.1.8	bg_used_dirs_count . . . . .	15
3.8	ext2_inode Struct Reference . . . . .	15
3.8.1	Member Data Documentation . . . . .	16
3.8.1.1	h_i_author . . . . .	16
3.8.1.2	h_i_frag . . . . .	16
3.8.1.3	h_i_fsize . . . . .	16
3.8.1.4	h_i_gid_high . . . . .	16
3.8.1.5	h_i_mode_high . . . . .	16
3.8.1.6	h_i_uid_high . . . . .	16
3.8.1.7	hurd2 . . . . .	16
3.8.1.8	i_atime . . . . .	16
3.8.1.9	i_block . . . . .	16
3.8.1.10	i_blocks . . . . .	16
3.8.1.11	i_ctime . . . . .	16
3.8.1.12	i_dir_acl . . . . .	16
3.8.1.13	i_dtime . . . . .	16
3.8.1.14	i_faddr . . . . .	16
3.8.1.15	i_file_acl . . . . .	16
3.8.1.16	i_flags . . . . .	16
3.8.1.17	i_gid . . . . .	16
3.8.1.18	i_links_count . . . . .	16
3.8.1.19	i_mode . . . . .	17
3.8.1.20	i_mtime . . . . .	17
3.8.1.21	i_reserved1 . . . . .	17

3.8.1.22	<code>i_size</code>	17
3.8.1.23	<code>i_uid</code>	17
3.8.1.24	<code>i_version</code>	17
3.8.1.25	<code>l_i_frag</code>	17
3.8.1.26	<code>l_i_fsize</code>	17
3.8.1.27	<code>l_i_gid_high</code>	17
3.8.1.28	<code>l_i_reserved1</code>	17
3.8.1.29	<code>l_i_reserved2</code>	17
3.8.1.30	<code>l_i_uid_high</code>	17
3.8.1.31	<code>linux</code>	17
3.8.1.32	<code>m_i_frag</code>	17
3.8.1.33	<code>m_i_fsize</code>	17
3.8.1.34	<code>m_i_reserved2</code>	17
3.8.1.35	<code>m_pad1</code>	17
3.8.1.36	<code>masix2</code>	17
3.8.1.37	<code>osd2</code>	17
3.9	<code>ext2_super_block</code> Struct Reference	17
3.9.1	Member Data Documentation	19
3.9.1.1	<code>s_algorithm_usage_bitmap</code>	19
3.9.1.2	<code>s_block_group_nr</code>	19
3.9.1.3	<code>s_blocks_count</code>	19
3.9.1.4	<code>s_blocks_per_group</code>	19
3.9.1.5	<code>s_checkinterval</code>	19
3.9.1.6	<code>s_errors</code>	19
3.9.1.7	<code>s_feature_compat</code>	19
3.9.1.8	<code>s_feature_incompat</code>	19
3.9.1.9	<code>s_feature_ro_compat</code>	19
3.9.1.10	<code>s_first_data_block</code>	19
3.9.1.11	<code>s_first_ino</code>	19
3.9.1.12	<code>s_frags_per_group</code>	19
3.9.1.13	<code>s_free_blocks_count</code>	19
3.9.1.14	<code>s_free_inodes_count</code>	19
3.9.1.15	<code>s_gid</code>	19
3.9.1.16	<code>s_inode_size</code>	19

3.9.1.17	s_inodes_count . . . . .	19
3.9.1.18	s_inodes_per_group . . . . .	19
3.9.1.19	s_journal_dev . . . . .	19
3.9.1.20	s_journal_inum . . . . .	19
3.9.1.21	s_journal_uuid . . . . .	19
3.9.1.22	s_last_mounted . . . . .	19
3.9.1.23	s_last_orphan . . . . .	19
3.9.1.24	s_lastcheck . . . . .	20
3.9.1.25	s_log_block_size . . . . .	20
3.9.1.26	s_log_frag_size . . . . .	20
3.9.1.27	s_magic . . . . .	20
3.9.1.28	s_maj_pad . . . . .	20
3.9.1.29	s_max_mnt_count . . . . .	20
3.9.1.30	s_min_pad . . . . .	20
3.9.1.31	s_mnt_count . . . . .	20
3.9.1.32	s_mtime . . . . .	20
3.9.1.33	s_os_id . . . . .	20
3.9.1.34	s_padding1 . . . . .	20
3.9.1.35	s_prealloc_blocks . . . . .	20
3.9.1.36	s_prealloc_dir_blocks . . . . .	20
3.9.1.37	s_r_blocks_count . . . . .	20
3.9.1.38	s_reserved . . . . .	20
3.9.1.39	s_state . . . . .	20
3.9.1.40	s_uid . . . . .	20
3.9.1.41	s_uuid . . . . .	20
3.9.1.42	s_volume_name . . . . .	20
3.9.1.43	s_wtime . . . . .	20
3.10	Gdtdesc Struct Reference . . . . .	20
3.10.1	Member Data Documentation . . . . .	21
3.10.1.1	base . . . . .	21
3.10.1.2	size . . . . .	21
3.11	gpr_regs_t Struct Reference . . . . .	21
3.11.1	Member Data Documentation . . . . .	21
3.11.1.1	eax . . . . .	21

---

3.11.1.2  ebp . . . . .	21
3.11.1.3  ebx . . . . .	21
3.11.1.4  ecx . . . . .	21
3.11.1.5  edi . . . . .	21
3.11.1.6  edx . . . . .	21
3.11.1.7  esi . . . . .	22
3.11.1.8  esp . . . . .	22
3.12  Page Struct Reference . . . . .	22
3.12.1  Member Data Documentation . . . . .	22
3.12.1.1  link . . . . .	22
3.12.1.2  ref . . . . .	22
3.13  proc Struct Reference . . . . .	22
3.13.1  Member Function Documentation . . . . .	23
3.13.1.1  LIFO_ENTRY . . . . .	23
3.13.1.2  LIST_ENTRY . . . . .	23
3.13.2  Member Data Documentation . . . . .	24
3.13.2.1  cr3 . . . . .	24
3.13.2.2  cs . . . . .	24
3.13.2.3  dequeued . . . . .	24
3.13.2.4  eflags . . . . .	24
3.13.2.5  eip . . . . .	24
3.13.2.6  esp . . . . .	24
3.13.2.7  gpr_regs . . . . .	24
3.13.2.8  page_directory . . . . .	24
3.13.2.9  preempted . . . . .	24
3.13.2.10  proc_id . . . . .	24
3.13.2.11  proc_status . . . . .	24
3.13.2.12  seg_regs . . . . .	24
3.13.2.13  ss . . . . .	24
3.14  proghdr Struct Reference . . . . .	24
3.14.1  Member Data Documentation . . . . .	25
3.14.1.1  align . . . . .	25
3.14.1.2  filesz . . . . .	25
3.14.1.3  flags . . . . .	25

3.14.1.4	memsz . . . . .	25
3.14.1.5	offset . . . . .	25
3.14.1.6	paddr . . . . .	25
3.14.1.7	type . . . . .	25
3.14.1.8	vaddr . . . . .	25
3.15	sechdr Struct Reference . . . . .	25
3.15.1	Member Data Documentation . . . . .	25
3.15.1.1	addr . . . . .	25
3.15.1.2	addralign . . . . .	25
3.15.1.3	entsize . . . . .	26
3.15.1.4	flags . . . . .	26
3.15.1.5	info . . . . .	26
3.15.1.6	link . . . . .	26
3.15.1.7	name . . . . .	26
3.15.1.8	offset . . . . .	26
3.15.1.9	size . . . . .	26
3.15.1.10	type . . . . .	26
3.16	seg_regs_t Struct Reference . . . . .	26
3.16.1	Member Data Documentation . . . . .	26
3.16.1.1	ds . . . . .	26
3.16.1.2	es . . . . .	26
3.16.1.3	fs . . . . .	26
3.16.1.4	gs . . . . .	26
3.17	Segdesc Struct Reference . . . . .	27
3.17.1	Member Data Documentation . . . . .	27
3.17.1.1	base . . . . .	27
3.17.1.2	base_0 . . . . .	27
3.17.1.3	base_1 . . . . .	27
3.17.1.4	flags . . . . .	27
3.17.1.5	limit_0 . . . . .	27
3.17.1.6	limit_1 . . . . .	27
3.17.1.7	permission . . . . .	27

4.1	include/arch/x86/cpu_state.h File Reference . . . . .	29
4.1.1	Define Documentation . . . . .	30
4.1.1.1	SEGMENT_TSS . . . . .	30
4.1.1.2	TASK_BUSY . . . . .	31
4.1.1.3	TASK_INACTIVE . . . . .	31
4.1.1.4	TSS_AVL . . . . .	31
4.1.1.5	TSS_DPL_KERNEL . . . . .	31
4.1.1.6	TSS_DPL_USER . . . . .	31
4.1.1.7	TSS_GRANULARITY . . . . .	31
4.1.1.8	TSS_IOMAP_SIZE . . . . .	31
4.1.1.9	TSS_PRESENT . . . . .	31
4.1.1.10	TSS_SIZE . . . . .	31
4.2	include/arch/x86/elf.h File Reference . . . . .	31
4.2.1	Define Documentation . . . . .	33
4.2.1.1	ELF_MAGIC . . . . .	33
4.2.1.2	M_ABIVERSION . . . . .	33
4.2.1.3	M_CLASS32 . . . . .	33
4.2.1.4	M_CLASS64 . . . . .	33
4.2.1.5	M_CLASS_OFF . . . . .	33
4.2.1.6	M_CLASSNONE . . . . .	33
4.2.1.7	M_CLASSNUM . . . . .	33
4.2.1.8	M_DATA2BE . . . . .	33
4.2.1.9	M_DATA2LE . . . . .	33
4.2.1.10	M_DATA_OFF . . . . .	33
4.2.1.11	M_DATANONE . . . . .	33
4.2.1.12	M_DATANUM . . . . .	33
4.2.1.13	M_ELF_PADDING . . . . .	33
4.2.1.14	M_MACHINE_I386 . . . . .	33
4.2.1.15	M_OSABI . . . . .	33
4.2.1.16	M_OSABI_HPUX . . . . .	33
4.2.1.17	M_OSABI_SYSV . . . . .	33
4.2.1.18	M_VERSION . . . . .	33
4.2.1.19	MAGIC_LEN . . . . .	33
4.2.1.20	P_PROGHDR_E . . . . .	33

4.2.1.21	P_PROGHDR_R . . . . .	33
4.2.1.22	P_PROGHDR_W . . . . .	33
4.2.1.23	PRG_LOAD . . . . .	33
4.2.1.24	T_TYPE_CORE . . . . .	34
4.2.1.25	T_TYPE_DYN . . . . .	34
4.2.1.26	T_TYPE_EXEC . . . . .	34
4.2.1.27	T_TYPE_HIPROC . . . . .	34
4.2.1.28	T_TYPE_LOPROC . . . . .	34
4.2.1.29	T_TYPE_NONE . . . . .	34
4.2.1.30	T_TYPE_REL . . . . .	34
4.2.1.31	V_VERSION_CURRENT . . . . .	34
4.2.1.32	V_VERSION_NONE . . . . .	34
4.2.1.33	V_VERSION_NUM . . . . .	34
4.2.2	Typedef Documentation . . . . .	34
4.2.2.1	elfhdr . . . . .	34
4.2.2.2	prohdr . . . . .	34
4.2.2.3	secthdr . . . . .	34
4.3	include/arch/x86/interrupt.h File Reference . . . . .	35
4.3.1	Define Documentation . . . . .	36
4.3.1.1	_CATERNEL_X86_INTERRUPT_H_ . . . . .	36
4.3.1.2	GATE_FILL . . . . .	37
4.3.1.3	GATE_INT16 . . . . .	37
4.3.1.4	GATE_INT32 . . . . .	37
4.3.1.5	GATE_OFFSET . . . . .	37
4.3.1.6	GATE_TASK . . . . .	37
4.3.1.7	GATE_TRAP16 . . . . .	37
4.3.1.8	GATE_TRAP32 . . . . .	37
4.3.1.9	GP . . . . .	37
4.3.1.10	IDT_ENTRIES . . . . .	37
4.3.1.11	PF . . . . .	37
4.3.1.12	PF_FROM_KERNEL . . . . .	37
4.3.1.13	PF_FROM_USER . . . . .	37
4.3.1.14	PF_IFETCH . . . . .	37
4.3.1.15	PF_NOT_PRESENT . . . . .	37

4.3.1.16	PF_ON_READ . . . . .	38
4.3.1.17	PF_ON_WRITE . . . . .	38
4.3.1.18	PF_RSRVD . . . . .	38
4.3.1.19	PF_VIOLATION . . . . .	38
4.3.1.20	VECTOR_INDEX . . . . .	38
4.3.2	Typedef Documentation . . . . .	38
4.3.2.1	Idtdesc . . . . .	38
4.3.3	Function Documentation . . . . .	38
4.3.3.1	idt_init . . . . .	38
4.3.3.2	map_exception . . . . .	39
4.3.3.3	page_fault_handler . . . . .	39
4.3.4	Variable Documentation . . . . .	39
4.3.4.1	idt . . . . .	39
4.3.4.2	x86_exception_names . . . . .	40
4.4	include/arch/x86/mm/page.h File Reference . . . . .	40
4.4.1	Define Documentation . . . . .	42
4.4.1.1	KA2PA . . . . .	42
4.4.1.2	PA2KA . . . . .	42
4.4.1.3	PAGE_ACCESSIONED . . . . .	43
4.4.1.4	PAGE_CDISABLE . . . . .	43
4.4.1.5	PAGE_DIRTY . . . . .	43
4.4.1.6	PAGE_DIRTY . . . . .	43
4.4.1.7	PAGE_GLOBAL . . . . .	43
4.4.1.8	PAGE_PAGESZ . . . . .	43
4.4.1.9	PAGE_PRESENT . . . . .	43
4.4.1.10	PAGE_USER . . . . .	43
4.4.1.11	PAGE_WRITABLE . . . . .	43
4.4.1.12	PAGE_WTHROUGH . . . . .	43
4.4.1.13	PGDIR_SHIFT . . . . .	43
4.4.1.14	PGDIRX . . . . .	43
4.4.1.15	PGOFF . . . . .	43
4.4.1.16	PGPN . . . . .	43
4.4.1.17	PGSHIFT . . . . .	43
4.4.1.18	PGTBL_SHIFT . . . . .	43

4.4.1.19	PGTBLX . . . . .	43
4.4.1.20	PTD_ADDR . . . . .	43
4.4.1.21	PTSHIFT . . . . .	43
4.4.1.22	VIRT_PGDIRX . . . . .	43
4.4.1.23	VIRT_PGPN . . . . .	43
4.4.1.24	VIRT_PGTBLX . . . . .	43
4.4.2	Typedef Documentation . . . . .	43
4.4.2.1	pde_t . . . . .	43
4.4.3	Function Documentation . . . . .	44
4.4.3.1	LIST_ENTRY . . . . .	44
4.4.3.2	LIST_HEAD . . . . .	44
4.4.3.3	map_segment_page . . . . .	44
4.4.3.4	x86_page_alloc . . . . .	44
4.4.3.5	x86_page_detach . . . . .	44
4.4.3.6	x86_page_free . . . . .	45
4.4.3.7	x86_page_init . . . . .	45
4.4.3.8	x86_page_insert . . . . .	45
4.4.3.9	x86_page_lookup . . . . .	45
4.4.3.10	x86_page_remove . . . . .	46
4.4.3.11	x86_paging_init . . . . .	46
4.4.3.12	x86_pgdir_find . . . . .	46
4.4.3.13	x86_test.paging . . . . .	46
4.4.3.14	x86_test_pgdir . . . . .	47
4.4.4	Variable Documentation . . . . .	47
4.4.4.1	gdt . . . . .	47
4.4.4.2	global_cr3 . . . . .	47
4.4.4.3	global_pgdir . . . . .	47
4.4.4.4	kernel_stack . . . . .	47
4.4.4.5	kernel_stack_end . . . . .	47
4.4.4.6	next_free . . . . .	47
4.4.4.7	page_count . . . . .	47
4.4.4.8	pages . . . . .	47
4.4.4.9	virtpgd . . . . .	47
4.4.4.10	virttgt . . . . .	47

4.5	include/arch/x86/mm/segdesc.h File Reference . . . . .	48
4.5.1	Define Documentation . . . . .	48
4.5.1.1	SEG_KERNCODE . . . . .	48
4.5.1.2	SEG_KERNDATA . . . . .	48
4.5.1.3	SEG_TSS . . . . .	48
4.5.1.4	SEG_USERCODE . . . . .	48
4.5.1.5	SEG_USERDATA . . . . .	48
4.6	include/arch/x86/processor.h File Reference . . . . .	49
4.6.1	Define Documentation . . . . .	50
4.6.1.1	FLAG_AC . . . . .	50
4.6.1.2	FLAG_AF . . . . .	50
4.6.1.3	FLAG_CF . . . . .	50
4.6.1.4	FLAG_DF . . . . .	50
4.6.1.5	FLAG_ID . . . . .	50
4.6.1.6	FLAG_IF . . . . .	50
4.6.1.7	FLAG_IOPL . . . . .	50
4.6.1.8	FLAG_NT . . . . .	50
4.6.1.9	FLAG_OF . . . . .	50
4.6.1.10	FLAG_PF . . . . .	50
4.6.1.11	FLAG_RF . . . . .	50
4.6.1.12	FLAG_SF . . . . .	50
4.6.1.13	FLAG_TF . . . . .	51
4.6.1.14	FLAG_VIF . . . . .	51
4.6.1.15	FLAG_VIP . . . . .	51
4.6.1.16	FLAG_VM . . . . .	51
4.6.1.17	FLAG_ZF . . . . .	51
4.6.1.18	X86_CR0_AM . . . . .	51
4.6.1.19	X86_CR0_CD . . . . .	51
4.6.1.20	X86_CR0_EM . . . . .	51
4.6.1.21	X86_CR0_ET . . . . .	51
4.6.1.22	X86_CR0_MP . . . . .	51
4.6.1.23	X86_CR0_NE . . . . .	51
4.6.1.24	X86_CR0_NW . . . . .	51
4.6.1.25	X86_CR0_PE . . . . .	51

4.6.1.26	X86_CR0_PG	51
4.6.1.27	X86_CR0_TS	51
4.6.1.28	X86_CR0_WP	51
4.6.1.29	X86_CR3_PCD	51
4.6.1.30	X86_CR3_PWT	51
4.6.1.31	X86_CR4_DE	51
4.6.1.32	X86_CR4_MCE	51
4.6.1.33	X86_CR4_OSFXSR	51
4.6.1.34	X86_CR4_OSXMMEXCPT	51
4.6.1.35	X86_CR4 OSXSAVE	51
4.6.1.36	X86_CR4_PAE	52
4.6.1.37	X86_CR4_PCE	52
4.6.1.38	X86_CR4_PGE	52
4.6.1.39	X86_CR4_PSE	52
4.6.1.40	X86_CR4_PVI	52
4.6.1.41	X86_CR4_TSD	52
4.6.1.42	X86_CR4_VME	52
4.6.1.43	X86_CR4_VMXE	52
4.6.1.44	X86_CR8_TPR	52
4.7	include/arch/x86/vectors.h File Reference	52
4.7.1	Variable Documentation	54
4.7.1.1	vector_0	54
4.7.1.2	vector_1	54
4.7.1.3	vector_10	54
4.7.1.4	vector_11	54
4.7.1.5	vector_12	54
4.7.1.6	vector_13	54
4.7.1.7	vector_14	54
4.7.1.8	vector_15	54
4.7.1.9	vector_16	54
4.7.1.10	vector_17	54
4.7.1.11	vector_18	54
4.7.1.12	vector_19	54
4.7.1.13	vector_2	54

4.7.1.14	vector_20	54
4.7.1.15	vector_21	54
4.7.1.16	vector_22	54
4.7.1.17	vector_23	54
4.7.1.18	vector_24	55
4.7.1.19	vector_25	55
4.7.1.20	vector_26	55
4.7.1.21	vector_27	55
4.7.1.22	vector_28	55
4.7.1.23	vector_29	55
4.7.1.24	vector_3	55
4.7.1.25	vector_30	55
4.7.1.26	vector_31	55
4.7.1.27	vector_32	55
4.7.1.28	vector_33	55
4.7.1.29	vector_34	55
4.7.1.30	vector_35	55
4.7.1.31	vector_36	55
4.7.1.32	vector_37	55
4.7.1.33	vector_38	55
4.7.1.34	vector_39	55
4.7.1.35	vector_4	55
4.7.1.36	vector_40	55
4.7.1.37	vector_41	55
4.7.1.38	vector_42	55
4.7.1.39	vector_43	55
4.7.1.40	vector_44	55
4.7.1.41	vector_45	55
4.7.1.42	vector_46	56
4.7.1.43	vector_47	56
4.7.1.44	vector_48	56
4.7.1.45	vector_49	56
4.7.1.46	vector_5	56
4.7.1.47	vector_50	56

4.7.1.48	vector_51	56
4.7.1.49	vector_52	56
4.7.1.50	vector_53	56
4.7.1.51	vector_54	56
4.7.1.52	vector_55	56
4.7.1.53	vector_56	56
4.7.1.54	vector_57	56
4.7.1.55	vector_58	56
4.7.1.56	vector_59	56
4.7.1.57	vector_6	56
4.7.1.58	vector_60	56
4.7.1.59	vector_61	56
4.7.1.60	vector_62	56
4.7.1.61	vector_63	56
4.7.1.62	vector_7	56
4.7.1.63	vector_8	56
4.7.1.64	vector_9	56
4.8	include/arch/x86/x86.h File Reference	57
4.9	include/cli.h File Reference	57
4.9.1	Define Documentation	58
4.9.1.1	MAXBUFSIZE	58
4.9.2	Function Documentation	58
4.9.2.1	console_getc	58
4.9.2.2	console_init	59
4.9.2.3	console_interrupt	59
4.9.2.4	console_putc	59
4.9.3	Variable Documentation	59
4.9.3.1	buf	59
4.9.3.2	rpos	59
4.9.3.3	wpos	59
4.10	include/cmos.h File Reference	60
4.10.1	Define Documentation	61
4.10.1.1	CMOS_DATAPORT	61
4.10.1.2	CMOS_DRIVE_C	61

4.10.1.3 CMOS_DRIVE_D . . . . .	61
4.10.1.4 CMOS_EQUIP_BYTE . . . . .	61
4.10.1.5 CMOS_EXTMEM_LSB . . . . .	61
4.10.1.6 CMOS_EXTMEM_MSB . . . . .	61
4.10.1.7 CMOS_FIXED_DISK . . . . .	61
4.10.1.8 CMOS_INDEXPORT . . . . .	61
4.10.1.9 CMOS_STATUS_10 . . . . .	61
4.10.1.10 CMOS_STATUS_C . . . . .	61
4.10.1.11 CMOS_STATUS_D . . . . .	61
4.10.1.12 CMOS_STATUS_E . . . . .	61
4.10.1.13 CMOS_STATUS_F . . . . .	61
4.10.1.14 CMOS_SYSBASE_LSB . . . . .	61
4.10.1.15 CMOS_SYSBASE_MSB . . . . .	62
4.10.1.16 RTC_ALRMHOUR . . . . .	62
4.10.1.17 RTC_ALRMMINUTE . . . . .	62
4.10.1.18 RTC_ALARMSECOND . . . . .	62
4.10.1.19 RTC_DAY_MONTH . . . . .	62
4.10.1.20 RTC_DAY_WEEK . . . . .	62
4.10.1.21 RTC_HOUR . . . . .	62
4.10.1.22 RTC_MINUTES . . . . .	62
4.10.1.23 RTC_MONTH . . . . .	62
4.10.1.24 RTC_SECONDS . . . . .	62
4.10.1.25 RTC_STATUS_A . . . . .	62
4.10.1.26 RTC_STATUS_B . . . . .	62
4.10.1.27 RTC_YEAR . . . . .	62
4.10.1.28 STAT_ALRM_INTR . . . . .	62
4.10.1.29 STAT_CAL_BCD . . . . .	62
4.10.1.30 STAT_CAL_BIN . . . . .	62
4.10.1.31 STAT_CAL_HR12 . . . . .	62
4.10.1.32 STAT_CAL_HR24 . . . . .	62
4.10.1.33 STAT_DAY_LGHT . . . . .	62
4.10.1.34 STAT_HALT . . . . .	62
4.10.1.35 STAT_PER_INTR . . . . .	62
4.10.1.36 STAT_RUN . . . . .	62

4.10.1.37 STAT_SQRWV_INTR . . . . .	62
4.10.1.38 STAT_UPDT_INTR . . . . .	63
4.11 include/cpuid.h File Reference . . . . .	63
4.11.1 Define Documentation . . . . .	63
4.11.1.1 CPUID_INFO . . . . .	63
4.11.1.2 CPUID_VERSION . . . . .	63
4.12 include/device.h File Reference . . . . .	63
4.13 include/fs/ext2fs.h File Reference . . . . .	64
4.13.1 Define Documentation . . . . .	66
4.13.1.1 EXT2_64FZ . . . . .	66
4.13.1.2 EXT2_ACL_DATA_INO . . . . .	66
4.13.1.3 EXT2_ACL_IDX_INO . . . . .	66
4.13.1.4 EXT2_AFS_INO . . . . .	66
4.13.1.5 EXT2_BAD_INO . . . . .	66
4.13.1.6 EXT2_BOOT_LOADER_INO . . . . .	66
4.13.1.7 EXT2_CLEAN . . . . .	66
4.13.1.8 EXT2_COMPRESSION . . . . .	66
4.13.1.9 EXT2_DIND_BLOCK . . . . .	66
4.13.1.10 EXT2_DIR_BIN_TREE . . . . .	66
4.13.1.11 EXT2_DIR_HASHI . . . . .	66
4.13.1.12 EXT2_DIR_TYPE_F . . . . .	66
4.13.1.13 EXT2_ERROR . . . . .	66
4.13.1.14 EXT2_FIRST_INO . . . . .	66
4.13.1.15 EXT2_FS_JOR_DEV . . . . .	67
4.13.1.16 EXT2_FS_REPLAY_JOR . . . . .	67
4.13.1.17 EXT2_FS_RESIZE . . . . .	67
4.13.1.18 EXT2_IGNORE . . . . .	67
4.13.1.19 EXT2_IND_BLOCK . . . . .	67
4.13.1.20 EXT2_INO_EXTNDATTR . . . . .	67
4.13.1.21 EXT2_INOSZ_V0 . . . . .	67
4.13.1.22 EXT2_JRNL . . . . .	67
4.13.1.23 EXT2_KPANIC . . . . .	67
4.13.1.24 EXT2_LINK_MAX . . . . .	67
4.13.1.25 EXT2_N_BLOCKS . . . . .	67

4.13.1.26 EXT2_NAME_LEN . . . . .	67
4.13.1.27 EXT2_NDIR_BLOCKS . . . . .	67
4.13.1.28 EXT2_OS_FBSD . . . . .	67
4.13.1.29 EXT2_OS_HURD . . . . .	67
4.13.1.30 EXT2_OS_LINUX . . . . .	67
4.13.1.31 EXT2_OS_LITES . . . . .	67
4.13.1.32 EXT2_OS_MASIX . . . . .	67
4.13.1.33 EXT2_PRE_02B_MAGIC . . . . .	67
4.13.1.34 EXT2_PRELOC . . . . .	67
4.13.1.35 EXT2_REMOUNT_RO . . . . .	67
4.13.1.36 EXT2_ROOT_INO . . . . .	67
4.13.1.37 EXT2_ROOT_INODE . . . . .	67
4.13.1.38 EXT2_SPARSE . . . . .	68
4.13.1.39 EXT2_SUPER_MAGIC . . . . .	68
4.13.1.40 EXT2_TIND_BLOCK . . . . .	68
4.13.1.41 EXT2_UNDEL_DIR_INO . . . . .	68
4.13.1.42 I_AFS_DIR . . . . .	68
4.13.1.43 I_APPEND_ONLY . . . . .	68
4.13.1.44 I_BLOCK_DEVICE . . . . .	68
4.13.1.45 I_CHAR_DEVICE . . . . .	68
4.13.1.46 I_DIR_BLOCK_DEV . . . . .	68
4.13.1.47 I_DIR_CHAR_DEV . . . . .	68
4.13.1.48 I_DIR_DIRECTORY . . . . .	68
4.13.1.49 I_DIR_FIFO . . . . .	68
4.13.1.50 I_DIR_REGULAR_F . . . . .	68
4.13.1.51 I_DIR SOCK . . . . .	68
4.13.1.52 I_DIR_SYM_LINK . . . . .	68
4.13.1.53 I_DIR_UNKOWN . . . . .	68
4.13.1.54 I_DIRECTORY . . . . .	68
4.13.1.55 I_FIFO . . . . .	68
4.13.1.56 I_FILE_COMPRESSION . . . . .	68
4.13.1.57 I_GEXEC_P . . . . .	68
4.13.1.58 I_GREAD_P . . . . .	68
4.13.1.59 I_GWRITE_P . . . . .	68

4.13.1.60 <code>I_HASH_INDEXED_DIR</code>	68
4.13.1.61 <code>I_IMMUTABLE_FILE</code>	69
4.13.1.62 <code>I_JOURNAL_FILE_DATA</code>	69
4.13.1.63 <code>I_KEEP_CPY</code>	69
4.13.1.64 <code>I_NO_DUMP</code>	69
4.13.1.65 <code>I_NO_LAT</code>	69
4.13.1.66 <code>I_OEXEC_P</code>	69
4.13.1.67 <code>I_OREAD_P</code>	69
4.13.1.68 <code>I_OWRITE_P</code>	69
4.13.1.69 <code>I_REG_FILE</code>	69
4.13.1.70 <code>I_SEC_DEL</code>	69
4.13.1.71 <code>I_SET_GID</code>	69
4.13.1.72 <code>I_SET_UID</code>	69
4.13.1.73 <code>I_STICKY</code>	69
4.13.1.74 <code>I_SYM_LINK</code>	69
4.13.1.75 <code>I_SYNC_UPDATES</code>	69
4.13.1.76 <code>I_UEXEC_P</code>	69
4.13.1.77 <code>I_UNIX SOCK</code>	69
4.13.1.78 <code>I_UREAD_P</code>	69
4.13.1.79 <code>I_UWRITE_P</code>	69
4.14 <code>include/init.h</code> File Reference	70
4.14.1 Define Documentation	71
4.14.1.1 <code>ELFHDR</code>	71
4.14.1.2 <code>SECTOR</code>	71
4.14.2 Function Documentation	71
4.14.2.1 <code>Init_userspace</code>	71
4.15 <code>include/kbc.h</code> File Reference	72
4.15.1 Define Documentation	74
4.15.1.1 <code>KBC_AUXWRITE</code>	74
4.15.1.2 <code>KBC_COMMAND</code>	74
4.15.1.3 <code>KBC_DATAIN</code>	74
4.15.1.4 <code>KBC_DATAPORT</code>	74
4.15.1.5 <code>KBC_DEFAULT</code>	74
4.15.1.6 <code>KBC_DIGDUMP</code>	74

4.15.1.7 KBC_DISA20 . . . . .	74
4.15.1.8 KBC_DISABLE . . . . .	74
4.15.1.9 KBC_DISKBD . . . . .	74
4.15.1.10 KBC_DISMOUSE . . . . .	74
4.15.1.11 KBC_ECHODIG . . . . .	74
4.15.1.12 KBC_ENABLE . . . . .	74
4.15.1.13 KBC_ENBA20 . . . . .	74
4.15.1.14 KBC_ENBKBD . . . . .	74
4.15.1.15 KBC_ENMOUSE . . . . .	74
4.15.1.16 KBC_FULLBUF . . . . .	74
4.15.1.17 KBC_INDICATOR . . . . .	74
4.15.1.18 KBC_INPREAD . . . . .	74
4.15.1.19 KBC_INTRTEST . . . . .	74
4.15.1.20 KBC_OUTAUX . . . . .	75
4.15.1.21 KBC_OUTKBD . . . . .	75
4.15.1.22 KBC_OUTREAD . . . . .	75
4.15.1.23 KBC_OUTWRTE . . . . .	75
4.15.1.24 KBC_PARITY . . . . .	75
4.15.1.25 KBC_PULSE0 . . . . .	75
4.15.1.26 KBC_PULSE1 . . . . .	75
4.15.1.27 KBC_PULSE2 . . . . .	75
4.15.1.28 KBC_PULSE3 . . . . .	75
4.15.1.29 KBC_READID . . . . .	75
4.15.1.30 KBC_READLOW . . . . .	75
4.15.1.31 KBC_READRAM . . . . .	75
4.15.1.32 KBC_READVER . . . . .	75
4.15.1.33 KBC_REBOOT . . . . .	75
4.15.1.34 KBC resend . . . . .	75
4.15.1.35 KBC_RESET . . . . .	75
4.15.1.36 KBC_RTIMOUT . . . . .	75
4.15.1.37 KBC_SECLOCK . . . . .	75
4.15.1.38 KBC_SELFTEST . . . . .	75
4.15.1.39 KBC_SETSCAN . . . . .	75
4.15.1.40 KBC_STATUSPORT . . . . .	75

4.15.1.41 KBC_TSTMOUSE . . . . .	76
4.15.1.42 KBC_TTIMEOUT . . . . .	76
4.15.1.43 KBC_TYPEMATIC . . . . .	76
4.15.1.44 KBC_WITERAM . . . . .	76
4.15.1.45 KEY_DEL . . . . .	76
4.15.1.46 KEY_DN . . . . .	76
4.15.1.47 KEY_END . . . . .	76
4.15.1.48 KEY_HOME . . . . .	76
4.15.1.49 KEY_INS . . . . .	76
4.15.1.50 KEY_LF . . . . .	76
4.15.1.51 KEY_PGDN . . . . .	76
4.15.1.52 KEY_PGUP . . . . .	76
4.15.1.53 KEY_RT . . . . .	76
4.15.1.54 KEY_UP . . . . .	76
4.15.2 Function Documentation . . . . .	76
4.15.2.1 kbc_data . . . . .	76
4.15.2.2 kbc_interrupt . . . . .	77
4.16 include/memvals.h File Reference . . . . .	77
4.16.1 Define Documentation . . . . .	78
4.16.1.1 EXTMEM . . . . .	78
4.16.1.2 GB . . . . .	78
4.16.1.3 IOPHYMEM . . . . .	78
4.16.1.4 KB . . . . .	79
4.16.1.5 KERNEL_ADDR . . . . .	79
4.16.1.6 KERNEL_STACK . . . . .	79
4.16.1.7 KERNEL_STACK_TOP . . . . .	79
4.16.1.8 MB . . . . .	79
4.16.1.9 PA . . . . .	79
4.16.1.10 PAGECNT . . . . .	79
4.16.1.11 PAGELG . . . . .	79
4.16.1.12 PAGESZ . . . . .	79
4.16.1.13 PAGETSZ . . . . .	79
4.16.1.14 PROC_LIST . . . . .	79
4.16.1.15 SEG_NULL . . . . .	79

4.16.1.16 SEGACS_D . . . . .	79
4.16.1.17 SEGACS_RW . . . . .	79
4.16.1.18 SEGACS_USR . . . . .	79
4.16.1.19 SEGACS_X . . . . .	79
4.16.1.20 SEGMENT . . . . .	80
4.16.1.21 USEREND . . . . .	80
4.16.1.22 USERPAGES . . . . .	80
4.16.1.23 USERSPACESTART . . . . .	80
4.16.1.24 USERSTACK_TOP . . . . .	80
4.16.1.25 USERSTART . . . . .	80
4.16.1.26 USERVIRTPGT . . . . .	80
4.16.1.27 VIRTPGT . . . . .	80
4.16.2 Function Documentation . . . . .	80
4.16.2.1 __attribute__ . . . . .	80
4.16.3 Variable Documentation . . . . .	80
4.16.3.1 base . . . . .	80
4.16.3.2 size . . . . .	80
4.17 include/proc/proc.h File Reference . . . . .	80
4.17.1 Define Documentation . . . . .	82
4.17.1.1 MAX_PROC_NAME . . . . .	82
4.17.1.2 MAX_PROCS . . . . .	82
4.17.1.3 NON_RUNNABLE . . . . .	82
4.17.1.4 PROC_EMPTY . . . . .	82
4.17.1.5 PROC_TABLE_SIZE . . . . .	82
4.17.1.6 RUNNABLE . . . . .	82
4.17.2 Typedef Documentation . . . . .	83
4.17.2.1 proc_t . . . . .	83
4.17.3 Function Documentation . . . . .	83
4.17.3.1 init_proc_table . . . . .	83
4.17.3.2 LIFO_HEAD . . . . .	83
4.17.3.3 LIST_HEAD . . . . .	83
4.17.3.4 proc_setup . . . . .	83
4.17.3.5 proc_setup_mem . . . . .	83
4.17.3.6 switch_address_space . . . . .	83

4.17.3.7	test_lifo	83
4.17.4	Variable Documentation	84
4.17.4.1	proc_table	84
4.17.4.2	running_procs	84
4.18	include/rdisk.h File Reference	84
4.18.1	Define Documentation	85
4.18.1.1	ELF_MAGIC2	85
4.18.1.2	SECTOR	85
4.18.2	Function Documentation	85
4.18.2.1	readsect	85
4.18.2.2	readseg	86
4.18.2.3	waitdisk	86
4.19	include/stdarg.h File Reference	86
4.19.1	Define Documentation	87
4.19.1.1	__va_size	87
4.19.1.2	va_arg	87
4.19.1.3	va_end	87
4.19.1.4	va_start	87
4.19.2	Typedef Documentation	87
4.19.2.1	va_list	87
4.20	include/stdio.h File Reference	87
4.20.1	Function Documentation	88
4.20.1.1	getchar	88
4.20.1.2	ksprintkn	89
4.20.1.3	kvprintk	89
4.20.1.4	printk	90
4.20.1.5	putchr	90
4.20.1.6	readline	91
4.20.1.7	vprintk	92
4.21	include/string.h File Reference	92
4.21.1	Function Documentation	93
4.21.1.1	memcmp	93
4.21.1.2	memcpy	93
4.21.1.3	memmove	93

4.21.1.4	memset . . . . .	93
4.21.1.5	strcat . . . . .	93
4.21.1.6	strchr . . . . .	93
4.21.1.7	strcmp . . . . .	93
4.21.1.8	strcpy . . . . .	94
4.21.1.9	strlcat . . . . .	94
4.21.1.10	strlcpy . . . . .	94
4.21.1.11	strlen . . . . .	94
4.21.1.12	strncat . . . . .	94
4.21.1.13	strnchr . . . . .	94
4.21.1.14	strncmp . . . . .	94
4.21.1.15	strncpy . . . . .	94
4.21.1.16	strnlen . . . . .	94
4.21.1.17	strrchr . . . . .	94
4.22	include/structs/linkedlist.h File Reference . . . . .	94
4.22.1	Define Documentation . . . . .	95
4.22.1.1	LIST_EMPTY . . . . .	95
4.22.1.2	LIST_ENTRY . . . . .	95
4.22.1.3	LIST_FIRST . . . . .	95
4.22.1.4	LIST_FOREACH . . . . .	95
4.22.1.5	LIST_HEAD . . . . .	95
4.22.1.6	LIST_HEAD_INIT . . . . .	95
4.22.1.7	LIST_INIT . . . . .	95
4.22.1.8	LIST_INSERT_AFTER . . . . .	96
4.22.1.9	LIST_INSERT_BEFORE . . . . .	96
4.22.1.10	LIST_INSERT_HEAD . . . . .	96
4.22.1.11	LIST_NEXT . . . . .	96
4.22.1.12	LIST_REMOVE . . . . .	96
4.23	include/structs/list.h File Reference . . . . .	97
4.24	include/structs/queue.h File Reference . . . . .	97
4.24.1	Define Documentation . . . . .	98
4.24.1.1	LIFO_ENTRY . . . . .	98
4.24.1.2	LIFO_FIRST . . . . .	98
4.24.1.3	LIFO_HEAD . . . . .	98

4.24.1.4	LIFO_HEAD_INIT . . . . .	98
4.24.1.5	LIFO_INIT . . . . .	98
4.24.1.6	LIFO_POP . . . . .	98
4.24.1.7	LIFO_PUSH . . . . .	98
4.25	include/test.h File Reference . . . . .	98
4.25.1	Define Documentation . . . . .	100
4.25.1.1	assert . . . . .	100
4.25.1.2	panic . . . . .	100
4.25.2	Function Documentation . . . . .	100
4.25.2.1	_panic_ . . . . .	100
4.26	include/types.h File Reference . . . . .	101
4.26.1	Define Documentation . . . . .	101
4.26.1.1	NULL . . . . .	101
4.26.1.2	ROUND_DOWN . . . . .	101
4.26.1.3	ROUND_UP . . . . .	102
4.26.2	Typedef Documentation . . . . .	102
4.26.2.1	int16_t . . . . .	102
4.26.2.2	int32_t . . . . .	102
4.26.2.3	int64_t . . . . .	102
4.26.2.4	int8_t . . . . .	102
4.26.2.5	intmax_t . . . . .	102
4.26.2.6	paddr_t . . . . .	102
4.26.2.7	reg_t . . . . .	102
4.26.2.8	size_t . . . . .	102
4.26.2.9	uint16_t . . . . .	102
4.26.2.10	uint32_t . . . . .	102
4.26.2.11	uint64_t . . . . .	102
4.26.2.12	uint8_t . . . . .	102
4.26.2.13	uintmax_t . . . . .	102
4.26.2.14	vaddr_t . . . . .	102
4.27	include/video.h File Reference . . . . .	103
4.27.1	Define Documentation . . . . .	104
4.27.1.1	BACKGROUND_BLINK . . . . .	104
4.27.1.2	BACKGROUND_BLUE . . . . .	104

4.27.1.3 BACKGROUND_GRAY . . . . .	104
4.27.1.4 BACKGROUND_GREEN . . . . .	104
4.27.1.5 BACKGROUND_RED . . . . .	104
4.27.1.6 BACKGROUND_WHITE . . . . .	104
4.27.1.7 CGA_BUFF_OFF . . . . .	104
4.27.1.8 CGA_COLS . . . . .	104
4.27.1.9 CGA_DATA1 . . . . .	104
4.27.1.10 CGA_INDEX1 . . . . .	104
4.27.1.11 CGA_ROWS . . . . .	104
4.27.1.12 CGA_SIZE . . . . .	104
4.27.1.13 COLOR_BLUE . . . . .	104
4.27.1.14 COLOR_DARK_GRAY . . . . .	104
4.27.1.15 COLOR_GRAY . . . . .	104
4.27.1.16 COLOR_GREEN . . . . .	104
4.27.1.17 COLOR_RED . . . . .	104
4.27.1.18 COLOR_WHITE . . . . .	104
4.27.2 Function Documentation . . . . .	104
4.27.2.1 cga_get_pos . . . . .	104
4.27.2.2 cga_init . . . . .	105
4.27.2.3 cga_putc . . . . .	105
4.27.2.4 cga_set_attr . . . . .	105
4.27.2.5 cga_set_pos . . . . .	105
4.27.3 Variable Documentation . . . . .	105
4.27.3.1 char_buff . . . . .	105
4.27.3.2 cursor_position . . . . .	105
4.28 kernel/arch/x86/interrupt.c File Reference . . . . .	106
4.28.1 Function Documentation . . . . .	106
4.28.1.1 idt_init . . . . .	107
4.28.1.2 interrupt_init . . . . .	107
4.28.1.3 map_exception . . . . .	107
4.28.1.4 page_fault_handler . . . . .	108
4.28.1.5 register_exception . . . . .	108
4.28.1.6 trap . . . . .	108
4.28.2 Variable Documentation . . . . .	108

4.28.2.1	idt	109
4.28.2.2	int_generic	109
4.28.2.3	x86_exception_names	109
4.29	kernel/arch/x86/mm/init_mem.c File Reference	109
4.29.1	Function Documentation	110
4.29.1.1	init_tss	110
4.29.1.2	scan_memory	111
4.29.1.3	x86_setup_memory	111
4.29.2	Variable Documentation	111
4.29.2.1	catgdt	111
4.29.2.2	gdtdesc	112
4.29.2.3	idtdesc	112
4.29.2.4	max_addr	112
4.29.2.5	next_free	112
4.29.2.6	pages	112
4.29.2.7	proc_table	112
4.29.2.8	tss	112
4.30	kernel/arch/x86/mm/page.c File Reference	112
4.30.1	Function Documentation	114
4.30.1.1	map_segment_page	114
4.30.1.2	x86_page_alloc	114
4.30.1.3	x86_page_detach	114
4.30.1.4	x86_page_free	115
4.30.1.5	x86_page_init	115
4.30.1.6	x86_page_insert	115
4.30.1.7	x86_page_lookup	115
4.30.1.8	x86_page_remove	116
4.30.1.9	x86_paging_init	116
4.30.1.10	x86_pgdir_find	116
4.30.1.11	x86_test_pgdir	116
4.30.2	Variable Documentation	117
4.30.2.1	global_cr3	117
4.30.2.2	global_pgdir	117
4.30.2.3	next_free	117

4.30.2.4	page_count . . . . .	117
4.30.2.5	pages . . . . .	117
4.31	kernel/cli.c File Reference . . . . .	117
4.31.1	Function Documentation . . . . .	118
4.31.1.1	console_clear . . . . .	118
4.31.1.2	console_getc . . . . .	118
4.31.1.3	console_init . . . . .	119
4.31.1.4	console_interrupt . . . . .	119
4.31.1.5	console_putc . . . . .	119
4.31.1.6	getchar . . . . .	120
4.31.1.7	putchr . . . . .	120
4.32	kernel/cmos.c File Reference . . . . .	121
4.32.1	Function Documentation . . . . .	121
4.32.1.1	cmos_get_reg . . . . .	121
4.32.1.2	cmos_get_time . . . . .	121
4.32.1.3	cmos_set_power_stat . . . . .	121
4.33	kernel/cpuid.c File Reference . . . . .	122
4.33.1	Function Documentation . . . . .	122
4.33.1.1	cpuid_get_eax . . . . .	122
4.33.1.2	cpuid_get_ebx . . . . .	122
4.33.1.3	cpuid_get_ecx . . . . .	122
4.33.1.4	cpuid_get_edx . . . . .	123
4.33.1.5	cpuid_print . . . . .	123
4.33.1.6	set_eax . . . . .	123
4.34	kernel/init.c File Reference . . . . .	123
4.34.1	Function Documentation . . . . .	124
4.34.1.1	Init_userspace . . . . .	124
4.34.2	Variable Documentation . . . . .	124
4.34.2.1	catgdt . . . . .	124
4.34.2.2	global_pgdir . . . . .	124
4.34.2.3	processes . . . . .	124
4.35	kernel/kbc.c File Reference . . . . .	125
4.35.1	Define Documentation . . . . .	125
4.35.1.1	ALT . . . . .	125

4.35.1.2	CAPSLOCK	126
4.35.1.3	CL	126
4.35.1.4	CTL	126
4.35.1.5	ESCODE	126
4.35.1.6	NUL	126
4.35.1.7	NUMLOCK	126
4.35.1.8	SCROLLLOCK	126
4.35.1.9	SHIFT	126
4.35.2	Function Documentation	126
4.35.2.1	kbc_data	126
4.35.2.2	kbc_interrupt	126
4.36	kernel/play.c File Reference	126
4.36.1	Detailed Description	127
4.36.2	Function Documentation	128
4.36.2.1	_panic_	128
4.36.2.2	bootup	128
4.36.2.3	play	129
4.36.2.4	time_print	129
4.37	kernel/printf.c File Reference	130
4.37.1	Define Documentation	130
4.37.1.1	hex2ascii	130
4.37.2	Function Documentation	130
4.37.2.1	getint	131
4.37.2.2	getuint	131
4.37.2.3	ksprintkn	131
4.37.2.4	kvprintk	131
4.37.2.5	printk	132
4.37.2.6	vprintk	133
4.38	kernel/proc/load_elf.c File Reference	133
4.38.1	Function Documentation	134
4.38.1.1	elf_load_to_proc	134
4.39	kernel/proc/proc.c File Reference	134
4.39.1	Function Documentation	135
4.39.1.1	create_proc	136

4.39.1.2	init_proc . . . . .	136
4.39.1.3	init_proc0 . . . . .	136
4.39.1.4	init_proc_table . . . . .	137
4.39.1.5	proc_alloc_mem . . . . .	137
4.39.1.6	switch_address_space . . . . .	137
4.39.1.7	test_lifo . . . . .	137
4.39.2	Variable Documentation . . . . .	138
4.39.2.1	empty_procs . . . . .	138
4.39.2.2	global_pgdir . . . . .	138
4.39.2.3	proc_table . . . . .	138
4.39.2.4	running_procs . . . . .	138
4.40	kernel/rdisk.c File Reference . . . . .	138
4.40.1	Function Documentation . . . . .	139
4.40.1.1	readsect . . . . .	139
4.40.1.2	readseg . . . . .	139
4.40.1.3	waitdisk . . . . .	139
4.41	kernel/readline.c File Reference . . . . .	140
4.41.1	Define Documentation . . . . .	140
4.41.1.1	BUFSIZE . . . . .	140
4.41.2	Function Documentation . . . . .	140
4.41.2.1	readline . . . . .	141
4.42	kernel/string.c File Reference . . . . .	142
4.42.1	Function Documentation . . . . .	142
4.42.1.1	memcpy . . . . .	142
4.42.1.2	memset . . . . .	142
4.42.1.3	strnlen . . . . .	142
4.43	kernel/tmp/init_elf.c File Reference . . . . .	142
4.43.1	Function Documentation . . . . .	143
4.43.1.1	main . . . . .	143
4.44	kernel/video.c File Reference . . . . .	143
4.44.1	Function Documentation . . . . .	143
4.44.1.1	cga_clear . . . . .	144
4.44.1.2	cga_get_pos . . . . .	144
4.44.1.3	cga_init . . . . .	144

---

4.44.1.4	cga_putc . . . . .	145
4.44.1.5	cga_putstr . . . . .	145
4.44.1.6	cga_set_attr . . . . .	145
4.44.1.7	cga_set_pos . . . . .	145
4.45	kernel/work_it_out.c File Reference . . . . .	145
4.45.1	Function Documentation . . . . .	146
4.45.1.1	work_it_out . . . . .	147



# Chapter 1

## Class Index

### 1.1 Class List

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

<a href="#">__attribute__</a>	5
<a href="#">cpu_state_t</a>	10
<a href="#">cpuid_regs</a>	11
<a href="#">element</a>	12
<a href="#">elfhdr</a>	12
<a href="#">ext2_dir_entry</a>	13
<a href="#">ext2_group_desc</a>	14
<a href="#">ext2_inode</a>	15
<a href="#">ext2_super_block</a>	17
<a href="#">Gtdesc</a>	20
<a href="#">gpr_regs_t</a>	21
<a href="#">Page</a>	22
<a href="#">proc</a>	22
<a href="#">proghdr</a>	24
<a href="#">sechdr</a>	25
<a href="#">seg_REGS_t</a>	26
<a href="#">Segdesc</a>	27



# Chapter 2

## File Index

### 2.1 File List

Here is a list of all files with brief descriptions:

include/cli.h	57
include/cmos.h	60
include/cpuid.h	63
include/device.h	63
include/init.h	70
include/kbc.h	72
include/memvals.h	77
include/rdisk.h	84
include/stdarg.h	86
include/stdio.h	87
include/string.h	92
include/test.h	98
include/types.h	101
include/video.h	103
include/arch/x86/cpu_state.h	29
include/arch/x86/elf.h	31
include/arch/x86/interrupt.h	35
include/arch/x86/processor.h	49
include/arch/x86/vectors.h	52
include/arch/x86/x86.h	57
include/arch/x86/mm/page.h	40
include/arch/x86/mm/segdesc.h	48
include/fs/ext2fs.h	64
include/proc/proc.h	80
include/structs/linkedlist.h	94
include/structs/list.h	97
include/structs/queue.h	97
kernel/cli.c	117
kernel/cmos.c	121

kernel/cpuid.c . . . . .	122
kernel/init.c . . . . .	123
kernel/kbc.c . . . . .	125
kernel/play.c . . . . .	
Kernel main file . . . . .	126
kernel/printf.c . . . . .	130
kernel/rdisk.c . . . . .	138
kernel/readline.c . . . . .	140
kernel/string.c . . . . .	142
kernel/video.c . . . . .	143
kernel/work_it_out.c . . . . .	145
kernel/arch/x86/interrupt.c . . . . .	106
kernel/arch/x86/mm/init_mem.c . . . . .	109
kernel/arch/x86/mm/page.c . . . . .	112
kernel/proc/load_elf.c . . . . .	133
kernel/proc/proc.c . . . . .	134
kernel/tmp/init_elf.c . . . . .	142

# Chapter 3

## Class Documentation

### 3.1 \_\_attribute\_\_ Struct Reference

```
#include <cpu_state.h>
```

#### Public Attributes

- `uint16_t prelink`
- `uint16_t _rsrvd1`
- `reg_t esp0`
- `uint16_t ss0`
- `uint16_t _rsrvd2`
- `reg_t esp1`
- `uint16_t ss1`
- `uint16_t _rsrvd3`
- `reg_t esp2`
- `uint16_t ss2`
- `uint16_t _rsrvd4`
- `uint32_t cr3`
- `reg_t eip`
- `reg_t eflags`
- `reg_t eax`
- `reg_t ecx`
- `reg_t edx`
- `reg_t ebx`
- `reg_t esp`
- `reg_t ebp`
- `reg_t esi`
- `reg_t edi`
- `uint16_t es`
- `uint16_t _rsrvd5`

- `uint16_t cs`
- `uint16_t _rsrvd6`
- `uint16_t ss`
- `uint16_t _rsrvd7`
- `uint16_t ds`
- `uint16_t _rsrvd8`
- `uint16_t fs`
- `uint16_t _rsrvd9`
- `uint16_t gs`
- `uint16_t _rsrvda`
- `uint16_t ldt`
- `uint16_t _rsrvdb`
- `uint16_t trace`
- `uint16_t iomap_base`
- `uint16_t limit_0_15`
- `uint16_t base_0_15`
- `uint8_t base_16_23`
- `uint8_t access`
- `unsigned limit_16_19:4`
- `unsigned available:1`
- `unsigned unused:2`
- `unsigned granularity:1`
- `uint8_t base_24_31`
- `uint16_t offset_0_15`
- `uint16_t segment_s`
- `unsigned args:5`
- `unsigned reserved:3`
- `unsigned type:4`
- `unsigned s:1`
- `unsigned dpl:2`
- `unsigned p:1`
- `uint16_t offset_16_31`
- `unsigned present:1`
- `unsigned writable:1`
- `unsigned accessible:1`
- `unsigned write_through:1`
- `unsigned cache_disable:1`
- `unsigned accessed:1`
- `unsigned dirty:1`
- `unsigned pat:1`
- `unsigned global:1`
- `unsigned ignore_this:3`
- `unsigned address:20`

### 3.1.1 Member Data Documentation

- 3.1.1.1 `uint16_t __attribute__::_rsrvd1`
- 3.1.1.2 `uint16_t __attribute__::_rsrvd2`
- 3.1.1.3 `uint16_t __attribute__::_rsrvd3`
- 3.1.1.4 `uint16_t __attribute__::_rsrvd4`
- 3.1.1.5 `uint16_t __attribute__::_rsrvd5`
- 3.1.1.6 `uint16_t __attribute__::_rsrvd6`
- 3.1.1.7 `uint16_t __attribute__::_rsrvd7`
- 3.1.1.8 `uint16_t __attribute__::_rsrvd8`
- 3.1.1.9 `uint16_t __attribute__::_rsrvd9`
- 3.1.1.10 `uint16_t __attribute__::_rsrvda`
- 3.1.1.11 `uint16_t __attribute__::_rsrvdb`
- 3.1.1.12 `uint8_t __attribute__::access`
- 3.1.1.13 `unsigned __attribute__::accessed`
- 3.1.1.14 `unsigned __attribute__::accessible`
- 3.1.1.15 `unsigned __attribute__::address`
- 3.1.1.16 `unsigned __attribute__::args`
- 3.1.1.17 `unsigned __attribute__::available`
- 3.1.1.18 `uint16_t __attribute__::base_0_15`
- 3.1.1.19 `uint8_t __attribute__::base_16_23`
- 3.1.1.20 `uint8_t __attribute__::base_24_31`
- 3.1.1.21 `unsigned __attribute__::cache_disable`
- 3.1.1.22 `uint32_t __attribute__::cr3`
- 3.1.1.23 `uint16_t __attribute__::cs`

- 3.1.1.24 `unsigned __attribute__::dirty`
- 3.1.1.25 `unsigned __attribute__::dpl`
- 3.1.1.26 `uint16_t __attribute__::ds`
- 3.1.1.27 `reg_t __attribute__::eax`
- 3.1.1.28 `reg_t __attribute__::ebp`
- 3.1.1.29 `reg_t __attribute__::ebx`
- 3.1.1.30 `reg_t __attribute__::ecx`
- 3.1.1.31 `reg_t __attribute__::edi`
- 3.1.1.32 `reg_t __attribute__::edx`
- 3.1.1.33 `reg_t __attribute__::eflags`
- 3.1.1.34 `reg_t __attribute__::eip`
- 3.1.1.35 `uint16_t __attribute__::es`
- 3.1.1.36 `reg_t __attribute__::esi`
- 3.1.1.37 `reg_t __attribute__::esp`
- 3.1.1.38 `reg_t __attribute__::esp0`
- 3.1.1.39 `reg_t __attribute__::esp1`
- 3.1.1.40 `reg_t __attribute__::esp2`
- 3.1.1.41 `uint16_t __attribute__::fs`
- 3.1.1.42 `unsigned __attribute__::global`
- 3.1.1.43 `unsigned __attribute__::granularity`
- 3.1.1.44 `uint16_t __attribute__::gs`
- 3.1.1.45 `unsigned __attribute__::ignore_this`
- 3.1.1.46 `uint16_t __attribute__::iomap_base`
- 3.1.1.47 `uint16_t __attribute__::ldt`

- 3.1.1.48 `uint16_t __attribute__::limit_0_15`
- 3.1.1.49 `unsigned __attribute__::limit_16_19`
- 3.1.1.50 `uint16_t __attribute__::offset_0_15`
- 3.1.1.51 `uint16_t __attribute__::offset_16_31`
- 3.1.1.52 `unsigned __attribute__::p`
- 3.1.1.53 `unsigned __attribute__::pat`
- 3.1.1.54 `uint16_t __attribute__::prelink`
- 3.1.1.55 `unsigned __attribute__::present`
- 3.1.1.56 `unsigned __attribute__::reserved`
- 3.1.1.57 `unsigned __attribute__::s`
- 3.1.1.58 `uint16_t __attribute__::segment_s`
- 3.1.1.59 `uint16_t __attribute__::ss`
- 3.1.1.60 `uint16_t __attribute__::ss0`
- 3.1.1.61 `uint16_t __attribute__::ss1`
- 3.1.1.62 `uint16_t __attribute__::ss2`
- 3.1.1.63 `uint16_t __attribute__::trace`
- 3.1.1.64 `unsigned __attribute__::type`
- 3.1.1.65 `unsigned __attribute__::unused`
- 3.1.1.66 `unsigned __attribute__::writable`
- 3.1.1.67 `unsigned __attribute__::write_through`

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

- [include/arch/x86/cpu\\_state.h](#)
- [include/arch/x86/interrupt.h](#)
- [include/arch/x86/mm/page.h](#)

## 3.2 `cpu_state_t` Struct Reference

```
#include <cpu_state.h>
```

### Public Attributes

- `reg_t ebp_frame`
- `reg_t eip_frame`
- `reg_t gs`
- `reg_t fs`
- `reg_t es`
- `reg_t ds`
- `reg_t edi`
- `reg_t esi`
- `reg_t o_ebp`
- `reg_t o_esp`
- `reg_t ebx`
- `reg_t edx`
- `reg_t ecx`
- `reg_t eax`
- `uint32_t error_code`
- `reg_t eip`
- `reg_t cs`
- `reg_t eflags`
- `reg_t esp`
- `reg_t ss`

### 3.2.1 Detailed Description

[include/arch/x86/cpu\\_state.h](http://catreloaded.net/include/arch/x86/cpu_state.h) CATReloaded (C) Copyrights 2011 <http://catreloaded.net>

#### Date

28 Sept, 2012

### 3.2.2 Member Data Documentation

#### 3.2.2.1 `reg_t cpu_state_t::cs`

#### 3.2.2.2 `reg_t cpu_state_t::ds`

#### 3.2.2.3 `reg_t cpu_state_t::eax`

#### 3.2.2.4 `reg_t cpu_state_t::ebp_frame`

- 3.2.2.5 `reg_t cpu_state_t::ebx`
- 3.2.2.6 `reg_t cpu_state_t::ecx`
- 3.2.2.7 `reg_t cpu_state_t::edi`
- 3.2.2.8 `reg_t cpu_state_t::edx`
- 3.2.2.9 `reg_t cpu_state_t::eflags`
- 3.2.2.10 `reg_t cpu_state_t::eip`
- 3.2.2.11 `reg_t cpu_state_t::eip_frame`
- 3.2.2.12 `uint32_t cpu_state_t::error_code`
- 3.2.2.13 `reg_t cpu_state_t::es`
- 3.2.2.14 `reg_t cpu_state_t::esi`
- 3.2.2.15 `reg_t cpu_state_t::esp`
- 3.2.2.16 `reg_t cpu_state_t::fs`
- 3.2.2.17 `reg_t cpu_state_t::gs`
- 3.2.2.18 `reg_t cpu_state_t::o_ebp`
- 3.2.2.19 `reg_t cpu_state_t::o_esp`
- 3.2.2.20 `reg_t cpu_state_t::ss`

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

- `include/arch/x86/cpu_state.h`

### 3.3 cpuid\_regs Struct Reference

#### Public Attributes

- `uint32_t eax`
- `uint32_t ebx`
- `uint32_t ecx`
- `uint32_t edx`

### 3.3.1 Member Data Documentation

3.3.1.1 `uint32_t cpuid_regs::eax`

3.3.1.2 `uint32_t cpuid_regs::ebx`

3.3.1.3 `uint32_t cpuid_regs::ecx`

3.3.1.4 `uint32_t cpuid_regs::edx`

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

- kernel/[cpuid.c](#)

### 3.4 element Struct Reference

```
#include <list.h>
```

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

- include/structs/[list.h](#)

### 3.5 elfhdr Struct Reference

```
#include <elf.h>
```

#### Public Attributes

- `uint32_t magic`
- `uint8_t magic2 [MAGIC_LEN-4]`
- `uint16_t type`
- `uint16_t machine`
- `uint32_t version`
- `uint32_t entry`
- `uint32_t phroff`
- `uint32_t shroff`
- `uint32_t flags`
- `uint16_t ehsizze`
- `uint16_t phrsize`
- `uint16_t phrnum`
- `uint16_t shrsize`
- `uint16_t shrnum`
- `uint16_t shrstrtbl`

### 3.5.1 Member Data Documentation

- 3.5.1.1 `uint16_t elfhdr::ehsize`
- 3.5.1.2 `uint32_t elfhdr::entry`
- 3.5.1.3 `uint32_t elfhdr::flags`
- 3.5.1.4 `uint16_t elfhdr::machine`
- 3.5.1.5 `uint32_t elfhdr::magic`
- 3.5.1.6 `uint8_t elfhdr::magic2[MAGIC_LEN-4]`
- 3.5.1.7 `uint16_t elfhdr::phrnum`
- 3.5.1.8 `uint32_t elfhdr::phroff`
- 3.5.1.9 `uint16_t elfhdr::phrsizze`
- 3.5.1.10 `uint16_t elfhdr::shrnum`
- 3.5.1.11 `uint32_t elfhdr::shroff`
- 3.5.1.12 `uint16_t elfhdr::shrsizze`
- 3.5.1.13 `uint16_t elfhdr::shrsttbl`
- 3.5.1.14 `uint16_t elfhdr::type`
- 3.5.1.15 `uint32_t elfhdr::version`

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

- include/arch/x86/elf.h

## 3.6 ext2\_dir\_entry Struct Reference

```
#include <ext2fs.h>
```

### Public Attributes

- `uint32_t inode`
- `uint16_t rec_len`
- `uint8_t name_len`
- `uint8_t type int8_t name [EXT2_NAME_LEN]`

### 3.6.1 Member Data Documentation

- 3.6.1.1 `uint32_t ext2_dir_entry::inode`
- 3.6.1.2 `uint8_t type int8_t ext2_dir_entry::name[EXT2_NAME_LEN]`
- 3.6.1.3 `uint8_t ext2_dir_entry::name_len`
- 3.6.1.4 `uint16_t ext2_dir_entry::rec_len`

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

- include/fs/[ext2fs.h](#)

## 3.7 ext2\_group\_desc Struct Reference

```
#include <ext2fs.h>
```

### Public Attributes

- `uint32_t bg_block_bitmap`
- `uint32_t bg_inode_bitmap`
- `uint32_t bg_inode_table`
- `uint16_t bg_free_blocks_count`
- `uint16_t bg_free_inodes_count`
- `uint16_t bg_used_dirs_count`
- `uint16_t bg_pad`
- `uint32_t bg_reserved [3]`

### 3.7.1 Member Data Documentation

- 3.7.1.1 `uint32_t ext2_group_desc::bg_block_bitmap`
- 3.7.1.2 `uint16_t ext2_group_desc::bg_free_blocks_count`
- 3.7.1.3 `uint16_t ext2_group_desc::bg_free_inodes_count`
- 3.7.1.4 `uint32_t ext2_group_desc::bg_inode_bitmap`
- 3.7.1.5 `uint32_t ext2_group_desc::bg_inode_table`
- 3.7.1.6 `uint16_t ext2_group_desc::bg_pad`
- 3.7.1.7 `uint32_t ext2_group_desc::bg_reserved[3]`

## 3.7.1.8 uint16\_t ext2\_group\_desc::bg\_used\_dirs\_count

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

- include/fs/ext2fs.h

## 3.8 ext2\_inode Struct Reference

```
#include <ext2fs.h>
```

### Public Attributes

- uint16\_t i\_mode
- uint16\_t i\_uid
- uint32\_t i\_size
- uint32\_t i\_atime
- uint32\_t i\_ctime
- uint32\_t i\_mtime
- uint32\_t i\_dtime
- uint16\_t i\_gid
- uint16\_t i\_links\_count
- uint32\_t i\_blocks
- uint32\_t i\_flags
- uint32\_t i\_reserved1
- uint32\_t i\_block [EXT2\_N\_BLOCKS]
- uint32\_t i\_version
- uint32\_t i\_file\_acl
- uint32\_t i\_dir\_acl
- uint32\_t i\_faddr
- union {
  - struct {
    - uint8\_t l\_i\_frag
    - uint8\_t l\_i\_fsize
    - uint16\_t l\_i\_reserved1
    - uint16\_t l\_i\_uid\_high
    - uint16\_t l\_i\_gid\_high
    - uint32\_t l\_i\_reserved2
  - } linux
  - struct {
    - uint8\_t h\_i\_frag
    - uint8\_t h\_i\_fsize
    - uint16\_t h\_i\_mode\_high
    - uint16\_t h\_i\_uid\_high
    - uint16\_t h\_i\_gid\_high
    - uint32\_t h\_i\_author
  - } hurd2

```
struct {
    uint8_t m_i_frag
    uint8_t m_i_fsize
    uint16_t m_pad1
    uint32_t m_i_reserved2 [2]
} masix2
} osd2
```

### 3.8.1 Member Data Documentation

- 3.8.1.1 `uint32_t ext2_inode::h_i_author`
- 3.8.1.2 `uint8_t ext2_inode::h_i_frag`
- 3.8.1.3 `uint8_t ext2_inode::h_i_fsize`
- 3.8.1.4 `uint16_t ext2_inode::h_i_gid_high`
- 3.8.1.5 `uint16_t ext2_inode::h_i_mode_high`
- 3.8.1.6 `uint16_t ext2_inode::h_i_uid_high`
- 3.8.1.7 `struct { ... } ext2_inode::hurd2`
- 3.8.1.8 `uint32_t ext2_inode::i_atime`
- 3.8.1.9 `uint32_t ext2_inode::i_block[EXT2_N_BLOCKS]`
- 3.8.1.10 `uint32_t ext2_inode::i_blocks`
- 3.8.1.11 `uint32_t ext2_inode::i_ctime`
- 3.8.1.12 `uint32_t ext2_inode::i_dir_acl`
- 3.8.1.13 `uint32_t ext2_inode::i_dtime`
- 3.8.1.14 `uint32_t ext2_inode::i_faddr`
- 3.8.1.15 `uint32_t ext2_inode::i_file_acl`
- 3.8.1.16 `uint32_t ext2_inode::i_flags`
- 3.8.1.17 `uint16_t ext2_inode::i_gid`
- 3.8.1.18 `uint16_t ext2_inode::i_links_count`

```
3.8.1.19 uint16_t ext2_inode::i_mode
3.8.1.20 uint32_t ext2_inode::i_mtime
3.8.1.21 uint32_t ext2_inode::i_reserved1
3.8.1.22 uint32_t ext2_inode::i_size
3.8.1.23 uint16_t ext2_inode::i_uid
3.8.1.24 uint32_t ext2_inode::i_version
3.8.1.25 uint8_t ext2_inode::l_i_frag
3.8.1.26 uint8_t ext2_inode::l_i_fsize
3.8.1.27 uint16_t ext2_inode::l_i_gid_high
3.8.1.28 uint16_t ext2_inode::l_i_reserved1
3.8.1.29 uint32_t ext2_inode::l_i_reserved2
3.8.1.30 uint16_t ext2_inode::l_i_uid_high
3.8.1.31 struct { ... } ext2_inode::linux
3.8.1.32 uint8_t ext2_inode::m_i_frag
3.8.1.33 uint8_t ext2_inode::m_i_fsize
3.8.1.34 uint32_t ext2_inode::m_i_reserved2[2]
3.8.1.35 uint16_t ext2_inode::m_pad1
3.8.1.36 struct { ... } ext2_inode::masix2
3.8.1.37 union { ... } ext2_inode::osd2
```

OS SPECIFIC VALUES from minix inode.h

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

- include/fs/ext2fs.h

## 3.9 ext2\_super\_block Struct Reference

```
#include <ext2fs.h>
```

## Public Attributes

- `uint32_t s_inodes_count`
- `uint32_t s_blocks_count`
- `uint32_t s_r_blocks_count`
- `uint32_t s_free_blocks_count`
- `uint32_t s_free_inodes_count`
- `uint32_t s_first_data_block`
- `uint32_t s_log_block_size`
- `uint32_t s_log_frag_size`
- `uint32_t s_blocks_per_group`
- `uint32_t s_frags_per_group`
- `uint32_t s_inodes_per_group`
- `uint32_t s_mtime`
- `uint32_t s_wtime`
- `uint16_t s_mnt_count`
- `uint16_t s_max_mnt_count`
- `uint16_t s_magic`
- `uint16_t s_state`
- `uint16_t s_errors`
- `uint16_t s_min_pad`
- `uint32_t s_lastcheck`
- `uint32_t s_checkinterval`
- `uint32_t s_os_id`
- `uint32_t s_maj_pad`
- `uint16_t s_uid`
- `uint16_t s_gid`
- `uint32_t s_first_ino`
- `uint16_t s_inode_size`
- `uint16_t s_block_group_nr`
- `uint32_t s_feature_compat`
- `uint32_t s_feature_incompat`
- `uint32_t s_feature_ro_compat`
- `uint8_t s_uuid [16]`
- `char s_volume_name [16]`
- `char s_last_mounted [64]`
- `uint32_t s_algorithm_usage_bitmap`
- `uint8_t s_prealloc_blocks`
- `uint8_t s_prealloc_dir_blocks`
- `uint16_t s_padding1`
- `uint8_t s_journal_uuid [16]`
- `uint32_t s_journal_inum`
- `uint32_t s_journal_dev`
- `uint32_t s_last_orphan`
- `uint32_t s_reserved [197]`

### 3.9.1 Member Data Documentation

- 3.9.1.1 `uint32_t ext2_super_block::s_algorithm_usage_bitmap`
- 3.9.1.2 `uint16_t ext2_super_block::s_block_group_nr`
- 3.9.1.3 `uint32_t ext2_super_block::s_blocks_count`
- 3.9.1.4 `uint32_t ext2_super_block::s_blocks_per_group`
- 3.9.1.5 `uint32_t ext2_super_block::s_checkinterval`
- 3.9.1.6 `uint16_t ext2_super_block::s_errors`
- 3.9.1.7 `uint32_t ext2_super_block::s_feature_compat`
- 3.9.1.8 `uint32_t ext2_super_block::s_feature_incompat`
- 3.9.1.9 `uint32_t ext2_super_block::s_feature_ro_compat`
- 3.9.1.10 `uint32_t ext2_super_block::s_first_data_block`
- 3.9.1.11 `uint32_t ext2_super_block::s_first_ino`
- 3.9.1.12 `uint32_t ext2_super_block::s_frags_per_group`
- 3.9.1.13 `uint32_t ext2_super_block::s_free_blocks_count`
- 3.9.1.14 `uint32_t ext2_super_block::s_free_inodes_count`
- 3.9.1.15 `uint16_t ext2_super_block::s_gid`
- 3.9.1.16 `uint16_t ext2_super_block::s_inode_size`
- 3.9.1.17 `uint32_t ext2_super_block::s_inodes_count`
- 3.9.1.18 `uint32_t ext2_super_block::s_inodes_per_group`
- 3.9.1.19 `uint32_t ext2_super_block::s_journal_dev`
- 3.9.1.20 `uint32_t ext2_super_block::s_journal_inum`
- 3.9.1.21 `uint8_t ext2_super_block::s_journal_uuid[16]`
- 3.9.1.22 `char ext2_super_block::s_last_mounted[64]`
- 3.9.1.23 `uint32_t ext2_super_block::s_last_orphan`

- 3.9.1.24 `uint32_t ext2_super_block::s_lastcheck`
- 3.9.1.25 `uint32_t ext2_super_block::s_log_block_size`
- 3.9.1.26 `uint32_t ext2_super_block::s_log_frag_size`
- 3.9.1.27 `uint16_t ext2_super_block::s_magic`
- 3.9.1.28 `uint32_t ext2_super_block::s_maj_pad`
- 3.9.1.29 `uint16_t ext2_super_block::s_max_mnt_count`
- 3.9.1.30 `uint16_t ext2_super_block::s_min_pad`
- 3.9.1.31 `uint16_t ext2_super_block::s_mnt_count`
- 3.9.1.32 `uint32_t ext2_super_block::s_mtime`
- 3.9.1.33 `uint32_t ext2_super_block::s_os_id`
- 3.9.1.34 `uint16_t ext2_super_block::s_padding1`
- 3.9.1.35 `uint8_t ext2_super_block::s_preadloc_blocks`
- 3.9.1.36 `uint8_t ext2_super_block::s_preadloc_dir_blocks`
- 3.9.1.37 `uint32_t ext2_super_block::s_r_blocks_count`
- 3.9.1.38 `uint32_t ext2_super_block::s_reserved[197]`
- 3.9.1.39 `uint16_t ext2_super_block::s_state`
- 3.9.1.40 `uint16_t ext2_super_block::s_uid`
- 3.9.1.41 `uint8_t ext2_super_block::s_uuid[16]`
- 3.9.1.42 `char ext2_super_block::s_volume_name[16]`
- 3.9.1.43 `uint32_t ext2_super_block::s_wtime`

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

- include/fs/[ext2fs.h](#)

## 3.10 Gdtdesc Struct Reference

```
#include <memvals.h>
```

## Public Attributes

- `uint16_t size`
- `uint32_t base`

### 3.10.1 Member Data Documentation

3.10.1.1 `uint32_t Gdtdesc::base`

3.10.1.2 `uint16_t Gdtdesc::size`

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

- include/memvals.h

## 3.11 gpr\_regs\_t Struct Reference

```
#include <cpu_state.h>
```

## Public Attributes

- `reg_t edi`
- `reg_t esi`
- `reg_t ebp`
- `reg_t esp`
- `reg_t ebx`
- `reg_t edx`
- `reg_t ecx`
- `reg_t eax`

### 3.11.1 Member Data Documentation

3.11.1.1 `reg_t gpr_regs_t::eax`

3.11.1.2 `reg_t gpr_regs_t::ebp`

3.11.1.3 `reg_t gpr_regs_t::ebx`

3.11.1.4 `reg_t gpr_regs_t::ecx`

3.11.1.5 `reg_t gpr_regs_t::edi`

3.11.1.6 `reg_t gpr_regs_t::edx`

3.11.1.7 `reg_t gpr_regs_t::esi`

3.11.1.8 `reg_t gpr_regs_t::esp`

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

- include/arch/x86/[cpu\\_state.h](#)

## 3.12 Page Struct Reference

```
#include <page.h>
```

### Public Attributes

- `page_entry_t link`
- `uint16_t ref`

### 3.12.1 Member Data Documentation

3.12.1.1 `page_entry_t Page::link`

3.12.1.2 `uint16_t Page::ref`

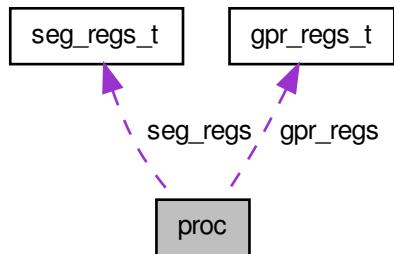
The documentation for this struct was generated from the following file:

- include/arch/x86/mm/[page.h](#)

## 3.13 proc Struct Reference

```
#include <proc.h>
```

Collaboration diagram for proc:



## Public Member Functions

- `LIST_ENTRY (proc) link`
- `LIFO_ENTRY (proc) q_link`

## Public Attributes

- `gpr_regs_t gpr_regs`
- `seg_regs_t seg_regs`
- `reg_t eip`
- `uint32_t cs`
- `reg_t eflags`
- `reg_t esp`
- `uint32_t ss`
- `uint32_t proc_id`
- `uint32_t proc_status`
- `pde_t * page_directory`
- `uint32_t cr3`
- `uint32_t preempted`
- `uint32_t dequeued`

### 3.13.1 Member Function Documentation

3.13.1.1 `proc::LIFO_ENTRY( proc )`

3.13.1.2 `proc::LIST_ENTRY( proc )`

### 3.13.2 Member Data Documentation

- 3.13.2.1 `uint32_t proc::cr3`
- 3.13.2.2 `uint32_t proc::cs`
- 3.13.2.3 `uint32_t proc::dequeqed`
- 3.13.2.4 `reg_t proc::eflags`
- 3.13.2.5 `reg_t proc::eip`
- 3.13.2.6 `reg_t proc::esp`
- 3.13.2.7 `gpr_regs_t proc::gpr_regs`
- 3.13.2.8 `pde_t* proc::page_directory`
- 3.13.2.9 `uint32_t proc::preempted`
- 3.13.2.10 `uint32_t proc::proc_id`
- 3.13.2.11 `uint32_t proc::proc_status`
- 3.13.2.12 `seg_regs_t proc::seg_regs`
- 3.13.2.13 `uint32_t proc::ss`

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

- include/proc/[proc.h](#)

## 3.14 proghdr Struct Reference

```
#include <elf.h>
```

### Public Attributes

- `uint32_t type`
- `uint32_t offset`
- `uint32_t vaddr`
- `uint32_t paddr`
- `uint32_t filesz`
- `uint32_t memsz`
- `uint32_t flags`
- `uint32_t align`

### 3.14.1 Member Data Documentation

- 3.14.1.1 `uint32_t proghdr::align`
- 3.14.1.2 `uint32_t proghdr::filesz`
- 3.14.1.3 `uint32_t proghdr::flags`
- 3.14.1.4 `uint32_t proghdr::memsz`
- 3.14.1.5 `uint32_t proghdr::offset`
- 3.14.1.6 `uint32_t proghdr::paddr`
- 3.14.1.7 `uint32_t proghdr::type`
- 3.14.1.8 `uint32_t proghdr::vaddr`

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

- include/arch/x86/elf.h

## 3.15 sechdr Struct Reference

```
#include <elf.h>
```

### Public Attributes

- `uint32_t name`
- `uint32_t type`
- `uint32_t flags`
- `uint32_t addr`
- `uint32_t offset`
- `uint32_t size`
- `uint32_t link`
- `uint32_t info`
- `uint32_t addralign`
- `uint32_t entsize`

### 3.15.1 Member Data Documentation

- 3.15.1.1 `uint32_t sechdr::addr`
- 3.15.1.2 `uint32_t sechdr::addralign`

3.15.1.3 `uint32_t sechdr::entsize`

3.15.1.4 `uint32_t sechdr::flags`

3.15.1.5 `uint32_t sechdr::info`

3.15.1.6 `uint32_t sechdr::link`

3.15.1.7 `uint32_t sechdr::name`

3.15.1.8 `uint32_t sechdr::offset`

3.15.1.9 `uint32_t sechdr::size`

3.15.1.10 `uint32_t sechdr::type`

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

- include/arch/x86/elf.h

## 3.16 `seg_regs_t` Struct Reference

```
#include <cpu_state.h>
```

### Public Attributes

- `reg_t gs`
- `reg_t fs`
- `reg_t es`
- `reg_t ds`

### 3.16.1 Member Data Documentation

3.16.1.1 `reg_t seg_regs_t::ds`

3.16.1.2 `reg_t seg_regs_t::es`

3.16.1.3 `reg_t seg_regs_t::fs`

3.16.1.4 `reg_t seg_regs_t::gs`

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

- include/arch/x86/cpu\_state.h

## 3.17 Segdesc Struct Reference

```
#include <memvals.h>
```

### Public Attributes

- unsigned `limit_0`: 16
- unsigned `base_0`: 16
- unsigned `base_1`: 8
- unsigned `permission`: 8
- unsigned `limit_1`: 4
- unsigned `flags`: 4
- unsigned `base`: 8

### 3.17.1 Member Data Documentation

3.17.1.1 unsigned `Segdesc::base`

3.17.1.2 unsigned `Segdesc::base_0`

3.17.1.3 unsigned `Segdesc::base_1`

3.17.1.4 unsigned `Segdesc::flags`

3.17.1.5 unsigned `Segdesc::limit_0`

3.17.1.6 unsigned `Segdesc::limit_1`

3.17.1.7 unsigned `Segdesc::permission`

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

- include/memvals.h

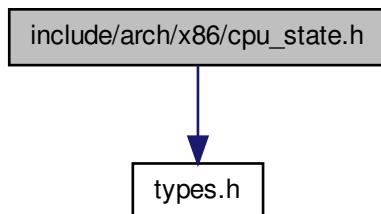


## Chapter 4

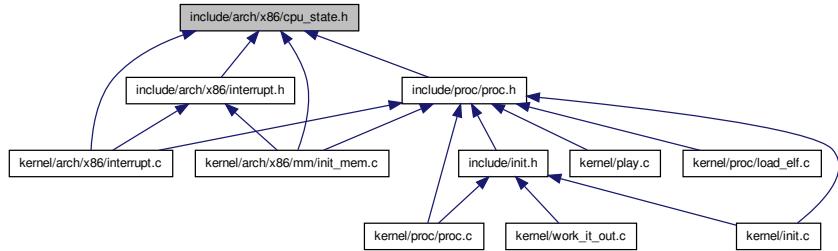
# File Documentation

### 4.1 include/arch/x86/cpu\_state.h File Reference

```
#include <types.h> Include dependency graph for cpu_state.h:
```



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



## Classes

- struct `cpu_state_t`
- struct `gpr_regs_t`
- struct `seg_regs_t`
- struct `__attribute__`
- struct `__attribute__`

## Defines

- #define `TSS_SIZE` 104
- #define `TSS_IOMAP_SIZE` ((0x8 \* 0x400) +1)
- #define `TASK_BUSY` 0xB
- #define `TASK_INACTIVE` 0x9
- #define `TSS_AVL` 1
- #define `TSS_DPL_KERNEL` 0
- #define `TSS_DPL_USER` 60
- #define `TSS_PRESENT` 0x80
- #define `TSS_GRANULARITY` 0x8
- #define `SEGMENT_TSS`(*limit*, *base*, *type*, *flags*)

### 4.1.1 Define Documentation

#### 4.1.1.1 #define SEGMENT\_TSS( *limit*, *base*, *type*, *flags* )

##### Value:

```
(struct Segdesc) \
{ \
    (((limit)>>12) & 0xfffff), \
    ((base) & 0xfffff), \
    (((base) >> 16) & 0xff), \
    \
```

```
(type), \
((limit) >> 28), \
((flags)), \
(((base) >> 24) & 0xff) \
}
```

4.1.1.2 #define **TASK\_BUSY** 0xB

4.1.1.3 #define **TASK\_INACTIVE** 0x9

4.1.1.4 #define **TSS\_AVL** 1

4.1.1.5 #define **TSS\_DPL\_KERNEL** 0

4.1.1.6 #define **TSS\_DPL\_USER** 60

4.1.1.7 #define **TSS\_GRANULARITY** 0x8

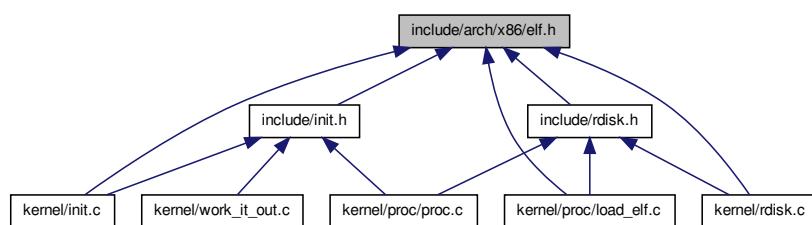
4.1.1.8 #define **TSS\_IOMAP\_SIZE** ((0x8 \* 0x400) +1)

4.1.1.9 #define **TSS\_PRESENT** 0x80

4.1.1.10 #define **TSS\_SIZE** 104

## 4.2 include/arch/x86/elf.h File Reference

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



## Classes

- struct **elfhdr**
- struct **proghdr**
- struct **sechdr**

## Defines

- #define `ELF_MAGIC` 0x464C457F
- #define `MAGIC_LEN` 16
- #define `M_CLASS_OFF` 4
- #define `M_CLASSNONE` 0
- #define `M_CLASS32` 1
- #define `M_CLASS64` 2
- #define `M_CLASSNUM` 3
- #define `M_DATA_OFF` 5
- #define `M_DATANONE` 0
- #define `M_DATA2LE` 1
- #define `M_DATA2BE` 2
- #define `M_DATANUM` 3
- #define `M_VERSION` 6
- #define `M_OSABI` 7
- #define `M_OSABI_SYSV` 0
- #define `M_OSABI_HPUX` 1
- #define `M_ABIVERSION` 8
- #define `M_ELF_PADDING` 9
- #define `T_TYPE_NONE` 0
- #define `T_TYPE_REL` 1
- #define `T_TYPE_EXEC` 2
- #define `T_TYPE_DYN` 3
- #define `T_TYPE_CORE` 4
- #define `T_TYPE_LOPROC` 0xff00
- #define `T_TYPE_HIPROC` 0xffff
- #define `M_MACHINE_I386` 3
- #define `V_VERSION_NONE` 0
- #define `V_VERSION_CURRENT` 1
- #define `V_VERSION_NUM` 2
- #define `P_PROGHDR_R` 0x4
- #define `P_PROGHDR_W` 0x2
- #define `P_PROGHDR_E` 0x1
- #define `PRG_LOAD` 1

## Typedefs

- typedef struct `elfhdr` `elfhdr`
- typedef struct `proghdr` `prohdr`
- typedef struct `sechdr` `secthdr`

#### 4.2.1 Define Documentation

- 4.2.1.1 #define **ELF\_MAGIC** 0x464C457F
- 4.2.1.2 #define **M\_ABIVERSION** 8
- 4.2.1.3 #define **M\_CLASS32** 1
- 4.2.1.4 #define **M\_CLASS64** 2
- 4.2.1.5 #define **M\_CLASS\_OFF** 4
- 4.2.1.6 #define **M\_CLASSNONE** 0
- 4.2.1.7 #define **M\_CLASSNUM** 3
- 4.2.1.8 #define **M\_DATA2BE** 2
- 4.2.1.9 #define **M\_DATA2LE** 1
- 4.2.1.10 #define **M\_DATA\_OFF** 5
- 4.2.1.11 #define **M\_DATANONE** 0
- 4.2.1.12 #define **M\_DATANUM** 3
- 4.2.1.13 #define **M\_ELF\_PADDING** 9
- 4.2.1.14 #define **M\_MACHINE\_I386** 3
- 4.2.1.15 #define **M\_OSABI** 7
- 4.2.1.16 #define **M\_OSABI\_HPUX** 1
- 4.2.1.17 #define **M\_OSABI\_SYSV** 0
- 4.2.1.18 #define **M\_VERSION** 6
- 4.2.1.19 #define **MAGIC\_LEN** 16
- 4.2.1.20 #define **P\_PROGHDR\_E** 0x1
- 4.2.1.21 #define **P\_PROGHDR\_R** 0x4
- 4.2.1.22 #define **P\_PROGHDR\_W** 0x2
- 4.2.1.23 #define **PRG\_LOAD** 1

4.2.1.24 #define T\_TYPE\_CORE 4

4.2.1.25 #define T\_TYPE\_DYN 3

4.2.1.26 #define T\_TYPE\_EXEC 2

4.2.1.27 #define T\_TYPE\_HIPROC 0xffff

4.2.1.28 #define T\_TYPE\_LOPROC 0xff00

4.2.1.29 #define T\_TYPE\_NONE 0

4.2.1.30 #define T\_TYPE\_REL 1

4.2.1.31 #define V\_VERSION\_CURRENT 1

4.2.1.32 #define V\_VERSION\_NONE 0

4.2.1.33 #define V\_VERSION\_NUM 2

## 4.2.2 Typedef Documentation

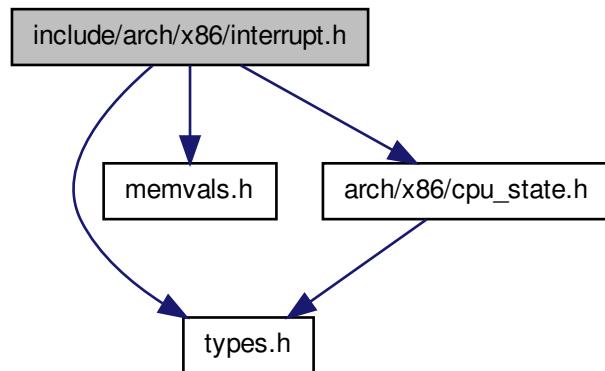
4.2.2.1 typedef struct elfhdr elfhdr

4.2.2.2 typedef struct proghdr prohdr

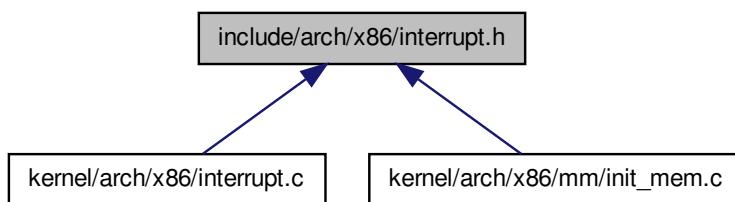
4.2.2.3 typedef struct sechdr secthdr

### 4.3 include/arch/x86/interrupt.h File Reference

```
#include <types.h> #include <memvals.h> #include <arch/x86/cpu-
_state.h> Include dependency graph for interrupt.h:
```



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



### Classes

- struct [\\_\\_attribute\\_\\_](#)

## Defines

- #define \_CATERNEL\_X86\_INTERRUPT\_H\_
- #define GATE\_TASK 0x5
- #define GATE\_INT16 0x6
- #define GATE\_TRAP16 0x7
- #define GATE\_INT32 0xE
- #define GATE\_TRAP32 0xF
- #define IDT\_ENTRIES 64
- #define GATE\_FILL(gate, type, dpl, sel, offset)
- #define VECTOR\_INDEX(cnt) vector\_##cnt
- #define GATE\_OFFSET(gate, vector)
- #define GP 13
- #define PF 14
- #define PF\_VIOLATION 0x1
- #define PF\_NOT\_PRESENT ~PF\_VIOLATION
- #define PF\_ON\_WRITE 0x2
- #define PF\_ON\_READ ~PF\_ON\_WRITE
- #define PF\_FROM\_USER 0x4
- #define PF\_FROM\_KERNEL ~PF\_FROM\_USER
- #define PF\_IFETCH 0x10
- #define PF\_RSRVD 0x8

## TypeDefs

- typedef struct Gdtdesc Idtdesc

## Functions

- void idt\_init (void)
- void map\_exception (uint32\_t, cpu\_state\_t \*)
- uint32\_t page\_fault\_handler (cpu\_state\_t \*)

## Variables

- gatedesc idt []
- char \* x86\_exception\_names []

### 4.3.1 Define Documentation

#### 4.3.1.1 #define \_CATERNEL\_X86\_INTERRUPT\_H\_

include/arch/x86/interrupt.h CATReloaded (C) Copyrights 2011 <http://catreloaded.net>

**Date**

27 Sept, 2012

**4.3.1.2 #define GATE\_FILL( gate, type, dpl, sel, offset )****Value:**

```
\n\n(gate)->offset_0_15 = (uint16_t)      (((offset) & 0xffff);\n(gate)->segment_s = (uint16_t) (sel);          \\\n(gate)->args      = 0;                      \\\n(gate)->reserved   = 0;                      \\\n(gate)->type       = 0xe;                   \\\n(gate)->dpl        = 0;                   \\\n(gate)->p          = 1;                   \\\n(gate)->offset_16_31 = (uint16_t) ((offset) >> 16);
```

**4.3.1.3 #define GATE\_INT16 0x6****4.3.1.4 #define GATE\_INT32 0xE****4.3.1.5 #define GATE\_OFFSET( gate, vector )****Value:**

```
(gate).offset_0_15 = (uint32_t) (((uint32_t) vector) & 0xffff);\n(gate).offset_16_31= (uint32_t) (((uint32_t) vector) >> 16);
```

**4.3.1.6 #define GATE\_TASK 0x5****4.3.1.7 #define GATE\_TRAP16 0x7****4.3.1.8 #define GATE\_TRAP32 0xF****4.3.1.9 #define GP 13****4.3.1.10 #define IDT\_ENTRIES 64****4.3.1.11 #define PF 14****4.3.1.12 #define PF\_FROM\_KERNEL ~PF\_FROM\_USER****4.3.1.13 #define PF\_FROM\_USER 0x4****4.3.1.14 #define PF\_IFETCH 0x10****4.3.1.15 #define PF\_NOT\_PRESENT ~PF\_VIOLATION**

4.3.1.16 #define PF\_ON\_READ ~PF\_ON\_WRITE

4.3.1.17 #define PF\_ON\_WRITE 0x2

4.3.1.18 #define PF\_RSRVD 0x8

4.3.1.19 #define PF\_VIOLATION 0x1

4.3.1.20 #define VECTOR\_INDEX( *cnt* ) vector\_##cnt

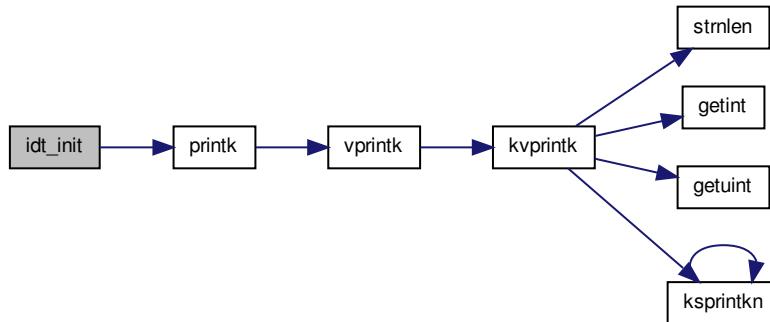
### 4.3.2 Typedef Documentation

4.3.2.1 **typedef struct Gdtdesc Idtdesc**

### 4.3.3 Function Documentation

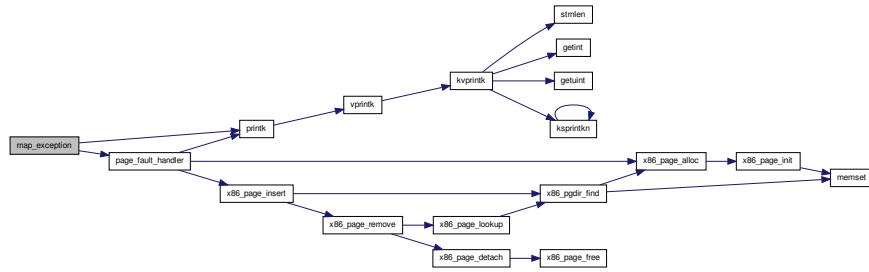
4.3.3.1 **void idt\_init( void )**

Here is the call graph for this function:



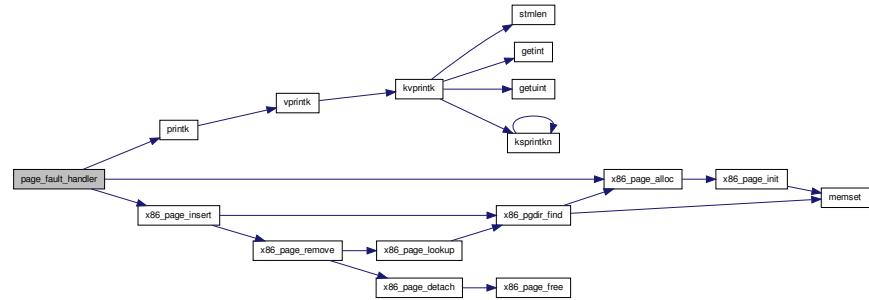
#### 4.3.3.2 void map\_exception ( uint32\_t , cpu\_state\_t \* )

Here is the call graph for this function:



#### 4.3.3.3 uint32\_t page\_fault\_handler ( cpu\_state\_t \* )

Here is the call graph for this function:



#### 4.3.4 Variable Documentation

##### 4.3.4.1 gatedesc idt[]

include/arch/x86/interrupt.h CATReloaded (C) Copyrights 2011 <http://catreload.net>

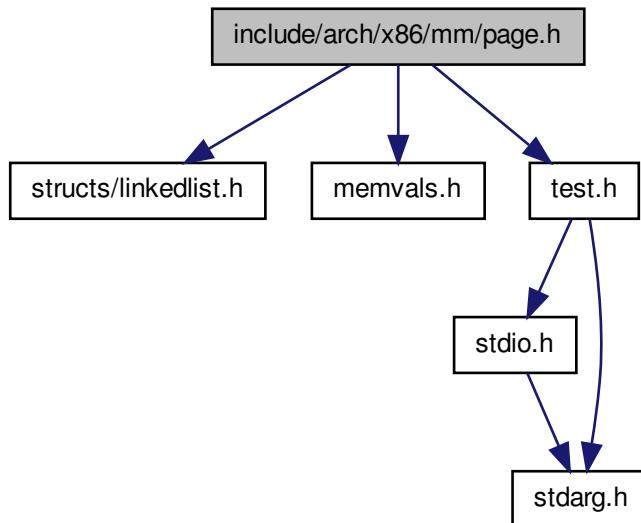
##### Date

27 Sept, 2012

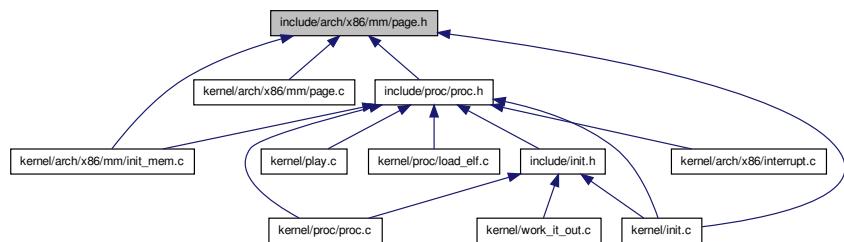
#### 4.3.4.2 char\* x86\_exception\_names[]

### 4.4 include/arch/x86/mm/page.h File Reference

```
#include <structs/linkedlist.h>      #include <memvals.h> x
#include <test.h> Include dependency graph for page.h:
```



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



## Classes

- struct `__attribute__`
- struct `Page`

## Defines

- `#define PAGE_PRESENT 0x1`
- `#define PAGE_WRITABLE 0x2`
- `#define PAGE_USER 0x4`
- `#define PAGE_WTHROUGH 0x8`
- `#define PAGE_CDISABLE 0x10`
- `#define PAGE_ACCESSED 0x20`
- `#define PAGE_DIRTY 0x40`
- `#define PAGE_DIRTY 0x40`
- `#define PAGE_GLOBAL 0x100`
- `#define PAGE_PAGESZ 0x80`
- `#define PGTBL_SHIFT 12`
- `#define PGDIR_SHIFT 22`
- `#define PGSHIFT 12`
- `#define PTSIIFT 22`
- `#define PGDIRX(lin) (((uint32_t)(lin)) >> PGDIR_SHIFT) & 0x3FF)`
- `#define VIRT_PGDIRX(lin) PGDIRX((lin))`
- `#define PGTBLX(lin) (((uint32_t)(lin)) >> PGTBL_SHIFT) & 0x3FF)`
- `#define VIRT_PGTBLX(lin) PGTBLX((lin))`
- `#define PGPN(lin) (((uint32_t)(lin)) >> PGTBL_SHIFT)`
- `#define VIRT_PGPN(lin) PGPN((lin))`
- `#define PGOFF(lin) (( (lin) & 0xFFFF))`
- `#define PTD_ADDR(lin) (( (uint32_t) (lin)) & ~0xFFFF)`
- `#define PA2KA(pa)`
- `#define KA2PA(va)`

## Typedefs

- `typedef uint32_t pde_t`

## Functions

- `LIST_HEAD (PageList, Page)`
- `typedef LIST_ENTRY (Page) page_entry_t`
- `void x86_paging_init (void)`
- `void x86_page_init (struct Page *)`
- `int x86_page_alloc (struct Page **)`
- `void x86_page_free (struct Page *)`
- `void x86_page_detach (struct Page *)`

- `pte_t * x86_pgdir_find (pde_t *, const void *, int)`
- `struct Page * x86_page_lookup (pde_t *, void *, pte_t **)`
- `void x86_page_remove (pde_t *, void *)`
- `int x86_page_insert (pde_t *, struct Page *, void *, uint32_t)`
- `void map_segment_page (pde_t *, vaddr_t, size_t, paddr_t, int)`
- `void x86_test_pgdir (void)`
- `void x86_test_paging (void)`

## Variables

- `size_t page_count`
- `volatile pte_t virtpgt []`
- `volatile pde_t virtpgd []`
- `char kernel_stack []`
- `char kernel_stack_end []`
- `struct Page * pages`
- `uint32_t global_cr3`
- `pde_t * global_pgdir`
- `struct Segdesc gdt []`
- `char * next_free`

### 4.4.1 Define Documentation

#### 4.4.1.1 #define KA2PA( va )

**Value:**

```
({
    vaddr_t m_va = (vaddr_t) (va); \
    if( m_va < KERNEL_ADDR){\
        panic("KA2PA called with bad address \n");\
    }\
    m_va - KERNEL_ADDR; \
})
```

#### 4.4.1.2 #define PA2KA( pa )

**Value:**

```
({
    paddr_t m_pa = ((paddr_t) pa); \
    paddr_t m_ppn = PGPN(m_pa); \
    if( m_ppn >= page_count){ \
        panic("PA2KA called with invalid paddr\n");\
    } \
    (void *) (m_pa + KERNEL_ADDR); \
})
```

```
4.4.1.3 #define PAGE_ACCESED 0x20
4.4.1.4 #define PAGE_CDISABLE 0x10
4.4.1.5 #define PAGE_DIRTY 0x40
4.4.1.6 #define PAGE_DIRTY 0x40
4.4.1.7 #define PAGE_GLOBAL 0x100
4.4.1.8 #define PAGE_PAGESZ 0x80
4.4.1.9 #define PAGE_PRESENT 0x1
4.4.1.10 #define PAGE_USER 0x4
4.4.1.11 #define PAGE_WRITABLE 0x2
4.4.1.12 #define PAGE_WTHROUGH 0x8
4.4.1.13 #define PGDIR_SHIFT 22
4.4.1.14 #define PGDIRX( lin ) (( (uint32_t)(lin) >> PGDIR_SHIFT) & 0x3FF)
4.4.1.15 #define PGOFF( lin ) (( (lin) & 0xFFFF))
4.4.1.16 #define PGPN( lin ) (((uint32_t)(lin) >> PGTBL_SHIFT)
4.4.1.17 #define PGSHIFT 12
4.4.1.18 #define PGTBL_SHIFT 12
4.4.1.19 #define PGTBLX( lin ) (( (uint32_t)(lin) >> PGTBL_SHIFT) & 0x3FF)
4.4.1.20 #define PTD_ADDR( lin ) (( (uint32_t)(lin) & ~0xFFFF)
4.4.1.21 #define PTSIIFT 22
4.4.1.22 #define VIRT_PGDIX( lin ) PGDIRX(lin)
4.4.1.23 #define VIRT_PGPN( lin ) PGPN(lin)
4.4.1.24 #define VIRT_PGTBLX( lin ) PGTBLX(lin)
```

## 4.4.2 Typedef Documentation

```
4.4.2.1 typedef uint32_t pde_t
```

#### 4.4.3 Function Documentation

4.4.3.1 `typedef LIST_ENTRY( Page )`

4.4.3.2 `LIST_HEAD( PageList , Page )`

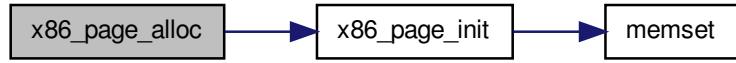
4.4.3.3 `void map_segment_page( pde_t * , vaddr_t , size_t , paddr_t , int )`

Here is the call graph for this function:



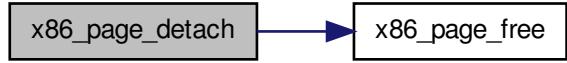
4.4.3.4 `int x86_page_alloc( struct Page ** )`

Here is the call graph for this function:



4.4.3.5 `void x86_page_detach( struct Page * )`

Here is the call graph for this function:



4.4.3.6 void x86\_page\_free ( struct Page \* )

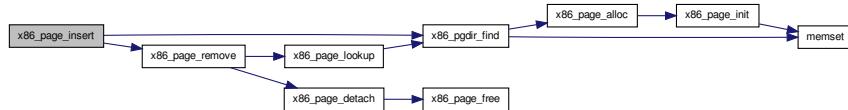
4.4.3.7 void x86\_page\_init ( struct Page \* )

Here is the call graph for this function:



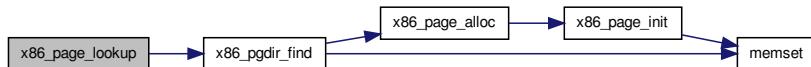
4.4.3.8 int x86\_page\_insert ( pde\_t \* , struct Page \* , void \* , uint32\_t )

Here is the call graph for this function:



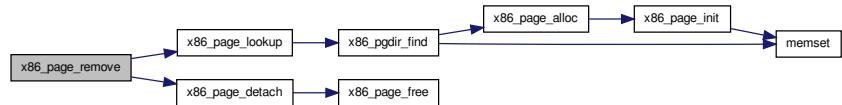
4.4.3.9 struct Page\* x86\_page\_lookup ( pde\_t \* , void \* , pte\_t \*\* ) [read]

Here is the call graph for this function:

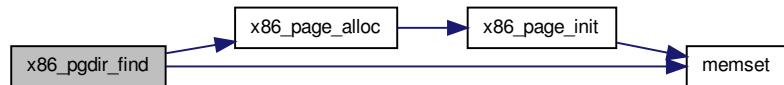


**4.4.3.10 void x86\_page\_remove ( pde\_t \*, void \* )**

Here is the call graph for this function:

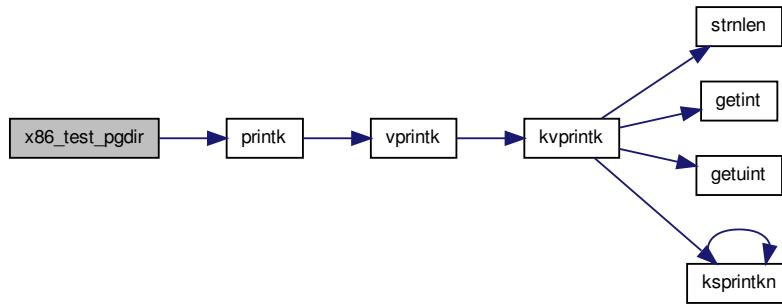
**4.4.3.11 void x86\_paging\_init ( void )****4.4.3.12 pte\_t\* x86\_pgdir\_find ( pde\_t \*, const void \* , int )**

Here is the call graph for this function:

**4.4.3.13 void x86\_test\_paging ( void )**

#### 4.4.3.14 void x86\_test\_pgdir( void )

Here is the call graph for this function:



#### 4.4.4 Variable Documentation

4.4.4.1 struct Segdesc gdt[]

4.4.4.2 uint32\_t global\_cr3

4.4.4.3 pde\_t\* global\_pgd

4.4.4.4 char kernel\_stack[]

4.4.4.5 char kernel\_stack\_end[]

4.4.4.6 char\* next\_free

4.4.4.7 size\_t page\_count

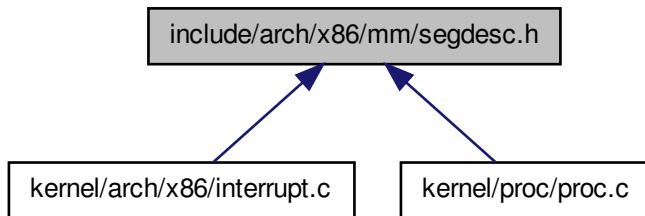
4.4.4.8 struct Page\* pages

4.4.4.9 volatile pde\_t virtpgd[]

4.4.4.10 volatile pte\_t virtpgt[]

## 4.5 include/arch/x86/mm/segdesc.h File Reference

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



### Defines

- #define SEG\_KERNCODE 0x8
- #define SEG\_KERNDATA 0x10
- #define SEG\_USERCODE 0x18
- #define SEG\_USERDATA 0x20
- #define SEG\_TSS 0x28

### 4.5.1 Define Documentation

4.5.1.1 #define SEG\_KERNCODE 0x8

4.5.1.2 #define SEG\_KERNDATA 0x10

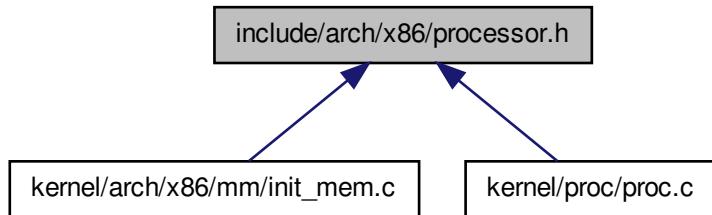
4.5.1.3 #define SEG\_TSS 0x28

4.5.1.4 #define SEG\_USERCODE 0x18

4.5.1.5 #define SEG\_USERDATA 0x20

## 4.6 include/arch/x86/processor.h File Reference

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



### Defines

- #define `X86_CR0_PE` 0x00000001 /\* Protection Enable \*/
- #define `X86_CR0_MP` 0x00000002 /\* Monitor Coprocessor \*/
- #define `X86_CR0_EM` 0x00000004 /\* Emulation \*/
- #define `X86_CR0_TS` 0x00000008 /\* Task Switched \*/
- #define `X86_CR0_ET` 0x00000010 /\* Extension type \*/
- #define `X86_CR0_NE` 0x00000020 /\* Numeric error \*/
- #define `X86_CR0_WP` 0x00100000 /\* Write Protect \*/
- #define `X86_CR0_AM` 0x00040000 /\* Alignment Mask \*/
- #define `X86_CR0_NW` 0x20000000 /\* Not writable through \*/
- #define `X86_CR0_CD` 0x40000000 /\* Cache disabled \*/
- #define `X86_CR0_PG` 0x80000000 /\* Paging enable \*/
- #define `X86_CR3_PWT` 0x00000008 /\* Page Write Through \*/
- #define `X86_CR3_PCD` 0x00000010 /\* Page Cache Disabled \*/
- #define `X86_CR4_VME` 0x00000001 /\* Enable VM86 Extension \*/
- #define `X86_CR4_PVI` 0x00000002 /\* Virtual interrupts flag enable \*/
- #define `X86_CR4_TSD` 0x00000004 /\* Disable time stamp at ipl 3 \*/
- #define `X86_CR4_DE` 0x00000008 /\* Enable Debugging extensions \*/
- #define `X86_CR4_PSE` 0x00000010 /\* Enable Page Size extension \*/
- #define `X86_CR4_PAE` 0x00000020 /\* Enable Physical address extension \*/
- #define `X86_CR4_MCE` 0x00000040 /\* Machine Check enable \*/
- #define `X86_CR4_PGE` 0x00000080 /\* Enable Global Pages \*/
- #define `X86_CR4_PCE` 0x00000100 /\* Enable Performance Counters at ipl 3\*/
- #define `X86_CR4_OSFXSR` 0x00000200 /\* Enable Fast FPU save and restore \*/
- #define `X86_CR4_OSXMMEXCPT` 0x00000400 /\* Enable unmasked SSE exceptions \*/

- #define X86\_CR4\_VMXE 0x00002000 /\* Enable VMX virtualization \*/
- #define X86\_CR4\_OSXSAVE 0x00004000 /\* Enable Xsave and xrestore \*/
- #define X86\_CR8\_TPR 0x0000000F /\* Task Priority Register \*/
- #define FLAG\_CF 0x1
- #define FLAG\_PF 0x4
- #define FLAG\_AF 0x10
- #define FLAG\_ZF 0x40
- #define FLAG\_SF 0x80
- #define FLAG\_TF 0x100
- #define FLAG\_IF 0x200
- #define FLAG\_DF 0x400
- #define FLAG\_OF 0x800
- #define FLAG\_IOPL 0x1000
- #define FLAG\_NT 0x2000
- #define FLAG\_RF 0x4000
- #define FLAG\_VM 0x8000
- #define FLAG\_AC 0x10000
- #define FLAG\_VIF 0x20000
- #define FLAG\_VIP 0x40000
- #define FLAG\_ID 0x80000

#### 4.6.1 Define Documentation

4.6.1.1 #define FLAG\_AC 0x10000

4.6.1.2 #define FLAG\_AF 0x10

4.6.1.3 #define FLAG\_CF 0x1

4.6.1.4 #define FLAG\_DF 0x400

4.6.1.5 #define FLAG\_ID 0x80000

4.6.1.6 #define FLAG\_IF 0x200

4.6.1.7 #define FLAG\_IOPL 0x1000

4.6.1.8 #define FLAG\_NT 0x2000

4.6.1.9 #define FLAG\_OF 0x800

4.6.1.10 #define FLAG\_PF 0x4

4.6.1.11 #define FLAG\_RF 0x4000

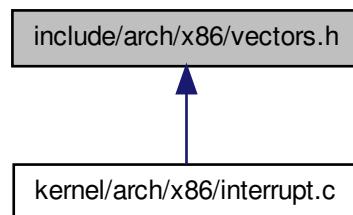
4.6.1.12 #define FLAG\_SF 0x80

```
4.6.1.13 #define FLAG_TF 0x100
4.6.1.14 #define FLAG_VIF 0x20000
4.6.1.15 #define FLAG_VIP 0x40000
4.6.1.16 #define FLAG_VM 0x8000
4.6.1.17 #define FLAG_ZF 0x40
4.6.1.18 #define X86_CR0_AM 0x00040000 /* Alignment Mask */
4.6.1.19 #define X86_CR0_CD 0x40000000 /* Cache disabled */
4.6.1.20 #define X86_CR0_EM 0x00000004 /* Emulation */
4.6.1.21 #define X86_CR0_ET 0x00000010 /* Extension type */
4.6.1.22 #define X86_CR0_MP 0x00000002 /* Monitor Coprocessor */
4.6.1.23 #define X86_CR0_NE 0x00000020 /* Numeric error */
4.6.1.24 #define X86_CR0_NW 0x20000000 /* Not writable through */
4.6.1.25 #define X86_CR0_PE 0x00000001 /* Protection Enable */
4.6.1.26 #define X86_CR0_PG 0x80000000 /* Paging enable */
4.6.1.27 #define X86_CR0_TS 0x00000008 /* Task Switched */
4.6.1.28 #define X86_CR0_WP 0x00010000 /* Write Protect */
4.6.1.29 #define X86_CR3_PCD 0x00000010 /* Page Cache Disabled */
4.6.1.30 #define X86_CR3_PWT 0x00000008 /* Page Write Through */
4.6.1.31 #define X86_CR4_DE 0x00000008 /* Enable Debugging extensions */
4.6.1.32 #define X86_CR4_MCE 0x00000040 /* Machine Check enable */
4.6.1.33 #define X86_CR4_OSFXSR 0x00000200 /* Enable Fast FPU save and restore */
4.6.1.34 #define X86_CR4_OSXMMEXCPT 0x00000400 /* Enable unmasked SSE exceptions */
4.6.1.35 #define X86_CR4_OSXSAVE 0x00004000 /* Enable Xsave and xrestore */
```

- 4.6.1.36 #define X86\_CR4\_PAE 0x00000020 /\* Enable Physical address extension \*/
- 4.6.1.37 #define X86\_CR4\_PCE 0x00000100 /\* Enable Performance Counters at ipl 3\*/
- 4.6.1.38 #define X86\_CR4\_PGE 0x00000080 /\* Enable Global Pages \*/
- 4.6.1.39 #define X86\_CR4\_PSE 0x00000010 /\* Enable Page Size extension \*/
- 4.6.1.40 #define X86\_CR4\_PVI 0x00000002 /\* Virtual interrupts flag enable \*/
- 4.6.1.41 #define X86\_CR4\_TSD 0x00000004 /\* Disable time stamp at ipl 3 \*/
- 4.6.1.42 #define X86\_CR4\_VME 0x00000001 /\* Enable VM86 Extension \*/
- 4.6.1.43 #define X86\_CR4\_VMXE 0x00002000 /\* Enable VMX virtualization \*/
- 4.6.1.44 #define X86\_CR8\_TPR 0x0000000F /\* Task Priority Register \*/

## 4.7 include/arch/x86/vectors.h File Reference

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



### Variables

- `uint32_t vector_0`
- `uint32_t vector_1`
- `uint32_t vector_2`
- `uint32_t vector_3`
- `uint32_t vector_4`
- `uint32_t vector_5`
- `uint32_t vector_6`
- `uint32_t vector_7`

- [uint32\\_t vector\\_8](#)
- [uint32\\_t vector\\_9](#)
- [uint32\\_t vector\\_10](#)
- [uint32\\_t vector\\_11](#)
- [uint32\\_t vector\\_12](#)
- [uint32\\_t vector\\_13](#)
- [uint32\\_t vector\\_14](#)
- [uint32\\_t vector\\_15](#)
- [uint32\\_t vector\\_16](#)
- [uint32\\_t vector\\_17](#)
- [uint32\\_t vector\\_18](#)
- [uint32\\_t vector\\_19](#)
- [uint32\\_t vector\\_20](#)
- [uint32\\_t vector\\_21](#)
- [uint32\\_t vector\\_22](#)
- [uint32\\_t vector\\_23](#)
- [uint32\\_t vector\\_24](#)
- [uint32\\_t vector\\_25](#)
- [uint32\\_t vector\\_26](#)
- [uint32\\_t vector\\_27](#)
- [uint32\\_t vector\\_28](#)
- [uint32\\_t vector\\_29](#)
- [uint32\\_t vector\\_30](#)
- [uint32\\_t vector\\_31](#)
- [uint32\\_t vector\\_32](#)
- [uint32\\_t vector\\_33](#)
- [uint32\\_t vector\\_34](#)
- [uint32\\_t vector\\_35](#)
- [uint32\\_t vector\\_36](#)
- [uint32\\_t vector\\_37](#)
- [uint32\\_t vector\\_38](#)
- [uint32\\_t vector\\_39](#)
- [uint32\\_t vector\\_40](#)
- [uint32\\_t vector\\_41](#)
- [uint32\\_t vector\\_42](#)
- [uint32\\_t vector\\_43](#)
- [uint32\\_t vector\\_44](#)
- [uint32\\_t vector\\_45](#)
- [uint32\\_t vector\\_46](#)
- [uint32\\_t vector\\_47](#)
- [uint32\\_t vector\\_48](#)
- [uint32\\_t vector\\_49](#)
- [uint32\\_t vector\\_50](#)
- [uint32\\_t vector\\_51](#)
- [uint32\\_t vector\\_52](#)
- [uint32\\_t vector\\_53](#)

- `uint32_t vector_54`
- `uint32_t vector_55`
- `uint32_t vector_56`
- `uint32_t vector_57`
- `uint32_t vector_58`
- `uint32_t vector_59`
- `uint32_t vector_60`
- `uint32_t vector_61`
- `uint32_t vector_62`
- `uint32_t vector_63`

#### 4.7.1 Variable Documentation

- 4.7.1.1 `uint32_t vector_0`
- 4.7.1.2 `uint32_t vector_1`
- 4.7.1.3 `uint32_t vector_10`
- 4.7.1.4 `uint32_t vector_11`
- 4.7.1.5 `uint32_t vector_12`
- 4.7.1.6 `uint32_t vector_13`
- 4.7.1.7 `uint32_t vector_14`
- 4.7.1.8 `uint32_t vector_15`
- 4.7.1.9 `uint32_t vector_16`
- 4.7.1.10 `uint32_t vector_17`
- 4.7.1.11 `uint32_t vector_18`
- 4.7.1.12 `uint32_t vector_19`
- 4.7.1.13 `uint32_t vector_2`
- 4.7.1.14 `uint32_t vector_20`
- 4.7.1.15 `uint32_t vector_21`
- 4.7.1.16 `uint32_t vector_22`
- 4.7.1.17 `uint32_t vector_23`

4.7.1.18 `uint32_t vector_24`  
4.7.1.19 `uint32_t vector_25`  
4.7.1.20 `uint32_t vector_26`  
4.7.1.21 `uint32_t vector_27`  
4.7.1.22 `uint32_t vector_28`  
4.7.1.23 `uint32_t vector_29`  
4.7.1.24 `uint32_t vector_3`  
4.7.1.25 `uint32_t vector_30`  
4.7.1.26 `uint32_t vector_31`  
4.7.1.27 `uint32_t vector_32`  
4.7.1.28 `uint32_t vector_33`  
4.7.1.29 `uint32_t vector_34`  
4.7.1.30 `uint32_t vector_35`  
4.7.1.31 `uint32_t vector_36`  
4.7.1.32 `uint32_t vector_37`  
4.7.1.33 `uint32_t vector_38`  
4.7.1.34 `uint32_t vector_39`  
4.7.1.35 `uint32_t vector_4`  
4.7.1.36 `uint32_t vector_40`  
4.7.1.37 `uint32_t vector_41`  
4.7.1.38 `uint32_t vector_42`  
4.7.1.39 `uint32_t vector_43`  
4.7.1.40 `uint32_t vector_44`  
4.7.1.41 `uint32_t vector_45`

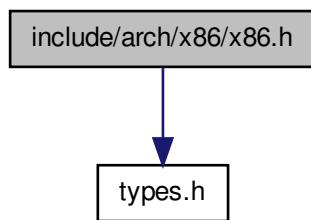
---

4.7.1.42 `uint32_t vector_46`  
4.7.1.43 `uint32_t vector_47`  
4.7.1.44 `uint32_t vector_48`  
4.7.1.45 `uint32_t vector_49`  
4.7.1.46 `uint32_t vector_5`  
4.7.1.47 `uint32_t vector_50`  
4.7.1.48 `uint32_t vector_51`  
4.7.1.49 `uint32_t vector_52`  
4.7.1.50 `uint32_t vector_53`  
4.7.1.51 `uint32_t vector_54`  
4.7.1.52 `uint32_t vector_55`  
4.7.1.53 `uint32_t vector_56`  
4.7.1.54 `uint32_t vector_57`  
4.7.1.55 `uint32_t vector_58`  
4.7.1.56 `uint32_t vector_59`  
4.7.1.57 `uint32_t vector_6`  
4.7.1.58 `uint32_t vector_60`  
4.7.1.59 `uint32_t vector_61`  
4.7.1.60 `uint32_t vector_62`  
4.7.1.61 `uint32_t vector_63`  
4.7.1.62 `uint32_t vector_7`  
4.7.1.63 `uint32_t vector_8`  
4.7.1.64 `uint32_t vector_9`

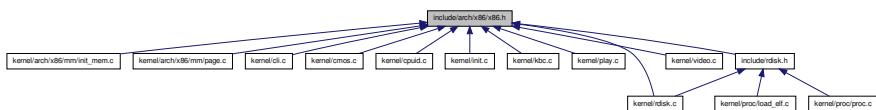
---

## 4.8 include/arch/x86/x86.h File Reference

#include <types.h> Include dependency graph for x86.h:

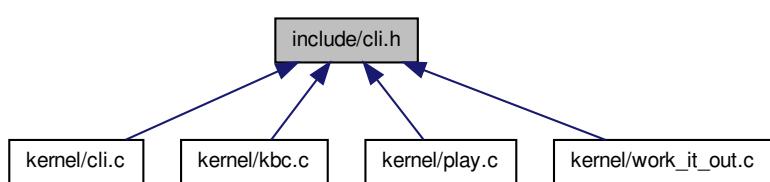


This graph shows which files directly or indirectly include this file:



## 4.9 include/cli.h File Reference

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



## Defines

- #define MAXBUFSIZE 512

## Functions

- void `console_init` (void)
- void `console_interrupt` (int(\*intr)(void))
- int `console_getc` (void)
- void `console_putc` (int c)

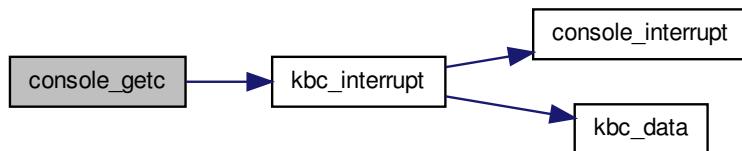
### 4.9.1 Define Documentation

#### 4.9.1.1 #define MAXBUFSIZE 512

### 4.9.2 Function Documentation

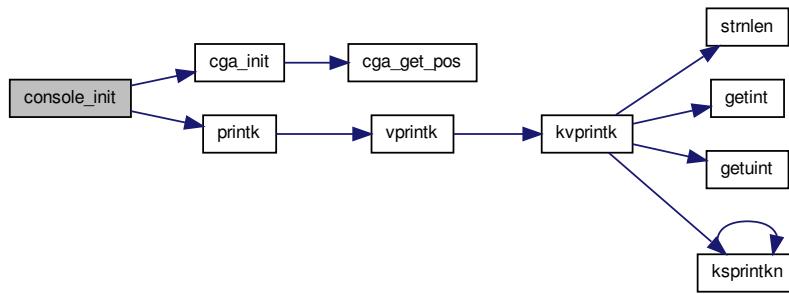
#### 4.9.2.1 int `console_getc` ( void )

Here is the call graph for this function:



#### 4.9.2.2 void console\_init( void )

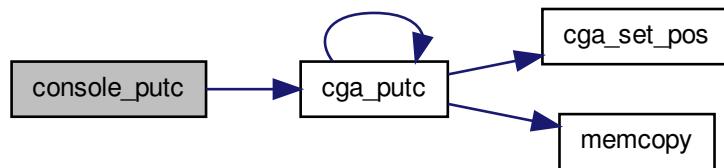
Here is the call graph for this function:



#### 4.9.2.3 void console\_interrupt( int(\*)(void) intr )

#### 4.9.2.4 void console\_putc( int c )

Here is the call graph for this function:



### 4.9.3 Variable Documentation

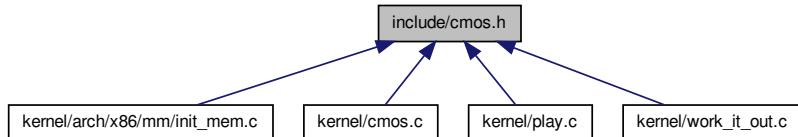
#### 4.9.3.1 uint8\_t buf[MAXBUFSIZE]

#### 4.9.3.2 uint32\_t rpos

#### 4.9.3.3 uint32\_t wpos

## 4.10 include/cmos.h File Reference

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



### Defines

- #define CMOS\_INDEXPORT 0x70 /\* the CMOS/RTC index register port \*/
- #define CMOS\_DATAPORT 0x71 /\* the CMOS/RTC data register port \*/
- #define RTC\_SECONDS 0x0
- #define RTC\_ALRMSECOND 0x1
- #define RTC\_MINUTES 0x2
- #define RTC\_ALRMMINUTE 0x3
- #define RTC\_HOUR 0x4
- #define RTC\_ALRMHOUR 0x5
- #define RTC\_DAY\_WEEK 0x6
- #define RTC\_DAY\_MONTH 0x7
- #define RTC\_MONTH 0x8
- #define RTC\_YEAR 0x9
- #define RTC\_STATUS\_A 0xA /\* the RTC Status register A \*/
- #define RTC\_STATUS\_B 0xB /\* the RTC Status register B \*/
- #define STAT\_RUN 0x7F
- #define STAT\_HALT 0x80
- #define STAT\_PER\_INTR 0x40
- #define STAT\_ALRM\_INTR 0x20
- #define STAT\_UPDT\_INTR 0x10
- #define STAT\_SQRWV\_INTR 0x08
- #define STAT\_CAL\_BIN 0x04
- #define STAT\_CAL\_BCD 0xFB
- #define STAT\_CAL\_HR24 0x02
- #define STAT\_CAL\_HR12 0xFD
- #define STAT\_DAY\_LGHT 0x01
- #define CMOS\_STATUS\_C 0xC
- #define CMOS\_STATUS\_D 0xD
- #define CMOS\_STATUS\_E 0xE /\* RTC diagnostic register E \*/
- #define CMOS\_STATUS\_F 0xF /\* RTC Shut-down status register F \*/

- #define CMOS\_STATUS\_10 0x10 /\* RTC DRIVER TYPE for A: and B: register \*/
- #define CMOS\_FIXED\_DISK 0x12 /\* RTC Fixed disk drive type for drive 0 and drive 1 register \*/
- #define CMOS\_EQUIP\_BYTE 0x14 /\* RTC Equip byte \*/
- #define CMOS\_SYSBASE\_LSB 0x15 /\* LSB of System base memory in Kilo byte\*/
- #define CMOS\_SYSBASE\_MSB 0x16 /\* MSB of ~~~~~\*/
- #define CMOS\_EXTMEM\_LSB 0x17 /\* LSB of total extended memory in Kilo byte \*/
- #define CMOS\_EXTMEM\_MSB 0x18 /\* MSB of ~~~~~\*/
- #define CMOS\_DRIVE\_C 0x19 /\* Drive C extention type \*/
- #define CMOS\_DRIVE\_D 0x1A /\* Drive D extention type \*/

#### 4.10.1 Define Documentation

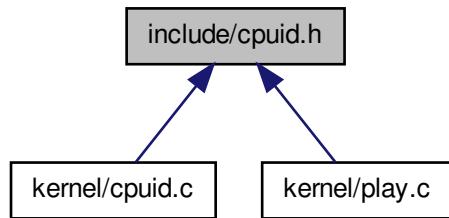
- 4.10.1.1 #define CMOS\_DATAPORT 0x71 /\* the CMOS/RTC data register port \*/
- 4.10.1.2 #define CMOS\_DRIVE\_C 0x19 /\* Drive C extention type \*/
- 4.10.1.3 #define CMOS\_DRIVE\_D 0x1A /\* Drive D extention type \*/
- 4.10.1.4 #define CMOS\_EQUIP\_BYTE 0x14 /\* RTC Equip byte \*/
- 4.10.1.5 #define CMOS\_EXTMEM\_LSB 0x17 /\* LSB of total extended memory in Kilo byte \*/
- 4.10.1.6 #define CMOS\_EXTMEM\_MSB 0x18 /\* MSB of ~~~~~\*/
- 4.10.1.7 #define CMOS\_FIXED\_DISK 0x12 /\* RTC Fixed disk drive type for drive 0 and drive 1 register \*/
- 4.10.1.8 #define CMOS\_INDEXPORT 0x70 /\* the CMOS/RTC index register port \*/
- 4.10.1.9 #define CMOS\_STATUS\_10 0x10 /\* RTC DRIVER TYPE for A: and B: register \*/
- 4.10.1.10 #define CMOS\_STATUS\_C 0xC
- 4.10.1.11 #define CMOS\_STATUS\_D 0xD
- 4.10.1.12 #define CMOS\_STATUS\_E 0xE /\* RTC diagnostic register E \*/
- 4.10.1.13 #define CMOS\_STATUS\_F 0xF /\* RTC Shut-down status register F \*/
- 4.10.1.14 #define CMOS\_SYSBASE\_LSB 0x15 /\* LSB of System base memory in Kilo byte\*/

```
4.10.1.15 #define CMOS_SYSBASE_MSB 0x16 /* MSB of  
~~~~~*/  
  
4.10.1.16 #define RTC_ALRMHOUR 0x5  
  
4.10.1.17 #define RTC_ALRMMINUTE 0x3  
  
4.10.1.18 #define RTC_ALARMSECOND 0x1  
  
4.10.1.19 #define RTC_DAY_MONTH 0x7  
  
4.10.1.20 #define RTC_DAY_WEEK 0x6  
  
4.10.1.21 #define RTC_HOUR 0x4  
  
4.10.1.22 #define RTC_MINUTES 0x2  
  
4.10.1.23 #define RTC_MONTH 0x8  
  
4.10.1.24 #define RTC_SECONDS 0x0  
  
4.10.1.25 #define RTC_STATUS_A 0xA /* the RTC Status register A */  
  
4.10.1.26 #define RTC_STATUS_B 0xB /* the RTC Status register B */  
  
4.10.1.27 #define RTC_YEAR 0x9  
  
4.10.1.28 #define STAT_ALRM_INTR 0x20  
  
4.10.1.29 #define STAT_CAL_BCD 0xFB  
  
4.10.1.30 #define STAT_CAL_BIN 0x04  
  
4.10.1.31 #define STAT_CAL_HR12 0xFD  
  
4.10.1.32 #define STAT_CAL_HR24 0x02  
  
4.10.1.33 #define STAT_DAY_LGHT 0x01  
  
4.10.1.34 #define STAT_HALT 0x80  
  
4.10.1.35 #define STAT_PER_INTR 0x40  
  
4.10.1.36 #define STAT_RUN 0x7F  
  
4.10.1.37 #define STAT_SQRWV_INTR 0x08
```

4.10.1.38 #define STAT\_UPDT\_INTR 0x10

## 4.11 include/cpuid.h File Reference

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



### Defines

- #define CPUID\_INFO 0x0
- #define CPUID\_VERSION 0x1

### 4.11.1 Define Documentation

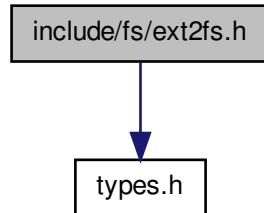
4.11.1.1 #define CPUID\_INFO 0x0

4.11.1.2 #define CPUID\_VERSION 0x1

## 4.12 include/device.h File Reference

## 4.13 include/fs/ext2fs.h File Reference

#include <types.h> Include dependency graph for ext2fs.h:



### Classes

- struct `ext2_super_block`
- struct `ext2_dir_entry`
- struct `ext2_group_desc`
- struct `ext2_inode`

### Defines

- #define `EXT2_NDIR_BLOCKS` 12
- #define `EXT2_IND_BLOCK` `EXT2_NDIR_BLOCKS`
- #define `EXT2_DIND_BLOCK` (`EXT2_IND_BLOCK` + 1)
- #define `EXT2_TIND_BLOCK` (`EXT2_DIND_BLOCK` + 1)
- #define `EXT2_N_BLOCKS` (`EXT2_TIND_BLOCK` + 1)
- #define `EXT2_INOSZ_V0` 128
- #define `EXT2_NAME_LEN` 255
- #define `EXT2_CLEAN` 1
- #define `EXT2_ERROR` 2
- #define `EXT2_BAD_INO` 1 /\* Bad blocks inode \*/
- #define `EXT2_ROOT_INO` 2 /\* Root inode \*/
- #define `EXT2_ACL_IDX_INO` 3 /\* ACL inode \*/
- #define `EXT2_ACL_DATA_INO` 4 /\* ACL inode \*/
- #define `EXT2_BOOT_LOADER_INO` 5 /\* Boot loader inode \*/
- #define `EXT2_UNDEL_DIR_INO` 6 /\* Undelete directory inode \*/
- #define `EXT2_FIRST_INO` 11 /\* First non reserved inode \*/
- #define `EXT2_PRE_02B_MAGIC` 0xEF51
- #define `EXT2_SUPER_MAGIC` 0xEF53

- #define EXT2\_LINK\_MAX 32000
- #define EXT2\_IGNORE 1
- #define EXT2\_REMOUNT\_RO 2
- #define EXT2\_KPANIC 3
- #define EXT2\_OS\_LINUX 0
- #define EXT2\_OS\_HURD 1
- #define EXT2\_OS\_MASIX 2
- #define EXT2\_OS\_FBSD 3
- #define EXT2\_OS\_LITES 4
- #define EXT2\_ROOT\_INODE 2 /\* Root inode \*/
- #define I\_FIFO 0x1000
- #define I\_CHAR\_DEVICE 0x2000
- #define I\_DIRECTORY 0x4000
- #define I\_BLOCK\_DEVICE 0x6000
- #define I\_REG\_FILE 0x8000
- #define I\_SYM\_LINK 0xA000
- #define I\_UNIX SOCK 0xC000
- #define I\_OEXEC\_P 0X001
- #define I\_OWRITE\_P 0x002
- #define I\_OREAD\_P 0x004
- #define I\_GEXEC\_P 0x008
- #define I\_GWRITE\_P 0x010
- #define I\_GREAD\_P 0x020
- #define I\_UEXEC\_P 0x040
- #define I\_UWRITE\_P 0x080
- #define I\_UREAD\_P 0x100
- #define I\_STICKY 0x200
- #define I\_SET\_GID 0x400
- #define I\_SET\_UID 0x800
- #define I\_SEC\_DEL 0x00000001 /\*Secure deletion\*/
- #define I\_KEEP\_CPY 0x00000002 /\*Keep a copy of data when deleted\*/
- #define I\_FILE\_COMPRESSION 0x00000004 /\*File compression\*/
- #define I\_SYNC\_UPDATES 0x00000008 /\*Synchronous updates\*/
- #define I\_IMMUTABLE\_FILE 0x00000010 /\*Immutable file\*/
- #define I\_APPEND\_ONLY 0x00000020 /\* Append only \*/
- #define I\_NO\_DUMP 0x00000040 /\* File is not included in 'dump' command \*/
- #define I\_NO\_LAT 0x00000080 /\* Last accessed time should not updated \*/
- #define I\_HASH\_INDEXED\_DIR 0x00010000 /\* Hash indexed directory \*/
- #define I\_AFS\_DIR 0x00020000 /\* AFS directory \*/
- #define I\_JOURNAL\_FILE\_DATA 0x00040000 /\* Journal file data \*/
- #define I\_DIR\_UNKOWN 0x0
- #define I\_DIR\_REGULAR\_F 0x1
- #define I\_DIR\_DIRECTORY 0x2
- #define I\_DIR\_CHAR\_DEV 0x3
- #define I\_DIR\_BLOCK\_DEV 0x4
- #define I\_DIR\_FIFO 0x5

- #define `I_DIR_SOCK` 0x6
- #define `I_DIR_SYM_LINK` 0x7
- #define `EXT2_PRELOC` 0x0001
- #define `EXT2_AFS_INO` 0x0002
- #define `EXT2_JRNL` 0x0004
- #define `EXT2_INO_EXTNDAATTR` 0x0008
- #define `EXT2_FS_RESIZE` 0x0010
- #define `EXT2_DIR_HASHI` 0x0020
- #define `EXT2_COMPRESSION` 0x0001
- #define `EXT2_DIR_TYPE_F` 0x0002
- #define `EXT2_FS_REPLY_JOR` 0x0004
- #define `EXT2_FS_JOR_DEV` 0x0008
- #define `EXT2_SPARSE` 0x0001
- #define `EXT2_64FZ` 0x0002
- #define `EXT2_DIR_BIN_TREE` 0x0004

#### 4.13.1 Define Documentation

- 4.13.1.1 #define `EXT2_64FZ` 0x0002
- 4.13.1.2 #define `EXT2_ACL_DATA_INO` 4 /\* ACL inode \*/
- 4.13.1.3 #define `EXT2_ACL_IDX_INO` 3 /\* ACL inode \*/
- 4.13.1.4 #define `EXT2_AFS_INO` 0x0002
- 4.13.1.5 #define `EXT2_BAD_INO` 1 /\* Bad blocks inode \*/
- 4.13.1.6 #define `EXT2_BOOT_LOADER_INO` 5 /\* Boot loader inode \*/
- 4.13.1.7 #define `EXT2_CLEAN` 1
- 4.13.1.8 #define `EXT2_COMPRESSION` 0x0001
- 4.13.1.9 #define `EXT2_DIND_BLOCK` (`EXT2_IND_BLOCK` + 1)
- 4.13.1.10 #define `EXT2_DIR_BIN_TREE` 0x0004
- 4.13.1.11 #define `EXT2_DIR_HASHI` 0x0020
- 4.13.1.12 #define `EXT2_DIR_TYPE_F` 0x0002
- 4.13.1.13 #define `EXT2_ERR0R` 2
- 4.13.1.14 #define `EXT2_FIRST_INO` 11 /\* First non reserved inode \*/

```
4.13.1.15 #define EXT2_FS_JOR_DEV 0x0008
4.13.1.16 #define EXT2_FS_REPLAY_JOR 0x0004
4.13.1.17 #define EXT2_FS_RESIZE 0x0010
4.13.1.18 #define EXT2_IGNORE 1
4.13.1.19 #define EXT2_IND_BLOCK EXT2_NDIR_BLOCKS
4.13.1.20 #define EXT2_INO_EXTNDATTR 0x0008
4.13.1.21 #define EXT2_INOSZ_V0 128
4.13.1.22 #define EXT2_JRNL 0x0004
4.13.1.23 #define EXT2_KPANIC 3
4.13.1.24 #define EXT2_LINK_MAX 32000
4.13.1.25 #define EXT2_N_BLOCKS (EXT2_TIND_BLOCK + 1)
4.13.1.26 #define EXT2_NAME_LEN 255
4.13.1.27 #define EXT2_NDIR_BLOCKS 12
4.13.1.28 #define EXT2_OS_FBSD 3
4.13.1.29 #define EXT2_OS_HURD 1
4.13.1.30 #define EXT2_OS_LINUX 0
4.13.1.31 #define EXT2_OS_LITES 4
4.13.1.32 #define EXT2_OS_MASIX 2
4.13.1.33 #define EXT2_PRE_02B_MAGIC 0xEF51
4.13.1.34 #define EXT2_PRELOC 0x0001
```

## Features

```
4.13.1.35 #define EXT2_REMOUNT_RO 2
4.13.1.36 #define EXT2_ROOT_INO 2 /* Root inode */
4.13.1.37 #define EXT2_ROOT_INODE 2 /* Root inode */
```

---

```
4.13.1.38 #define EXT2_SPARSE 0x0001  
4.13.1.39 #define EXT2_SUPER_MAGIC 0xEF53  
4.13.1.40 #define EXT2_TIND_BLOCK (EXT2_DIND_BLOCK + 1)  
4.13.1.41 #define EXT2_UNDEL_DIR_INO 6 /* Undelete directory inode */  
4.13.1.42 #define I_AFS_DIR 0x00020000 /* AFS directory */  
4.13.1.43 #define I_APPEND_ONLY 0x00000020 /* Append only */  
4.13.1.44 #define I_BLOCK_DEVICE 0x6000  
4.13.1.45 #define I_CHAR_DEVICE 0x2000  
4.13.1.46 #define I_DIR_BLOCK_DEV 0x4  
4.13.1.47 #define I_DIR_CHAR_DEV 0x3  
4.13.1.48 #define I_DIR_DIRECTORY 0x2  
4.13.1.49 #define I_DIR_FIFO 0x5  
4.13.1.50 #define I_DIR_REGULAR_F 0x1  
4.13.1.51 #define I_DIR SOCK 0x6  
4.13.1.52 #define I_DIR_SYM_LINK 0x7  
4.13.1.53 #define I_DIR_UNKOWN 0x0
```

#### Directory Entry Types

```
4.13.1.54 #define I_DIRECTORY 0x4000  
4.13.1.55 #define I_FIFO 0x1000  
4.13.1.56 #define I_FILE_COMPRESSION 0x00000004 /*File compression*/  
4.13.1.57 #define I_GEXEC_P 0x008  
4.13.1.58 #define I_GREAD_P 0x020  
4.13.1.59 #define I_GWRITE_P 0x010  
4.13.1.60 #define I_HASH_INDEXED_DIR 0x00010000 /* Hash indexed directory */
```

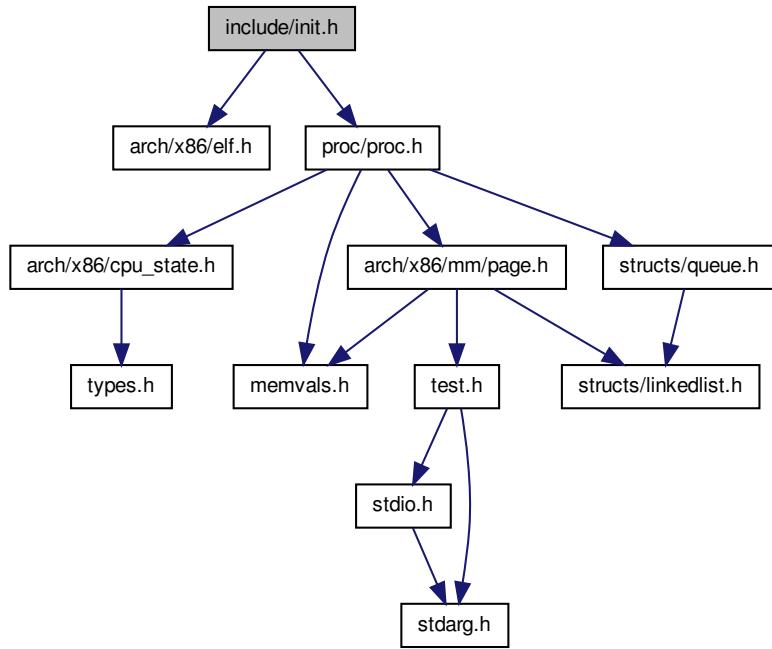
```
4.13.1.61 #define I_IMMUTABLE_FILE 0x00000010 /*Immutable file*/  
4.13.1.62 #define I_JOURNAL_FILE_DATA 0x00040000 /* Journal file data */  
4.13.1.63 #define I_KEEP_CPY 0x00000002 /*Keep a copy of data when deleted*/  
4.13.1.64 #define I_NO_DUMP 0x00000040 /* File is not included in 'dump' command */  
4.13.1.65 #define I_NO_LAT 0x00000080 /* Last accessed time should not updated */  
4.13.1.66 #define I_OEXEC_P 0X001  
4.13.1.67 #define I_OREAD_P 0x004  
4.13.1.68 #define I_OWRITE_P 0x002  
4.13.1.69 #define I_REG_FILE 0x8000  
4.13.1.70 #define I_SEC_DEL 0x00000001 /*Secure deletion*/
```

#### Inode flags

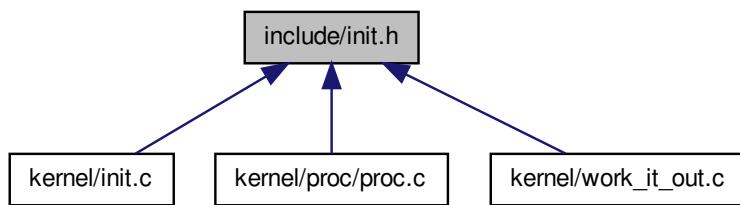
```
4.13.1.71 #define I_SET_GID 0x400  
4.13.1.72 #define I_SET_UID 0x800  
4.13.1.73 #define I_STICKY 0x200  
4.13.1.74 #define I_SYM_LINK 0xA000  
4.13.1.75 #define I_SYNC_UPDATES 0x00000008 /*Synchronous updates*/  
4.13.1.76 #define I_UEXEC_P 0x040  
4.13.1.77 #define I_UNIX SOCK 0xC000  
4.13.1.78 #define I_UREAD_P 0x100  
4.13.1.79 #define I_UWRITE_P 0x080
```

## 4.14 include/init.h File Reference

```
#include <arch/x86/elf.h> #include <proc/proc.h> Include dependency graph for init.h:
```



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



## Defines

- #define **ELFHDR** ((struct **elfhdr** \*) 0xA0000000)
- #define **SECTOR** 512

## Functions

- void **Init\_userspace** (**proc\_t** \*)

### 4.14.1 Define Documentation

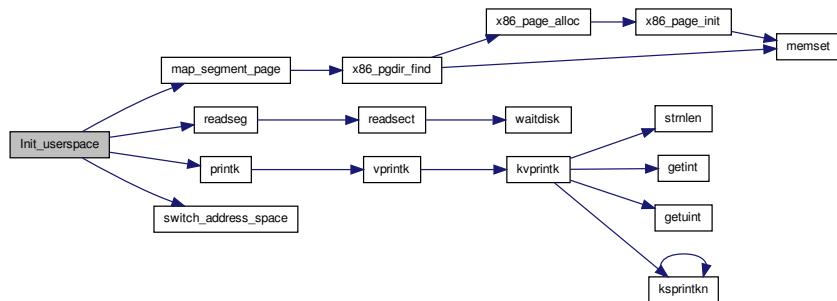
4.14.1.1 #define **ELFHDR** ((struct **elfhdr** \*) 0xA0000000)

4.14.1.2 #define **SECTOR** 512

### 4.14.2 Function Documentation

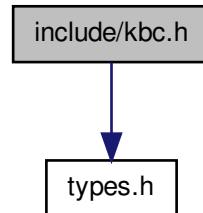
4.14.2.1 void **Init\_userspace** ( **proc\_t** \* )

Here is the call graph for this function:

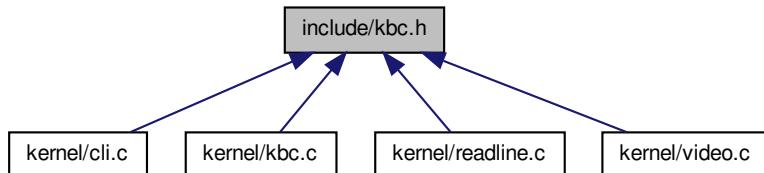


## 4.15 include/kbc.h File Reference

#include <types.h> Include dependency graph for kbc.h:



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



### Defines

- #define KBC\_STATUSPORT 0x64
- #define KBC\_DATAPORT 0x60
- #define KBC\_DATAIN 0x01 /\* New Data in buffer \*/
- #define KBC\_FULLBUF 0x02 /\* Buffer is full \*/
- #define KBC\_REBOOT 0x04 /\* soft reboot \*/
- #define KBC\_COMMAND 0x08 /\* data in output register is a command \*/
- #define KBC\_SECLOCK 0x10 /\* Security lock engaged \*/
- #define KBC\_TTIMEOUT 0x20 /\* transmission timeout error \*/
- #define KBC\_RTIMEOUT 0x40 /\* receive timeout error \*/
- #define KBC\_PARITY 0x80 /\* Parity error \*/
- #define KBC\_READRAM 0x20 /\* Read byte 0 from the internal RAM \*/

- #define **KBC\_READLOW** 0x21 /\* Read byte specified in low 5 bits of the command in 804x's internal RAM \*/
- #define **KBC\_WRITERAM** 0x60 /\* write the data to the address specified in the 5 lower bits of command \*/
- #define **KBC\_DISMOUSE** 0xA7 /\* disable mouse/auxillary port \*/
- #define **KBC\_ENMOUSE** 0xA8 /\* enable mouse/auxillary port \*/
- #define **KBC\_TSTMOUSE** 0xA9 /\* test mouse/auxillary port \*/
- #define **KBC\_SELFTEST** 0xAA /\* initiate the self-test \*/
- #define **KBC\_INTRTEST**
- #define **KBC\_DIGDUMP** 0xAC /\* dump the content of 804x's RAM, output port, input port, status word \*/
- #define **KBC\_DISKBD** 0xAD /\* disable keyboard \*/
- #define **KBC\_ENBKBD** 0xAE /\* enable keyboard \*/
- #define **KBC\_READVER** 0xAF /\* Read keyboard version \*/
- #define **KBC\_INPREAD** 0xC0 /\* Read input port \*/
- #define **KBC\_OUTREAD** 0xD0 /\* Read Output port \*/
- #define **KBC\_OUTWRTE** 0xD1 /\* write output port ,next byte will be written to 804x \*/
- #define **KBC\_OUTKBD** 0xD2 /\* Echo keyboard output buffer \*/
- #define **KBC\_OUTAUX** 0xD3 /\* Echo pointing device output buffer \*/
- #define **KBC\_AUXWRITE** 0xD4 /\* Write to pointing device output buffer \*/
- #define **KBC\_DISA20** 0xDD /\* Disable the A20 gate \*/
- #define **KBC\_ENBA20** 0xDF /\* Enable the A20 gate \*/
- #define **KBC\_PULSE0** 0xFE /\* Pulse output bit 0 \*/
- #define **KBC\_PULSE1** 0xFD /\* ~~~~~ 1 \*/
- #define **KBC\_PULSE2** 0xFB /\* ~~~~~ 2 \*/
- #define **KBC\_PULSE3** 0xF7 /\* ~~~~~ 3 \*/
- #define **KBC\_RESET** 0xFF /\* RESET the mouse and the keyboard \*/
- #define **KBC\_RESEND** 0xFE /\* Resend the last byte \*/
- #define **KBC\_DEFAULT** 0xF6 /\* Set the keyboard to default parameters \*/
- #define **KBC\_DISABLE** 0xF5 /\* disables key scanning, yet set the default parameters \*/
- #define **KBC\_ENABLE** 0xF4 /\* Enable key scanning \*/
- #define **KBC\_TYPEMATIC** 0xF3 /\* Set typematic rate/delay \*/
- #define **KBC\_READID** 0xF2 /\* Read Keyboard ID \*/
- #define **KBC\_SETSCAN** 0xF0 /\* Set/Get scancodes set \*/
- #define **KBC\_ECHODIG** 0xEE /\* request a diagnostic echo from the keyboard \*/
- #define **KBC\_INDICATOR** 0xED /\* Set mode indicators \*/
- #define **KEY\_HOME** 0xE0
- #define **KEY\_END** 0xE1
- #define **KEY\_UP** 0xE2
- #define **KEY\_DN** 0xE3
- #define **KEY\_LF** 0xE4
- #define **KEY\_RT** 0xE5
- #define **KEY\_PGUP** 0xE6
- #define **KEY\_PGDN** 0xE7
- #define **KEY\_INS** 0xE8
- #define **KEY\_DEL** 0xE9

## Functions

- int `kbc_data` (void)
- void `kbc_interrupt` (void)

### 4.15.1 Define Documentation

```
4.15.1.1 #define KBC_AUXWRITE 0xD4 /* Write to pointing device output buffer */

4.15.1.2 #define KBC_COMMAND 0x08 /* data in output register is a command */

4.15.1.3 #define KBC_DATAIN 0x01 /* New Data in buffer */

4.15.1.4 #define KBC_DATAPORT 0x60

4.15.1.5 #define KBC_DEFAULT 0xF6 /* Set the keyboard to default parameters */

4.15.1.6 #define KBC_DIGDUMP 0xAC /* dump the content of 804x's RAM, output port,
    input port, status word */

4.15.1.7 #define KBC_DISA20 0xDD /* Disable the A20 gate */

4.15.1.8 #define KBC_DISABLE 0xF5 /* disables key scanning, yet set the default
    parameters */

4.15.1.9 #define KBC_DISKBD 0xAD /* disable keyboard */

4.15.1.10 #define KBC_DISMOUSE 0xA7 /* disable mouse/auxillary port */

4.15.1.11 #define KBC_ECHODIG 0xEE /* request a diagnostic echo from the keyboard */

4.15.1.12 #define KBC_ENABLE 0xF4 /* Enable key scanning */

4.15.1.13 #define KBC_ENBA20 0xDF /* Enable the A20 gate */

4.15.1.14 #define KBC_ENBKBD 0xAE /* enable keyboard */

4.15.1.15 #define KBC_ENMOUSE 0xA8 /* enable mouse/auxillary port */

4.15.1.16 #define KBC_FULLBUF 0x02 /* Buffer is full */

4.15.1.17 #define KBC_INDICATOR 0xED /* Set mode indicators */

4.15.1.18 #define KBC_INPREAD 0xC0 /* Read input port */

4.15.1.19 #define KBC_INTRTEST
```

**Value:**

0xAB

```
4.15.1.20 #define KBC_OUTAUX 0xD3 /* Echo pointing device output buffer */

4.15.1.21 #define KBC_OUTKBD 0xD2 /* Echo keyboard output buffer */

4.15.1.22 #define KBC_OUTREAD 0xD0 /* Read Output port */

4.15.1.23 #define KBC_OUTWRTE 0xD1 /* write output port ,next byte will be written to
804x */

4.15.1.24 #define KBC_PARITY 0x80 /* Parity error */

4.15.1.25 #define KBC_PULSE0 0xFE /* Pulse output bit 0 */

4.15.1.26 #define KBC_PULSE1 0xFD /* ~~~~~~1 */

4.15.1.27 #define KBC_PULSE2 0xFB /* ~~~~~~2 */

4.15.1.28 #define KBC_PULSE3 0xF7 /* ~~~~~~3 */

4.15.1.29 #define KBC_READID 0xF2 /* Read Keyboard ID */

4.15.1.30 #define KBC_READLOW 0x21 /* Read byte specified in low 5 bits of the command
in 804x's internal RAM */

4.15.1.31 #define KBC_READRAM 0x20 /* Read byte 0 from the internal RAM */

4.15.1.32 #define KBC_READVER 0xAF /* Read keyboard version */

4.15.1.33 #define KBC_REBOOT 0x04 /* soft reboot */

4.15.1.34 #define KBC resend 0xFE /* Resend the last byte */

4.15.1.35 #define KBC_RESET 0xFF /* RESET the mouse and the keyboard */

4.15.1.36 #define KBC_RTIMEOUT 0x40 /* recieve timeout error */

4.15.1.37 #define KBC_SECLOCK 0x10 /* Security lock engaged */

4.15.1.38 #define KBC_SELFTEST 0xAA /* initiate the self-test */

4.15.1.39 #define KBC_SETSCAN 0xF0 /* Set/Get scancodes set */

4.15.1.40 #define KBC_STATUSPORT 0x64
```

4.15.1.41 #define KBC\_TSTMOUSE 0xA9 /\* test mouse/auxillary port \*/

4.15.1.42 #define KBC\_TTIMEOUT 0x20 /\* transmission timeout error \*/

4.15.1.43 #define KBC\_TYPEMATIC 0xF3 /\* Set typematic rate/delay \*/

4.15.1.44 #define KBC\_WRITERAM 0x60 /\* write the data to the address specified in the 5 lower bits of command \*/

4.15.1.45 #define KEY\_DEL 0xE9

4.15.1.46 #define KEY\_DN 0xE3

4.15.1.47 #define KEY\_END 0xE1

4.15.1.48 #define KEY\_HOME 0xE0

4.15.1.49 #define KEY\_INS 0xE8

4.15.1.50 #define KEY\_LF 0xE4

4.15.1.51 #define KEY\_PGDN 0xE7

4.15.1.52 #define KEY\_PGUP 0xE6

4.15.1.53 #define KEY\_RT 0xE5

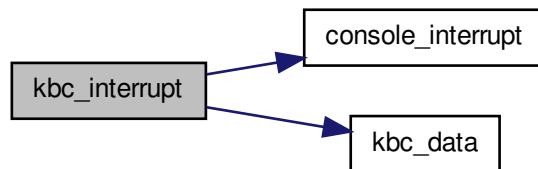
4.15.1.54 #define KEY\_UP 0xE2

## 4.15.2 Function Documentation

4.15.2.1 int kbc\_data( void )

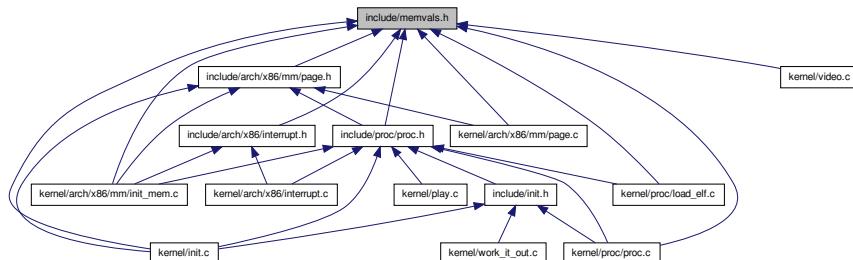
### 4.15.2.2 void kbc\_interrupt( void )

Here is the call graph for this function:



## 4.16 include/memvals.h File Reference

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



## Classes

- struct [Segdesc](#)
- struct [Gdtdesc](#)

## Defines

- #define [SEGACS\\_RW](#) 0x2 /\* Read and write flag Read for code segments and write for data segments \*/
- #define [SEGACS\\_X](#) 0x8 /\* The Executable flag \*/
- #define [SEGACS\\_D](#)

- #define SEGACS\_USR 0x60
- #define SEG\_NULL (struct Segdesc){0,0,0,0,0,0}
- #define SEGMENT(limit, base, access)
- #define EXTMEM 0x100000
- #define PAGESZ 0x1000
- #define PAGELG 0xC
- #define PAGECNT 0x400
- #define PAGETSZ (PAGESZ\*PAGECNT)
- #define KERNEL\_ADDR 0xF0000000
- #define KERNEL\_STACK PAGESZ\*8
- #define VIRTPGT (KERNEL\_ADDR - PAGETSZ)
- #define KERNEL\_STACK\_TOP VIRTPGT
- #define USEREND (VIRTPGT - PAGETSZ)
- #define USERVIRTPGT (USEREND - PAGETSZ)
- #define USERPAGES (USERVIRTPGT - PAGETSZ)
- #define PROC\_LIST (USERPAGES - PAGETSZ)
- #define USERSTART (PROC\_LIST)
- #define USERSTACK\_TOP (USERSTART - 2\*PAGESZ)
- #define USERSPACESTART 0XEE800000
- #define PA(va)
- #define IOPHYMEM 0x0A0000
- #define KB 0x400
- #define MB KB\*KB
- #define GB MB\*KB

## Functions

- struct Gdtdesc \_\_attribute\_\_ ((packed))

## Variables

- uint16\_t size
- uint32\_t base

### 4.16.1 Define Documentation

4.16.1.1 #define EXTMEM 0x100000

4.16.1.2 #define GB MB\*KB

4.16.1.3 #define IOPHYMEM 0x0A0000

## Paging

---

```

4.16.1.4 #define KB 0x400

4.16.1.5 #define KERNEL_ADDR 0xF0000000

4.16.1.6 #define KERNEL_STACK_PAGESZ*8

4.16.1.7 #define KERNEL_STACK_TOP_VIRTPGT

4.16.1.8 #define MB KB*KB

4.16.1.9 #define PA( va )

```

**Value:**

```

({ \
    uint32_t pa = (uint32_t) va; \
    pa-KERNEL_ADDR; \
})

```

```

4.16.1.10 #define PAGECNT 0x400

4.16.1.11 #define PAGELG 0xC

4.16.1.12 #define PAGESZ 0x1000

4.16.1.13 #define PAGETSZ (PAGESZ*PAGECNT)

4.16.1.14 #define PROC_LIST (USERPAGES - PAGETSZ)

4.16.1.15 #define SEG_NULL (struct Segdesc){0,0,0,0,0,0}

4.16.1.16 #define SEGACS_D

```

**Value:**

```

0x4      /* growing down segment for data segments and means that this segments
can be                                         Executed from the user applications
"Low Privilege" */

```

```

4.16.1.17 #define SEGACS_RW 0x2 /* Read and write flag Read for code segments and write
for data segments */

4.16.1.18 #define SEGACS_USR 0x60

4.16.1.19 #define SEGACS_X 0x8 /* The Executable flag */

```

---

4.16.1.20 #define SEGMENT( *limit*, *base*, *access* )

**Value:**

```
(struct Segdesc) \
{ \
    ((limit>>12) & 0xFFFF, \
     (base) & 0xFFFF, \
     ((base)>>16) & 0xFF, \
     (access) | 0x90, \
     ((limit) >> 28), \
     0xC, \
     (base)>>24 \
}
```

4.16.1.21 #define USEREND (VIRTPGT - PAGETSZ)

4.16.1.22 #define USERPAGES (USERVIRTPGT - PAGETSZ)

4.16.1.23 #define USERSPACESTART 0XEE800000

4.16.1.24 #define USERSTACK\_TOP (USERSTART - 2\*PAGESZ)

4.16.1.25 #define USERSTART (PROC\_LIST)

4.16.1.26 #define USERVIRTPGT (USEREND - PAGETSZ)

4.16.1.27 #define VIRTPGT (KERNEL\_ADDR - PAGETSZ)

## 4.16.2 Function Documentation

4.16.2.1 struct Gdtdesc \_\_attribute\_\_ ( (packed) )

## 4.16.3 Variable Documentation

4.16.3.1 uint32\_t base

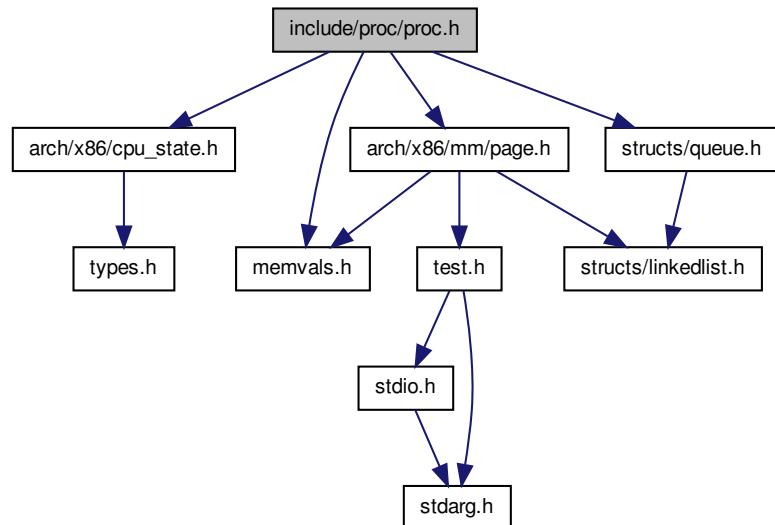
4.16.3.2 uint16\_t size

## 4.17 include/proc/proc.h File Reference

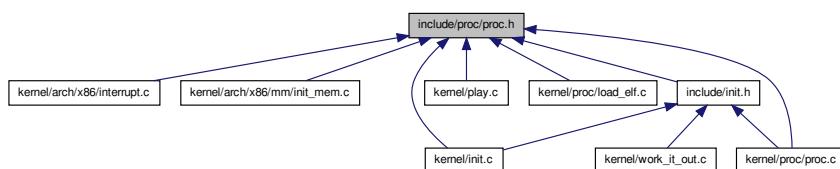
---

```
#include <arch/x86/cpu_state.h>      #include <memvals.h> x
#include <structs/queue.h>    #include <arch/x86/mm/page.-
```

h> Include dependency graph for proc.h:



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



## Classes

- struct [proc](#)

## Defines

- #define [MAX\\_PROCS](#) 256
- #define [MAX\\_PROC\\_NAME](#) 128
- #define [RUNNABLE](#) 0
- #define [PROC\\_EMPTY](#) 0x1

- #define NON\_RUNNABLE 0x2
- #define PROC\_TABLE\_SIZE ROUND\_UP(MAX\_PROCS \* sizeof(struct proc), - PAGESZ)

## TypeDefs

- typedef struct proc proc\_t

## Functions

- LIST\_HEAD (Proc\_List, proc)
- LIFO\_HEAD (Proc\_Lifo, proc)
- void init\_proc\_table (void)
- void switch\_address\_space (proc\_t \*)
- uint32\_t proc\_setup (proc\_t \*\*)
- uint32\_t proc\_setup\_mem (proc\_t \*)
- void test\_lifo (void)

## Variables

- proc\_t \* proc\_table
- struct Proc\_Lifo running\_procs

### 4.17.1 Define Documentation

4.17.1.1 #define MAX\_PROC\_NAME 128

4.17.1.2 #define MAX\_PROCS 256

include/arch/x86/interrupt.h CATReloaded (C) Copyrights 2011 <http://ecatreload.net>

#### Date

4 Oct, 2012

4.17.1.3 #define NON\_RUNNABLE 0x2

4.17.1.4 #define PROC\_EMPTY 0x1

4.17.1.5 #define PROC\_TABLE\_SIZE ROUND\_UP(MAX\_PROCS \* sizeof(struct proc), PAGESZ)

4.17.1.6 #define RUNNABLE 0

## 4.17.2 Typedef Documentation

### 4.17.2.1 `typedef struct proc proc_t`

## 4.17.3 Function Documentation

### 4.17.3.1 `void init_proc_table( void )`

Here is the call graph for this function:



### 4.17.3.2 `LIFO_HEAD( Proc_Lifo , proc )`

### 4.17.3.3 `LIST_HEAD( Proc_List , proc )`

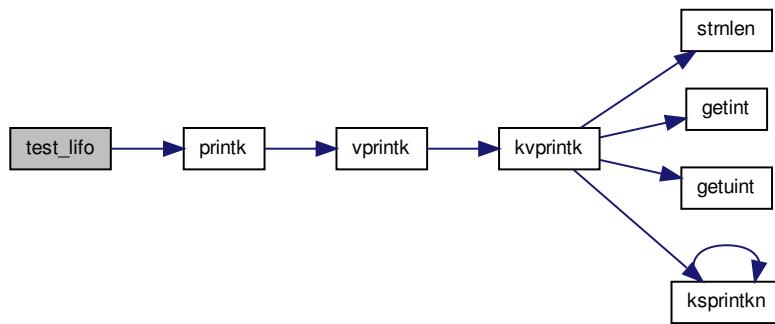
### 4.17.3.4 `uint32_t proc_setup( proc_t** )`

### 4.17.3.5 `uint32_t proc_setup_mem( proc_t* )`

### 4.17.3.6 `void switch_address_space( proc_t* )`

### 4.17.3.7 `void test_lifo( void )`

Here is the call graph for this function:



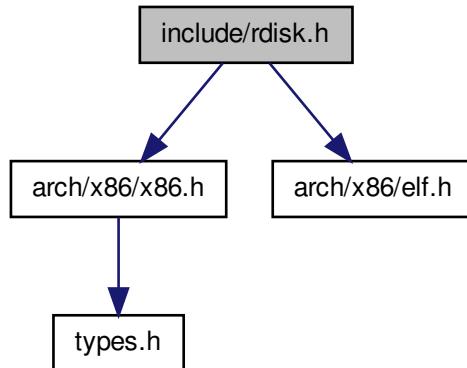
#### 4.17.4 Variable Documentation

##### 4.17.4.1 proc\_t\* proc\_table

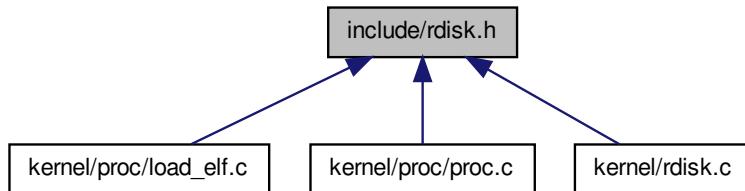
##### 4.17.4.2 struct Proc\_Lifo running\_procs

### 4.18 include/rdisk.h File Reference

```
#include <arch/x86/x86.h> #include <arch/x86/elf.h> Include
dependency graph for rdisk.h:
```



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



## Defines

- #define **SECTOR** 512
- #define **ELF\_MAGIC2** 0x8ec031fc

## Functions

- void **readsect** (void \*, uint32\_t)
- void **readseg** (uint32\_t, uint32\_t, uint32\_t)
- void **waitdisk** (void)

### 4.18.1 Define Documentation

4.18.1.1 #define **ELF\_MAGIC2** 0x8ec031fc

4.18.1.2 #define **SECTOR** 512

### 4.18.2 Function Documentation

4.18.2.1 void **readsect** ( void \*, uint32\_t )

Here is the call graph for this function:



#### 4.18.2.2 void readseg( uint32\_t, uint32\_t, uint32\_t )

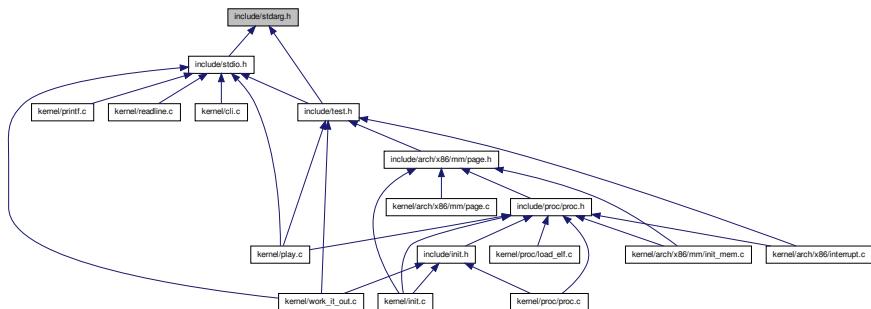
Here is the call graph for this function:



#### 4.18.2.3 void waitdisk( void )

## 4.19 include/stdarg.h File Reference

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



## Defines

- `#define __va_size(type) (((sizeof(type) + sizeof(int)-1)/sizeof(int))*sizeof(int))`
  - `#define va_start(ap, last) ((ap) = (va_list)&(last) + __va_size(last))`
  - `#define va_arg(ap, type) (*(type *)((ap) += __va_size(type), (ap) - __va_size(type)))`
  - `#define va_end(ap)`

## TypeDefs

- `typedef char * va list`

### 4.19.1 Define Documentation

4.19.1.1 `#define __va_size( type ) (((sizeof(type) + sizeof(int)-1)/sizeof(int))*sizeof(int))`

4.19.1.2 `#define va_arg( ap, type ) (*(type *)((ap) += __va_size(type), (ap) - __va_size(type)))`

4.19.1.3 `#define va_end( ap )`

4.19.1.4 `#define va_start( ap, last ) ((ap) = (va_list)&(last) + __va_size(last))`

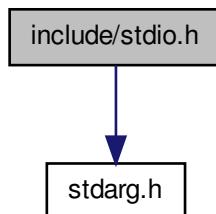
### 4.19.2 Typedef Documentation

4.19.2.1 `typedef char* va_list`

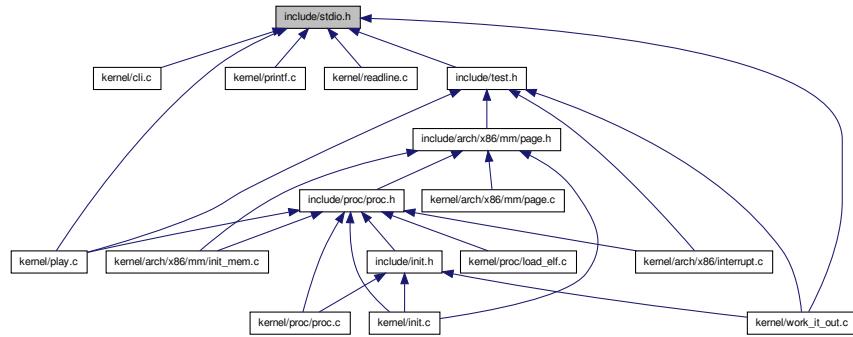
This code is a used in both linux and BSD

## 4.20 include/stdio.h File Reference

`#include <stdarg.h>` Include dependency graph for stdio.h:



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



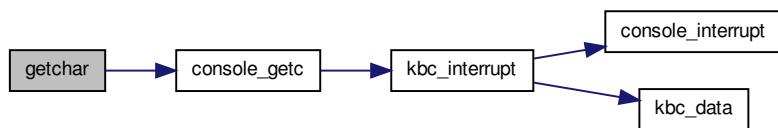
## Functions

- void [putchr](#) (int)
- int [getchar](#) (void)
- int [printf](#) (const char \*,...)
- int [vprintf](#) (const char \*, va\_list)
- int [kvprintf](#) (const char \*, void(\*func)(int, int \*), int \*count, va\_list ap)
- void [ksprintkn](#) (void(\*func)(int, int \*), int \*count, uintmax\_t num, int base, int width, int padc)
- char \* [readline](#) (const char \*)

### 4.20.1 Function Documentation

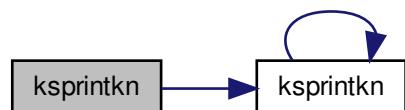
#### 4.20.1.1 int [getchar](#) ( void )

Here is the call graph for this function:



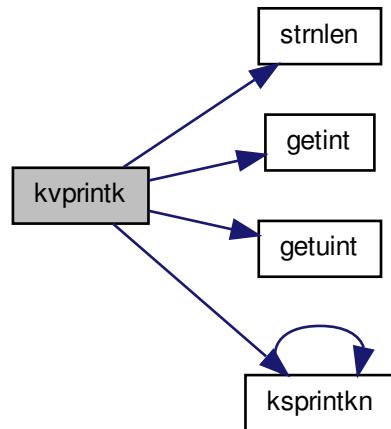
4.20.1.2 void ksprintkn ( void(\*)(int, int \*) *func*, int \* *count*, uintmax\_t *num*, int *base*, int *width*, int *padc* )

Here is the call graph for this function:



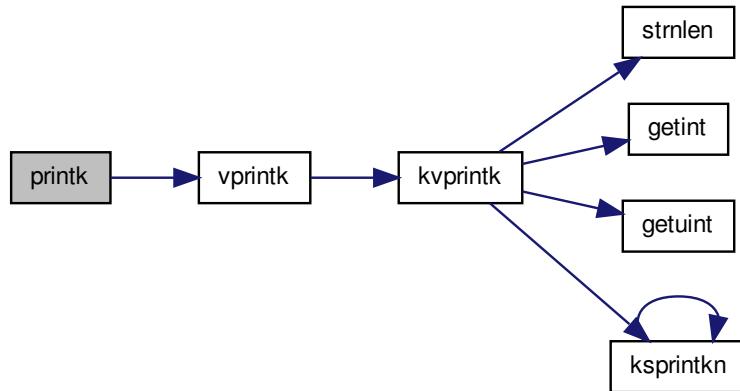
4.20.1.3 int kvprintf ( const char \*, void(\*)(int, int \*) *func*, int \* *count*, va\_list *ap* )

Here is the call graph for this function:

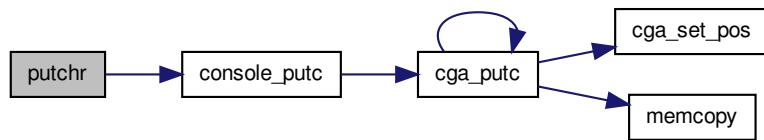


**4.20.1.4 int printk ( const char \*, ... )**

Here is the call graph for this function:

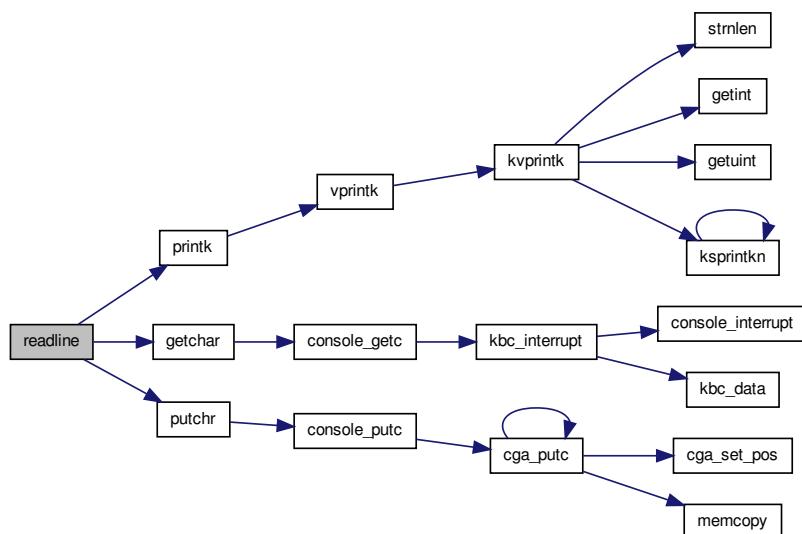
**4.20.1.5 void putchr ( int )**

Here is the call graph for this function:



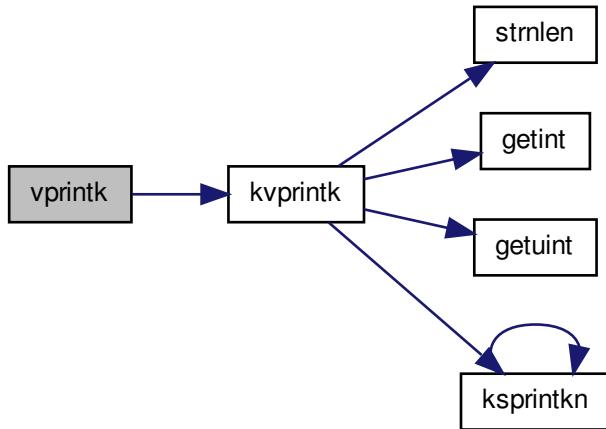
4.20.1.6 `char* readline( const char * )`

Here is the call graph for this function:



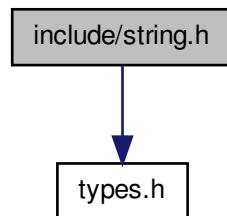
#### 4.20.1.7 int vprintfk ( const char \* , va\_list )

Here is the call graph for this function:

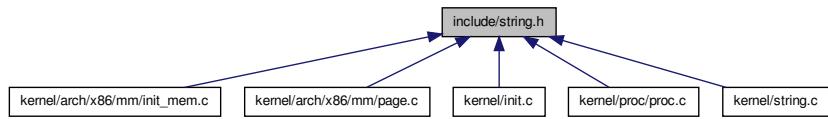


#### 4.21 include/string.h File Reference

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



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



## Functions

- `char * strcpy (char *dst, char *src)`
- `char * strncpy (char *dst, char *src, uint32_t count)`
- `char * strlcpy (char *dst, char *src, uint32_t count)`
- `char * strcat (char *dst, char *src)`
- `char * strncat (char *dst, char *src, uint32_t count)`
- `char * strlcat (char *dst, char *src, uint32_t count)`
- `int strcmp (const char *cmp1, const char *cmp2)`
- `int strncmp (const char *cmp1, const char *cmp2, uint32_t count)`
- `char * strchr (const char *str, int c)`
- `char * strchr (const char *str, int c)`
- `char * strnchr (const char *str, uint32_t count, int c)`
- `uint32_t strlen (const char *str)`
- `uint32_t strnlen (const char *str, uint32_t count)`
- `void * memset (void *dst, int c, uint32_t count)`
- `void * memcpy (void *dst, const void *src, uint32_t count)`
- `void * memmove (void *dst, const void *src, uint32_t count)`
- `void * memcmp (const void *cmp1, const void *cmp2, uint32_t count)`

### 4.21.1 Function Documentation

4.21.1.1 `void* memcmp ( const void * cmp1, const void * cmp2, uint32_t count )`

4.21.1.2 `void* memcpy ( void * dst, const void * src, uint32_t count )`

4.21.1.3 `void* memmove ( void * dst, const void * src, uint32_t count )`

4.21.1.4 `void* memset ( void * dst, int c, uint32_t count )`

4.21.1.5 `char* strcat ( char * dst, char * src )`

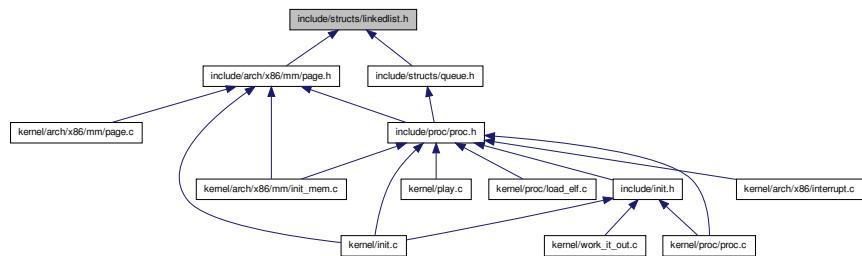
4.21.1.6 `char* strchr ( const char * str, int c )`

4.21.1.7 `int strcmp ( const char * cmp1, const char * cmp2 )`

- 4.21.1.8 `char* strcpy ( char * dst, char * src )`
- 4.21.1.9 `char* strlcat ( char * dst, char * src, uint32_t count )`
- 4.21.1.10 `char* strlcpy ( char * dst, char * src, uint32_t count )`
- 4.21.1.11 `uint32_t strlen ( const char * str )`
- 4.21.1.12 `char* strncat ( char * dst, char * src, uint32_t count )`
- 4.21.1.13 `char* strnchr ( const char * str, uint32_t count, int c )`
- 4.21.1.14 `int strncmp ( const char * cmp1, const char * cmp2, uint32_t count )`
- 4.21.1.15 `char* strncpy ( char * dst, char * src, uint32_t count )`
- 4.21.1.16 `uint32_t strnlen ( const char * str, uint32_t count )`
- 4.21.1.17 `char* strrchr ( const char * str, int c )`

## 4.22 include/structs/linkedlist.h File Reference

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



## Defines

- `#define LIST_HEAD(Instance, Type)`
- `#define LIST_HEAD_INIT(head) {NULL}`
- `#define LIST_ENTRY(type)`
- `#define LIST_EMPTY(xhead) ((xhead)->first == NULL)`
- `#define LIST_FIRST(xhead) ((xhead)->first)`
- `#define LIST_NEXT(element, field) ((element)->field.next)`
- `#define LIST_FOREACH(var, head, field)`
- `#define LIST_INIT(head)`

- #define LIST\_INSERT\_AFTER(pre, element, field)
- #define LIST\_INSERT\_BEFORE(post, element, field)
- #define LIST\_REMOVE(element, field)
- #define LIST\_INSERT\_HEAD(head, element, field)

### 4.22.1 Define Documentation

4.22.1.1 #define LIST\_EMPTY( *xhead* ) ((*xhead*)>first == NULL)

Functions of the list, first,next, insert, empty..etc

4.22.1.2 #define LIST\_ENTRY( *type* )

**Value:**

```
struct {\n    struct type *next;\n    struct type **prev;\n}
```

4.22.1.3 #define LIST\_FIRST( *xhead* ) ((*xhead*)>first)

4.22.1.4 #define LIST\_FOREACH( *var*, *head*, *field* )

**Value:**

```
for ((var) = LIST_FIRST((head));\n     (var);\n     (var) = LIST_NEXT((var), field))
```

For each loop on teh list

4.22.1.5 #define LIST\_HEAD( *Instance*, *Type* )

**Value:**

```
struct Instance{\n    struct Type *first;\n}
```

a boot time list is a macro coded list to provide ability to make sequential lists at boot time.

4.22.1.6 #define LIST\_HEAD\_INIT( *head* ) {NULL}

4.22.1.7 #define LIST\_INIT( *head* )

**Value:**

```
do (\n    LIST_FIRST((head)) = NULL; \  
}while(0)
```

#### 4.22.1.8 #define LIST\_INSERT\_AFTER( pre, element, field )

**Value:**

```
do(\n    if((LIST_NEXT((element),field) = (LIST_NEXT((pre),field))) != NULL) \  
        LIST_NEXT(pre,field)->field.prev = &(LIST_NEXT((element),field)\n    ); \  
    LIST_NEXT((pre),field)->field.next = (element); \  
    (element)->field.prev = &(LIST_NEXT((pre),field)); \  
} while(0)
```

#### 4.22.1.9 #define LIST\_INSERT\_BEFORE( post, element, field )

**Value:**

```
do(\n    (element)->field.prev = (post)->field.prev; \  
    LIST_NEXT((element),field) = (post); \  
    *(post)->field.prev = (element); \  
    (post)->field.prev = &(LIST_NEXT((element),field)); \  
} while(0)
```

#### 4.22.1.10 #define LIST\_INSERT\_HEAD( head, element, field )

**Value:**

```
do (\n    if((LIST_NEXT((element), field) = LIST_FIRST((head))) != NULL) \  
        LIST_FIRST((head))->field.prev = &(LIST_NEXT((element),field)); \  
    \n    LIST_FIRST((head)) = (element); \  
    (element)->field.prev = &(LIST_FIRST((head))); \  
} while(0)
```

#### 4.22.1.11 #define LIST\_NEXT( element, field ) ((element)->field.next)

#### 4.22.1.12 #define LIST\_REMOVE( element, field )

**Value:**

```
do (\n    if(LIST_NEXT((element),field) != NULL) \  
        LIST_NEXT((element),field)->field.prev = \  
            (element)->field.prev; \  
        *(element)->field.prev = LIST_NEXT( (element), field); \  
    }while(0)
```

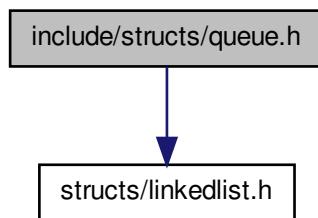
## 4.23 include/structs/list.h File Reference

### Classes

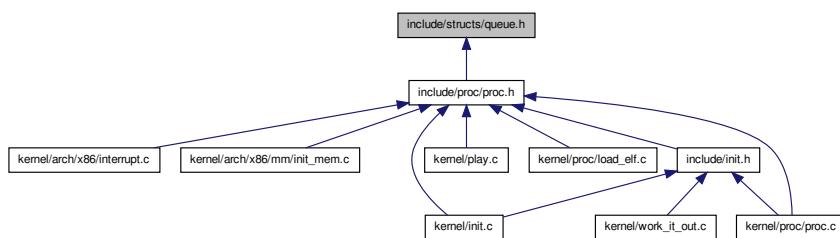
- struct `element`

## 4.24 include/structs/queue.h File Reference

```
#include <structs/linkedlist.h> Include dependency graph for queue.h:
```



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



### Defines

- #define `LIFO_HEAD(name, member)`
- #define `LIFO_HEAD_INIT(h) LIST_INIT(h)`
- #define `LIFO_ENTRY(entry) LIST_ENTRY(entry)`
- #define `LIFO_FIRST(h) ((h)->first)`

- #define LIFO\_INIT(h) LIFO\_FIRST((h)) = NULL
- #define LIFO\_PUSH(h, element, field) LIST\_INSERT\_HEAD(h, element, field)
- #define LIFO\_POP(h, field)

#### 4.24.1 Define Documentation

4.24.1.1 #define LIFO\_ENTRY( *entry* ) LIST\_ENTRY(*entry*)

4.24.1.2 #define LIFO\_FIRST( *h* ) ((*h*)->first)

4.24.1.3 #define LIFO\_HEAD( *name*, *member* )

**Value:**

```
struct name{           \
    struct member *first;   \
}
```

4.24.1.4 #define LIFO\_HEAD\_INIT( *h* ) LIST\_INIT(*h*)

4.24.1.5 #define LIFO\_INIT( *h* ) LIFO\_FIRST((*h*))=NULL

4.24.1.6 #define LIFO\_POP( *h*, *field* )

**Value:**

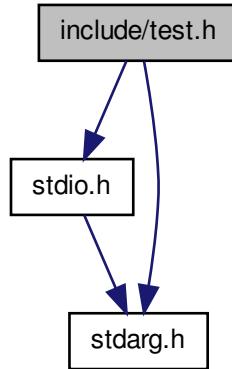
```
LIST_FIRST((h)); \
LIST_REMOVE(LIST_FIRST((h)), field)
```

4.24.1.7 #define LIFO\_PUSH( *h*, *element*, *field* ) LIST\_INSERT\_HEAD(*h*, *element*, *field*)

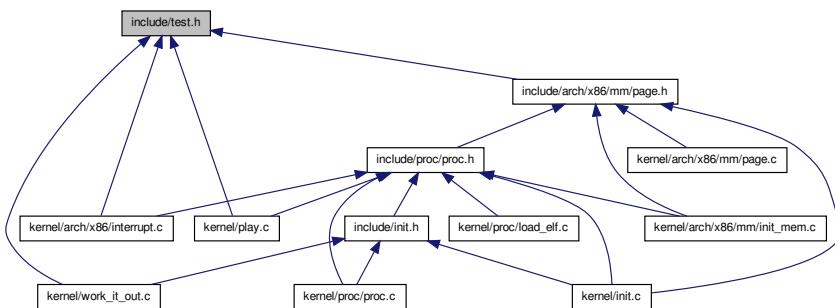
#### 4.25 include/test.h File Reference

```
#include <stdio.h> #include <stdarg.h> Include dependency graph
```

for test.h:



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



## Defines

- #define **panic**(...)
- #define **assert**(x)

## Functions

- void **\_panic\_** (const char \*, int, const char \*,...) **\_attribute\_**((noreturn))

### 4.25.1 Define Documentation

#### 4.25.1.1 #define assert( x )

**Value:**

```
if(! (x)) {           \
    panic("Assertion Failed %s\n", #x); \
}
```

#### 4.25.1.2 #define panic( ... )

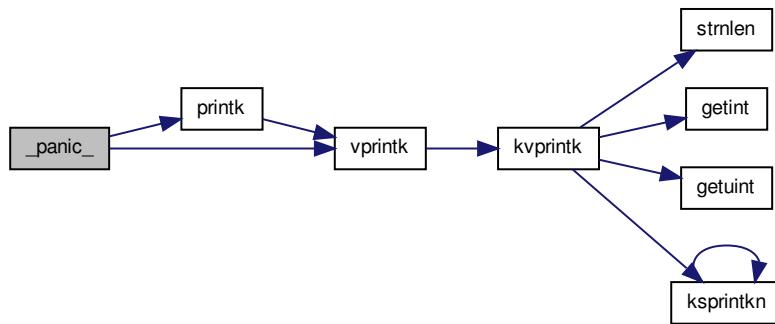
**Value:**

```
printk(">>>> KERNEL PANIC <<<<<\n"); \
_panic_(__FILE__, __LINE__, __VA_ARGS__)
```

### 4.25.2 Function Documentation

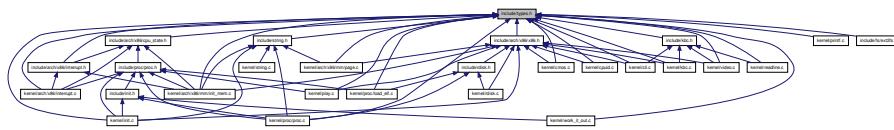
#### 4.25.2.1 void \_panic\_( const char \* , int , const char \* , ... )

Here is the call graph for this function:



## 4.26 include/types.h File Reference

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



### Defines

- #define **NULL** ((void\*) 0)
- #define **ROUND\_DOWN**(x, y)
- #define **ROUND\_UP**(x, y)

### Typedefs

- typedef unsigned int **uint32\_t**
- typedef unsigned short **uint16\_t**
- typedef unsigned char **uint8\_t**
- typedef unsigned long **uint64\_t**
- typedef char **int8\_t**
- typedef int **int32\_t**
- typedef short **int16\_t**
- typedef long long **int64\_t**
- typedef unsigned int **size\_t**
- typedef **uint64\_t uintmax\_t**
- typedef **int64\_t intmax\_t**
- typedef **uint32\_t reg\_t**
- typedef **uint32\_t paddr\_t**
- typedef **uint32\_t vaddr\_t**

#### 4.26.1 Define Documentation

##### 4.26.1.1 #define NULL ((void\*) 0)

##### 4.26.1.2 #define ROUND\_DOWN( x, y )

#### Value:

```
({
    uint32_t z = (uint32_t)(x); \
    (typeof(x)) (z - z % (y)); \
})
```

4.26.1.3 #define ROUND\_UP( x, y )

**Value:**

```
({\ \
    uint32_t z = (uint32_t)(y); \
    (typeof(x)) (ROUND_DOWN(x,y) + y); \
})
```

## 4.26.2 Typedef Documentation

4.26.2.1 `typedef short int16_t`

4.26.2.2 `typedef int int32_t`

4.26.2.3 `typedef long long int64_t`

4.26.2.4 `typedef char int8_t`

4.26.2.5 `typedef int64_t intmax_t`

4.26.2.6 `typedef uint32_t paddr_t`

4.26.2.7 `typedef uint32_t reg_t`

4.26.2.8 `typedef unsigned int size_t`

4.26.2.9 `typedef unsigned short uint16_t`

4.26.2.10 `typedef unsigned int uint32_t`

4.26.2.11 `typedef unsigned long uint64_t`

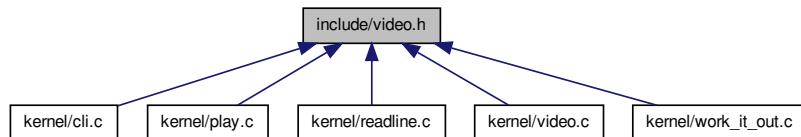
4.26.2.12 `typedef unsigned char uint8_t`

4.26.2.13 `typedef uint64_t uintmax_t`

4.26.2.14 `typedef uint32_t vaddr_t`

## 4.27 include/video.h File Reference

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



### Defines

- #define CGA\_INDEX1 0x3D4
- #define CGA\_DATA1 0x3D5
- #define CGA\_BUFF\_OFF 0xB8000
- #define CGA\_COLS 80
- #define CGA\_ROWS 25
- #define CGA\_SIZE (CGA\_ROWS \* CGA\_COLS)
- #define COLOR\_DARK\_GRAY 0x0700
- #define COLOR\_BLUE 0x0100
- #define COLOR\_GREEN 0x0200
- #define COLOR\_RED 0x0400
- #define COLOR\_GRAY 0x0800
- #define COLOR\_WHITE 0x0f00
- #define BACKGROUND\_GRAY 0x7000
- #define BACKGROUND\_BLUE 0x1000
- #define BACKGROUND\_GREEN 0x2000
- #define BACKGROUND\_RED 0x4000
- #define BACKGROUND\_BLINK 0x8000
- #define BACKGROUND\_WHITE 0xF000

### Functions

- void `cga_init` (void)
- `uint16_t cga_get_pos` (void)
- void `cga_set_pos` (`uint16_t`)
- void `cga_putc` (int c)
- void `cga_set_attr` (`uint16_t`)

## Variables

- `uint16_t cursor_position`
- `uint16_t * char_buff`

### 4.27.1 Define Documentation

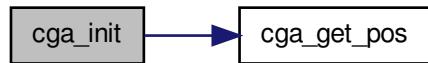
```
4.27.1.1 #define BACKGROUND_BLINK 0x8000  
4.27.1.2 #define BACKGROUND_BLUE 0x1000  
4.27.1.3 #define BACKGROUND_GRAY 0x7000  
4.27.1.4 #define BACKGROUND_GREEN 0x2000  
4.27.1.5 #define BACKGROUND_RED 0x4000  
4.27.1.6 #define BACKGROUND_WHITE 0xF000  
4.27.1.7 #define CGA_BUFF_OFF 0xB8000  
4.27.1.8 #define CGA_COLS 80  
4.27.1.9 #define CGA_DATA1 0x3D5  
4.27.1.10 #define CGA_INDEX1 0x3D4  
4.27.1.11 #define CGA_ROWS 25  
4.27.1.12 #define CGA_SIZE (CGA_ROWS * CGA_COLS)  
4.27.1.13 #define COLOR_BLUE 0x0100  
4.27.1.14 #define COLOR_DARK_GRAY 0x0700  
4.27.1.15 #define COLOR_GRAY 0x0800  
4.27.1.16 #define COLOR_GREEN 0x0200  
4.27.1.17 #define COLOR_RED 0x0400  
4.27.1.18 #define COLOR_WHITE 0x0f00
```

### 4.27.2 Function Documentation

```
4.27.2.1 uint16_t cga_get_pos( void )
```

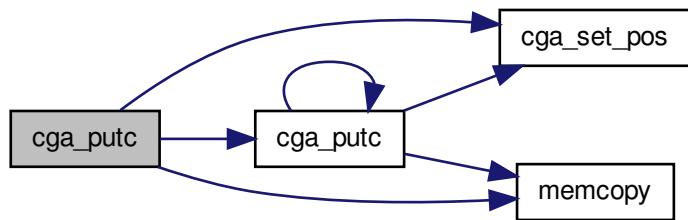
#### 4.27.2.2 void cga\_init( void )

Here is the call graph for this function:



#### 4.27.2.3 void cga\_putc( int c )

Here is the call graph for this function:



#### 4.27.2.4 void cga\_set\_attr( uint16\_t )

#### 4.27.2.5 void cga\_set\_pos( uint16\_t )

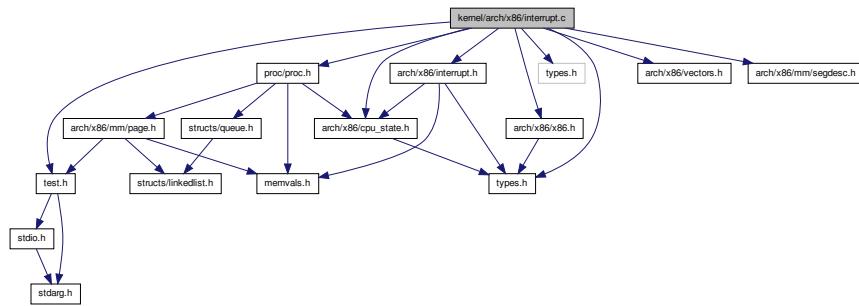
### 4.27.3 Variable Documentation

#### 4.27.3.1 uint16\_t\* char\_buff

#### 4.27.3.2 uint16\_t cursor\_position

## 4.28 kernel/arch/x86/interrupt.c File Reference

```
#include <types.h> #include <test.h> #include <proc/proc.h>
#include <arch/x86/x86.h> #include <arch/x86/interrupt.h>
#include <arch/x86/vectors.h> #include <arch/x86/mm/segdesc.h>
#include <arch/x86/cpu_state.h> Include dependency graph for
interrupt.c:
```



### Functions

- void `trap` (void)
- void `idt_init` (void)
- void `register_exception` (uint32\_t index, char \*name, uint16\_t present, void(handler)(void))
- void `interrupt_init` (void)
- void `map_exception` (uint32\_t int\_index, cpu\_state\_t \*cpu\_state)
- uint32\_t `page_fault_handler` (cpu\_state\_t \*cpu\_state)

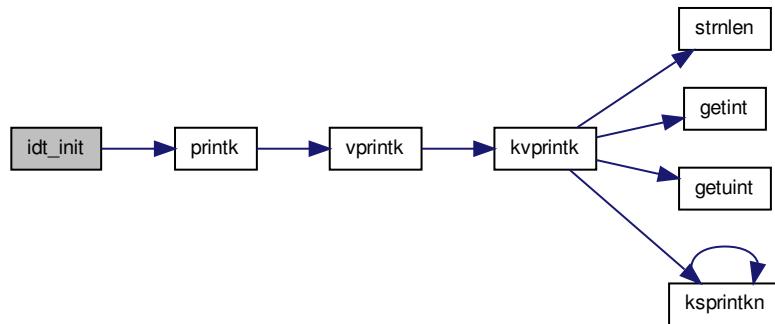
### Variables

- gatedesc `idt` [64]
- char \* `x86_exception_names` []
- `int_generic`

#### 4.28.1 Function Documentation

#### 4.28.1.1 void idt\_init( void )

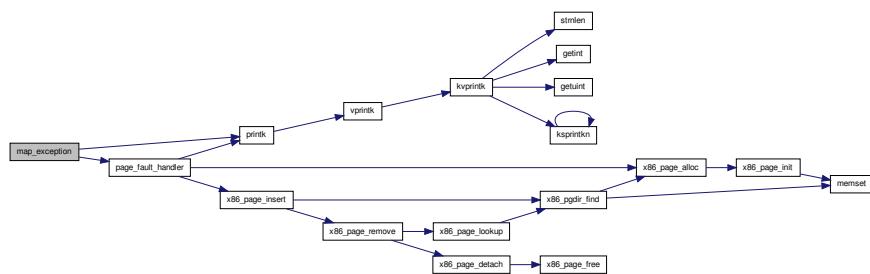
Here is the call graph for this function:



#### 4.28.1.2 void interrupt\_init( void )

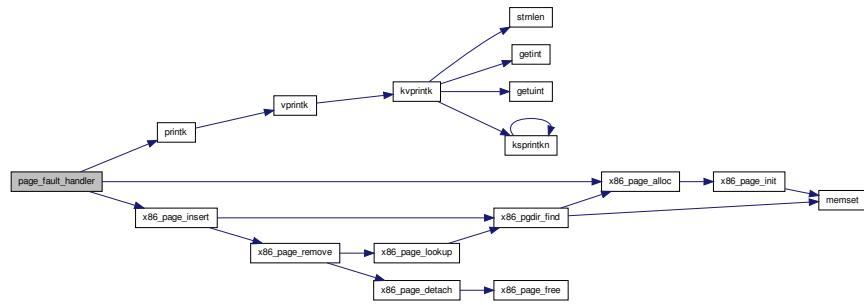
#### 4.28.1.3 void map\_exception( uint32\_t int\_index, cpu\_state\_t \*cpu\_state )

Here is the call graph for this function:



#### 4.28.1.4 uint32\_t page\_fault\_handler ( cpu\_state\_t \*cpu\_state )

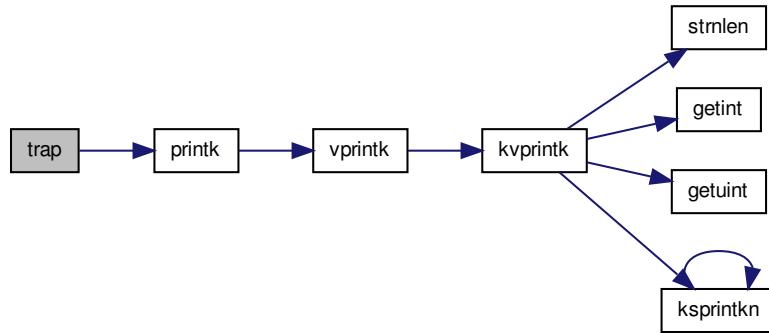
Here is the call graph for this function:



#### 4.28.1.5 void register\_exception ( uint32\_t index, char \* name, uint16\_t present, void(handler)(void) )

#### 4.28.1.6 void trap ( void )

Here is the call graph for this function:



## 4.28.2 Variable Documentation

**4.28.2.1 gatedesc idt**

[include/arch/x86/interrupt.h](#) CATReloaded (C) Copyrights 2011 <http://catreload.net>

**Date**

27 Sept, 2012

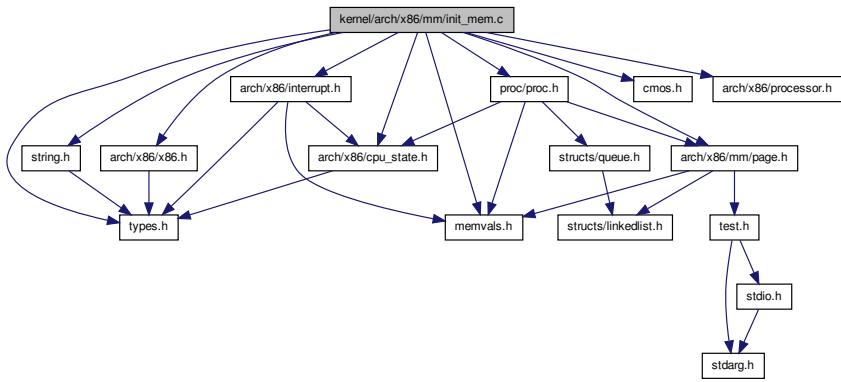
**4.28.2.2 int\_generic****4.28.2.3 char\* x86\_exception\_names[]****Initial value:**

```
{  
    "Divide Error #DE",  
    "Debug",  
    "NMI",  
    "Breakpoint #BP",  
    "Overflow #OV",  
    "Bound Range Exceeded #BR",  
    "Undefined Opcode",  
    "Device not available",  
    "Double fault",  
    "Coprocessor segment overrun",  
    "Invalid TSS",  
    "Segment not present",  
    "Stack Segment Fault",  
    "General Protection",  
    "Page Fault",  
    "\0",  
    "x87 FPU error",  
    "Alignment Mask",  
    "Machine Check",  
    "SIMD exception"  
}
```

**4.29 kernel/arch/x86/mm/init\_mem.c File Reference**

```
#include <types.h>  #include <arch/x86/x86.h>  #include  
<memvals.h> #include <cmos.h> #include <string.h> #include  
<arch/x86/processor.h>  #include <arch/x86/mm/page.h> x  
#include <arch/x86/interrupt.h>  #include <arch/x86/cpu-  
_state.h> #include <proc/proc.h> Include dependency graph for init_-
```

mem.c:



## Functions

- void `scan_memory` (void)
- void `init_tss` (void)
- void `x86_setup_memory` (void)

## Variables

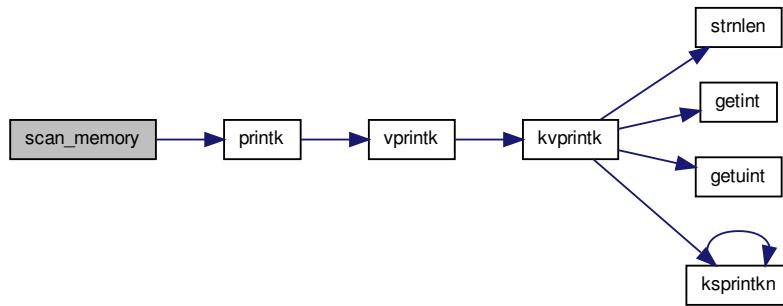
- `uint32_t max_addr`
- `char * next_free = 0`
- `struct Page * pages`
- `proc_t * proc_table`
- `struct Segdesc catgdt []`
- `struct Gdtdesc gdtdesc`
- `Idtdesc idtdesc`
- `tss_t tss`

### 4.29.1 Function Documentation

#### 4.29.1.1 void init\_tss( void )

## 4.29.1.2 void scan\_memory ( void )

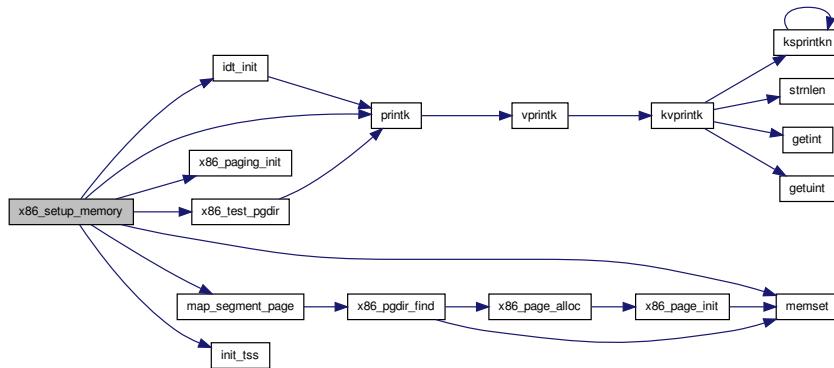
Here is the call graph for this function:



## 4.29.1.3 void x86\_setup\_memory ( void )

At this point paging is on

Here is the call graph for this function:



## 4.29.2 Variable Documentation

## 4.29.2.1 struct Segdesc catgdt[]

**Initial value:**

```
{
    SEG_NULL,
    [1] = SEGMENT(0xffffffff, 0, SEGACS_RW | SEGACS_X),
    [2] = SEGMENT(0xffffffff, 0, SEGACS_RW),
    [3] = SEGMENT(0xffffffff, 0x000000, SEGACS_RW | SEGACS_USR | SEGACS_X),
    [4] = SEGMENT(0xffffffff, 0x0, SEGACS_RW | SEGACS_USR),
    [5] = SEG_NULL
}
```

#### 4.29.2.2 struct Gdtdesc gdtdesc

**Initial value:**

```
{
    sizeof(catgdt)-1,
    (unsigned long) catgdt
}
```

#### 4.29.2.3 Idtdesc idtdesc

**Initial value:**

```
{
    (256*sizeof(gatedesc))-1,
    (unsigned long) idt
}
```

#### 4.29.2.4 uint32\_t max\_addr

#### 4.29.2.5 char\* next\_free = 0

#### 4.29.2.6 struct Page\* pages

#### 4.29.2.7 proc\_t\* proc\_table

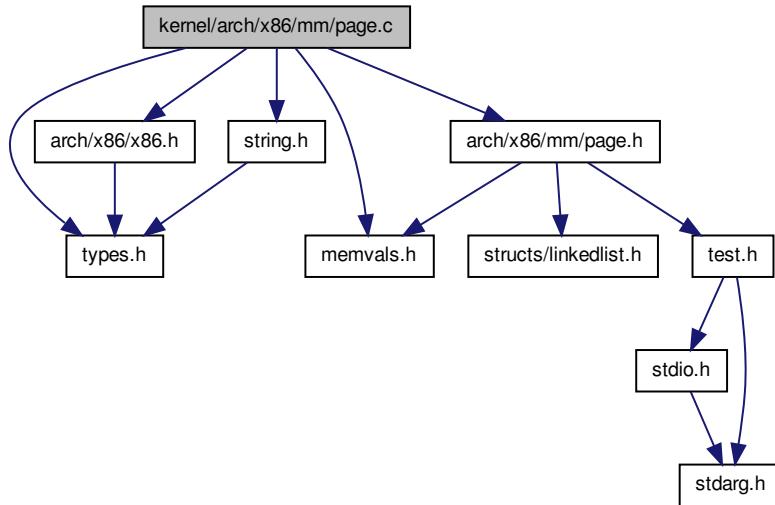
#### 4.29.2.8 tss\_t tss

### 4.30 kernel/arch/x86/mm/page.c File Reference

---

```
#include <types.h>    #include <arch/x86/x86.h>    #include
```

```
<memvals.h> #include <string.h> #include <arch/x86/mm/page.h>
Include dependency graph for page.c:
```



## Functions

- void `x86_paging_init` (void)
- void `x86_page_init` (struct `Page` \*page)
- int `x86_page_alloc` (struct `Page` \*\*page\_byref)
- void `x86_page_free` (struct `Page` \*page)
- void `x86_page_detach` (struct `Page` \*page)
- `pte_t` \* `x86_pgdir_find` (`pde_t` \*pgdir, const void \*va, int allocate)
- struct `Page` \* `x86_page_lookup` (`pde_t` \*pgdir, void \*va, `pte_t` \*\*pte)
- void `x86_page_remove` (`pde_t` \*pgdir, void \*va)
- int `x86_page_insert` (`pde_t` \*pgdir, struct `Page` \*page, void \*va, `uint32_t` perm)
- void `map_segment_page` (`pde_t` \*pgdir, `vaddr_t` linear, `size_t` size, `paddr_t` physical, int perm)
- void `x86_test_pgdir` (void)

## Variables

- `uint32_t` `page_count`
- `pde_t` \* `global_pgdir`
- `uint32_t` `global_cr3`
- `char` \* `next_free`
- struct `Page` \* `pages`

### 4.30.1 Function Documentation

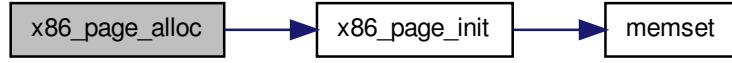
4.30.1.1 `void map_segment_page( pde_t *pgdir, vaddr_t linear, size_t size, paddr_t physical, int perm )`

Here is the call graph for this function:



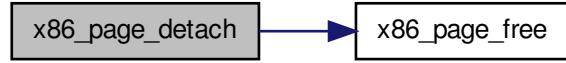
4.30.1.2 `int x86_page_alloc( struct Page **page_byref )`

Here is the call graph for this function:



4.30.1.3 `void x86_page_detach( struct Page *page )`

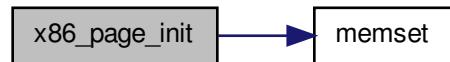
Here is the call graph for this function:



4.30.1.4 void x86\_page\_free ( struct Page \* page )

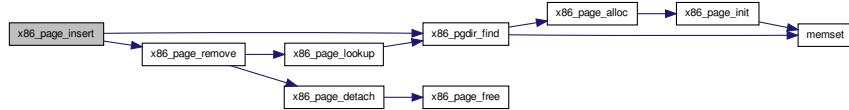
4.30.1.5 void x86\_page\_init ( struct Page \* page )

Here is the call graph for this function:



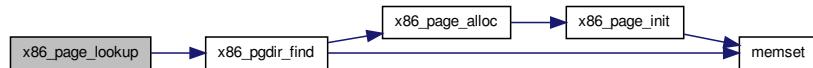
4.30.1.6 int x86\_page\_insert ( pde\_t \* pgdir, struct Page \* page, void \* va, uint32\_t perm )

Here is the call graph for this function:



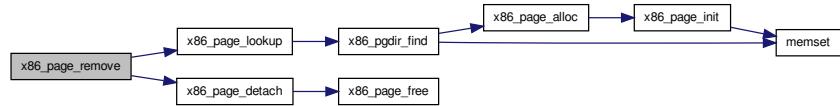
4.30.1.7 struct Page\* x86\_page\_lookup ( pde\_t \* pgdir, void \* va, pte\_t \*\* pte )  
[read]

Here is the call graph for this function:



#### 4.30.1.8 void x86\_page\_remove ( pde\_t \* pgdir, void \* va )

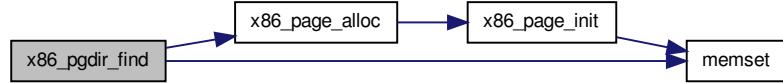
Here is the call graph for this function:



#### 4.30.1.9 void x86\_paging\_init ( void )

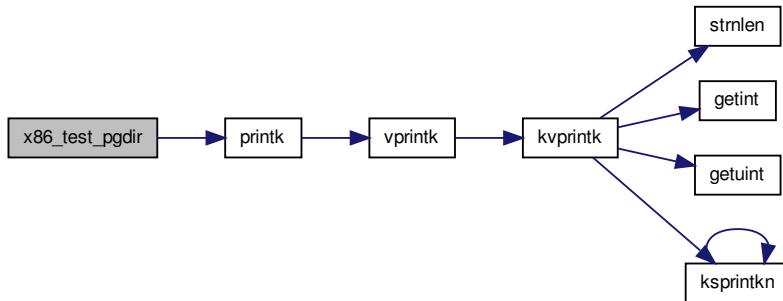
#### 4.30.1.10 pte\_t\* x86\_pgdir\_find ( pde\_t \* pgdir, const void \* va, int allocate )

Here is the call graph for this function:



#### 4.30.1.11 void x86\_test\_pgdir ( void )

Here is the call graph for this function:



### 4.30.2 Variable Documentation

4.30.2.1 `uint32_t global_cr3`

4.30.2.2 `pde_t* global_pgdir`

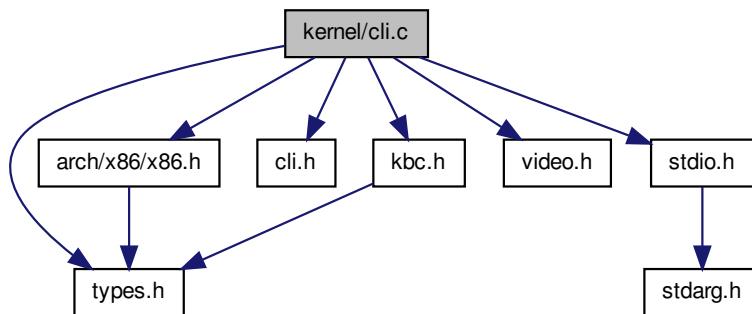
4.30.2.3 `char* next_free`

4.30.2.4 `uint32_t page_count`

4.30.2.5 `struct Page* pages`

## 4.31 kernel/cli.c File Reference

```
#include <types.h>    #include <arch/x86/x86.h>    #include
<cli.h>    #include <kbc.h>    #include <video.h>    #include
<stdio.h> Include dependency graph for cli.c:
```



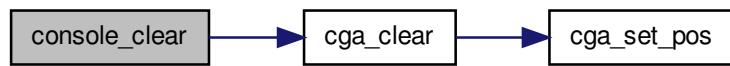
## Functions

- `void console_init (void)`
- `void console_interrupt (int(*intr)(void))`
- `void console_clear (void)`
- `int console_getc (void)`
- `void console_putc (int c)`
- `void putchr (int c)`
- `int getchar (void)`

### 4.31.1 Function Documentation

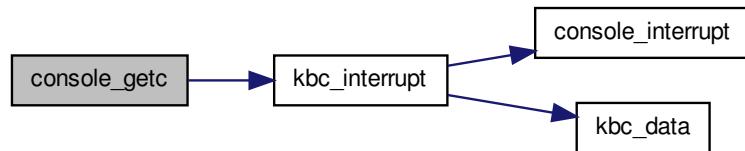
#### 4.31.1.1 void `console_clear( void )`

Here is the call graph for this function:



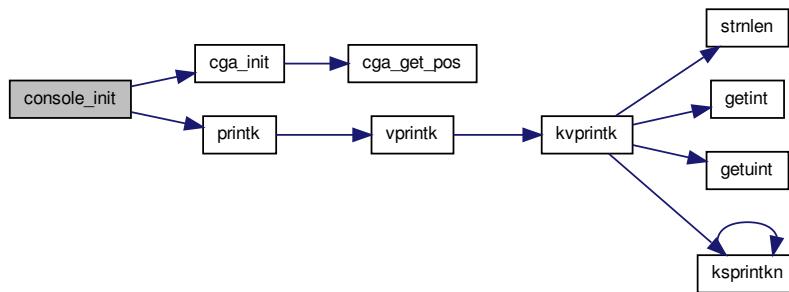
#### 4.31.1.2 int `console_getc( void )`

Here is the call graph for this function:



#### 4.31.1.3 void console\_init( void )

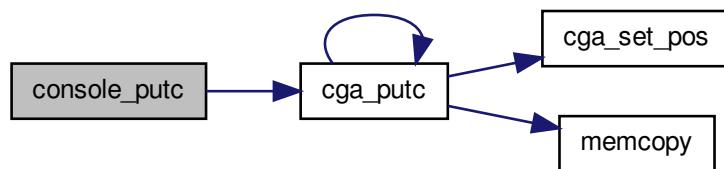
Here is the call graph for this function:



#### 4.31.1.4 void console\_interrupt( int(\*)(void) intr )

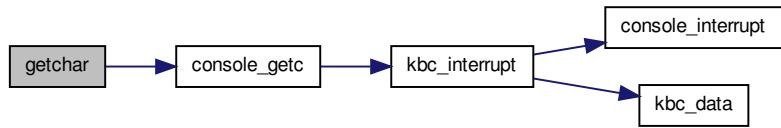
#### 4.31.1.5 void console\_putc( int c )

Here is the call graph for this function:

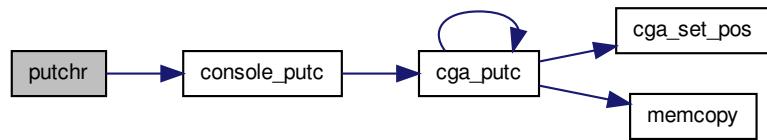


**4.31.1.6 int getchar ( void )**

Here is the call graph for this function:

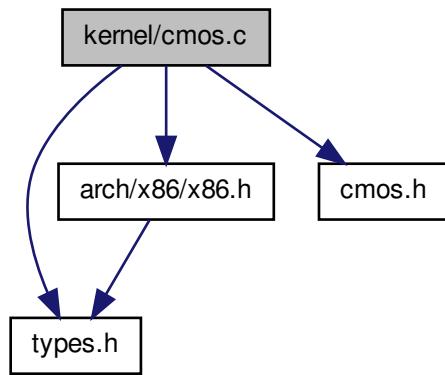
**4.31.1.7 void putchr ( int c )**

Here is the call graph for this function:



## 4.32 kernel/cmos.c File Reference

```
#include <types.h>    #include <arch/x86/x86.h>    #include  
<cmos.h> //Include dependency graph for cmos.c:
```



### Functions

- `uint8_t cmos_set_power_stat (uint8_t stat)`
- `uint32_t cmos_get_reg (uint8_t value)`
- `uint8_t cmos_get_time (uint8_t value)`

#### 4.32.1 Function Documentation

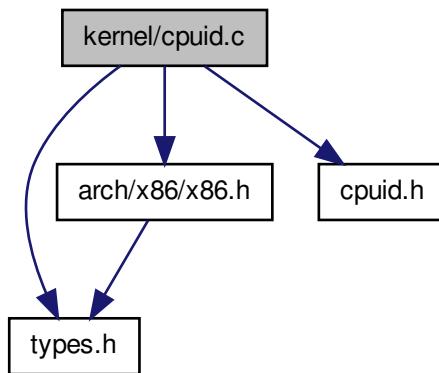
4.32.1.1 `uint32_t cmos_get_reg ( uint8_t value )`

4.32.1.2 `uint8_t cmos_get_time ( uint8_t value )`

4.32.1.3 `uint8_t cmos_set_power_stat ( uint8_t stat )`

### 4.33 kernel/cpuid.c File Reference

```
#include <types.h>    #include <arch/x86/x86.h>    #include
<cpuid.h> Include dependency graph for cpuid.c:
```



#### Classes

- struct [cpuid\\_regs](#)

#### Functions

- [uint32\\_t cpuid\\_get\\_ebx \(void\)](#)
- [uint32\\_t cpuid\\_get\\_ecx \(void\)](#)
- [uint32\\_t cpuid\\_get\\_edx \(void\)](#)
- [uint32\\_t cpuid\\_get\\_eax \(void\)](#)
- [uint32\\_t set\\_eax \(uint32\\_t value\)](#)
- [uint32\\_t cpuid\\_print \(uint32\\_t ax\\_value, uint32\\_t operation\)](#)

#### 4.33.1 Function Documentation

4.33.1.1 [uint32\\_t cpuid\\_get\\_eax \( void \)](#)

4.33.1.2 [uint32\\_t cpuid\\_get\\_ebx \( void \)](#)

4.33.1.3 [uint32\\_t cpuid\\_get\\_ecx \( void \)](#)

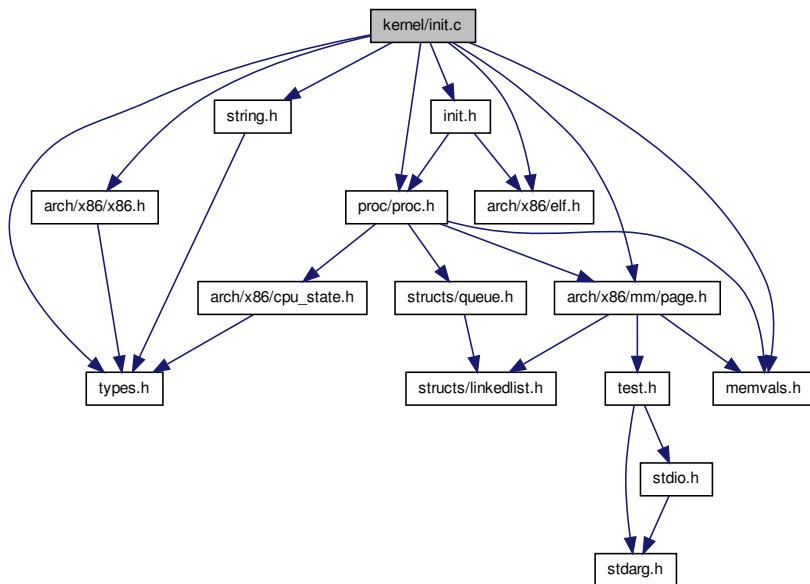
4.33.1.4 `uint32_t cpuid_get_edx( void )`

4.33.1.5 `uint32_t cpuid_print( uint32_t ax_value, uint32_t operation )`

4.33.1.6 `uint32_t set_eax( uint32_t value )`

## 4.34 kernel/init.c File Reference

```
#include <types.h>    #include <arch/x86/x86.h>    #include
<memvals.h> #include <string.h> #include <arch/x86/mm/page.h>    #include <arch/x86/elf.h>    #include <proc/proc.h> x
#include <init.h> Include dependency graph for init.c:
```



## Functions

- `void Init_userspace (proc_t *proc)`

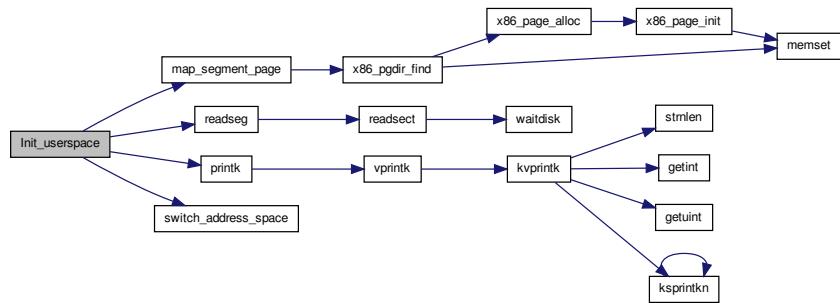
## Variables

- `pde_t * global_pgdir`
- `struct Segdesc catgdt []`
- `proc_t processes []`

#### 4.34.1 Function Documentation

##### 4.34.1.1 void Init\_userspace ( proc\_t \* proc )

Here is the call graph for this function:



#### 4.34.2 Variable Documentation

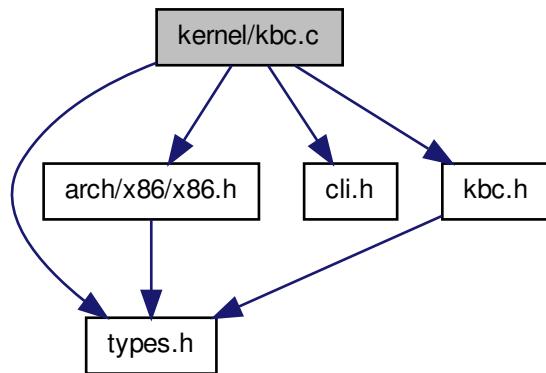
##### 4.34.2.1 struct Segdesc catgdt[]

##### 4.34.2.2 pde\_t\* global\_pgdir

##### 4.34.2.3 proc\_t processes[]

## 4.35 kernel/kbc.c File Reference

```
#include <types.h>    #include <arch/x86/x86.h>    #include  
<cli.h> #include <kbc.h> Include dependency graph for kbc.c:
```



### Defines

- #define **ESCODE** (1<<6)
- #define **CTL** (1<<1)
- #define **SHIFT** (1<<0)
- #define **ALT** (1<<2)
- #define **CAPSLOCK** (1<<3)
- #define **NUMLOCK** (1<<4)
- #define **SCROLLLOCK** (1<<5)
- #define **NUL** 0
- #define **CL(x)** ((x)-'@')

### Functions

- int **kbc\_data** (void)
- void **kbc\_interrupt** (void)

#### 4.35.1 Define Documentation

##### 4.35.1.1 #define ALT (1<<2)

4.35.1.2 #define CAPSLOCK (1<<3)

4.35.1.3 #define CL( x ) ((x)-'@')

4.35.1.4 #define CTL (1<<1)

4.35.1.5 #define ESCODE (1<<6)

4.35.1.6 #define NUL 0

4.35.1.7 #define NUMLOCK (1<<4)

4.35.1.8 #define SCROLLLOCK (1<<5)

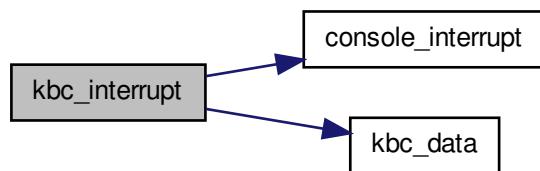
4.35.1.9 #define SHIFT (1<<0)

## 4.35.2 Function Documentation

4.35.2.1 int kbc\_data ( void )

4.35.2.2 void kbc\_interrupt ( void )

Here is the call graph for this function:

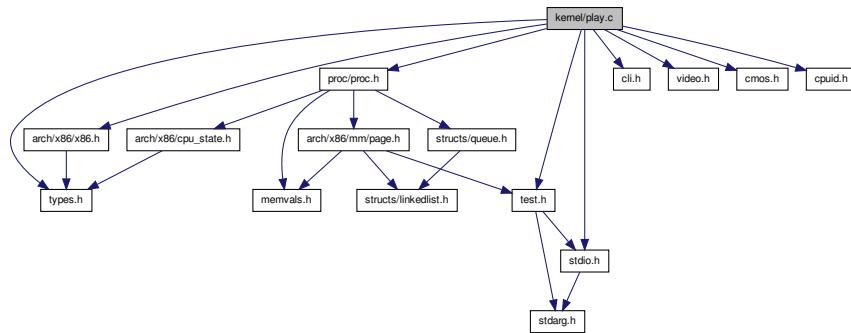


## 4.36 kernel/play.c File Reference

### kernel main file

```
#include <types.h> #include <cli.h> #include <stdio.h>
#include <video.h> #include <cmos.h> #include <arch/x86/x86.h> #include <cpuid.h> #include <proc/proc.h> #include
```

<test.h> Include dependency graph for play.c:



## Example API Actions

Example actions available.

This API provides certain actions as an example.

### Parameters

in	repeat	Number of times to do nothing.
----	--------	--------------------------------

### Return values

TRUE	Successfully did nothing.
FALSE	Oops, did something.

### Example Usage:

```
example_nada(3); // Do nothing 3 times.
```

- void [play](#) (void)
- void [\\_panic](#) (const char \*file, int nline, const char \*fmt,...)
- void [time\\_print](#) (void)
- void [bootup](#) (void)

## 4.36.1 Detailed Description

kernel main file Saad Talaat

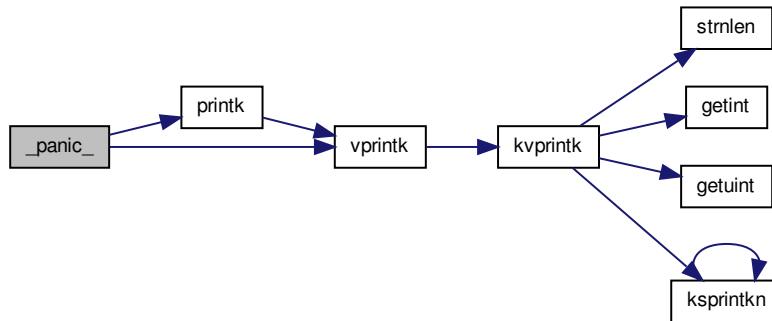
### Date

12/9/2011 Description goes here

## 4.36.2 Function Documentation

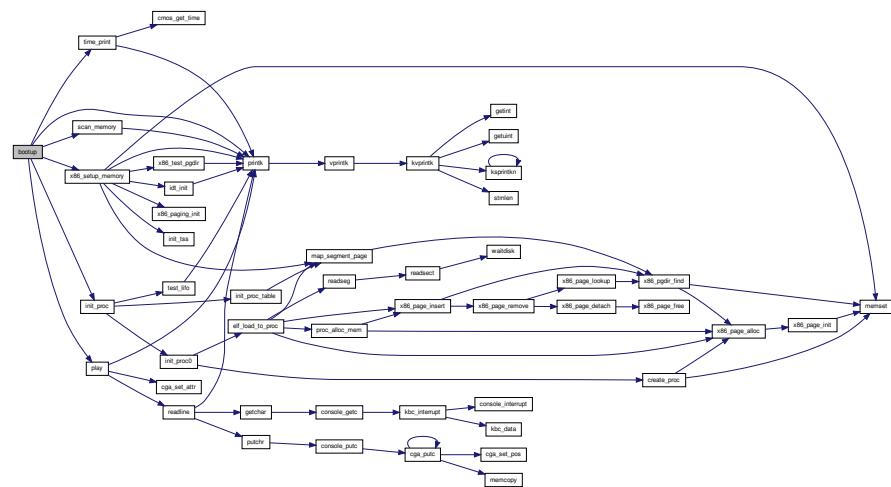
4.36.2.1 void \_panic\_( const char \*file, int nline, const char \*fmt, ... )

Here is the call graph for this function:



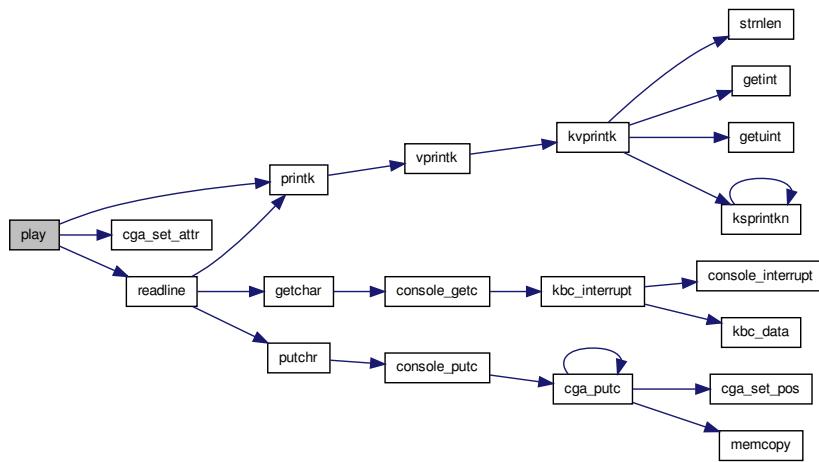
#### 4.36.2.2 void bootup ( void )

Here is the call graph for this function:



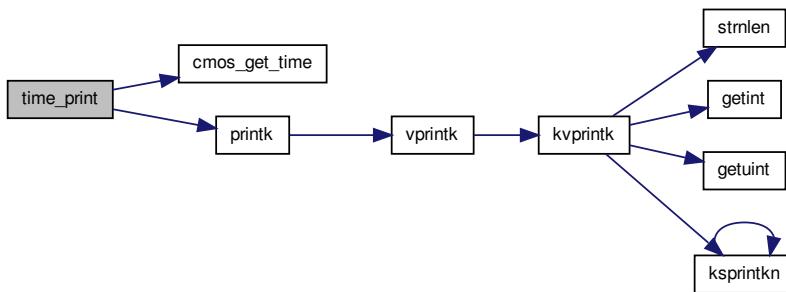
### 4.36.2.3 void play ( void )

Here is the call graph for this function:



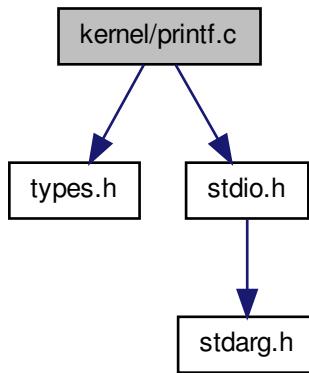
### 4.36.2.4 void time\_print ( void )

Here is the call graph for this function:



## 4.37 kernel/printf.c File Reference

```
#include <types.h> #include <stdio.h> Include dependency graph for
printf.c:
```



### Defines

- `#define hex2ascii(x) ("0123456789ABCDEF"[x])`

### Functions

- `void ksprintkn (void(*func)(int, int *), int *count, uintmax_t num, int base, int width, int padc)`
- `int getint (va_list *ap, int lflag)`
- `int getuint (va_list *ap, int lflag)`
- `int kvprintk (const char *format, void(*func)(int, int *), int *count, va_list ap)`
- `vprintk (const char *format, va_list ap)`
- `int printk (const char *format,...)`

#### 4.37.1 Define Documentation

4.37.1.1 `#define hex2ascii( x ) ("0123456789ABCDEF"[x])`

#### 4.37.2 Function Documentation

4.37.2.1 `int getint ( va_list * ap, int lflag )`

4.37.2.2 `int getuint ( va_list * ap, int lflag )`

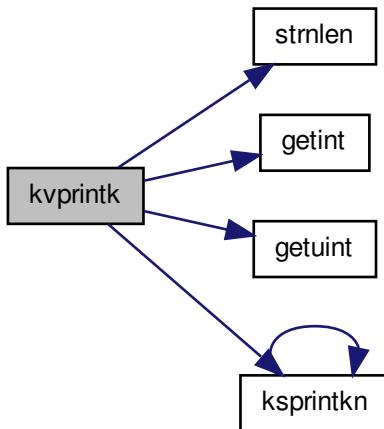
4.37.2.3 `void ksprintkn ( void(*)(int, int *) func, int * count, uintmax_t num, int base, int width, int padc )`

Here is the call graph for this function:



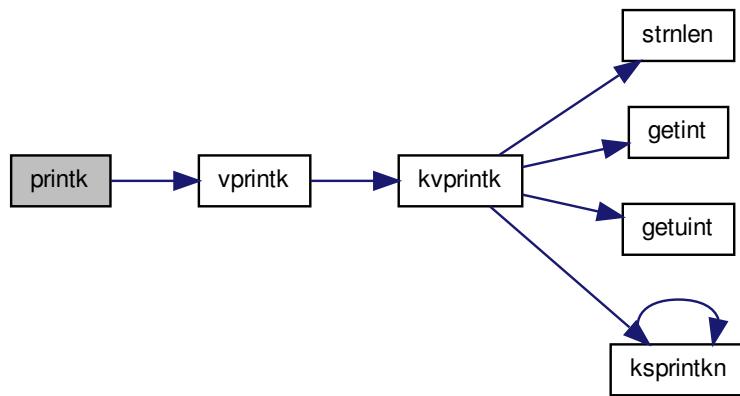
4.37.2.4 `int kvprintk ( const char * format, void(*)(int, int *) func, int * count, va_list ap )`

Here is the call graph for this function:



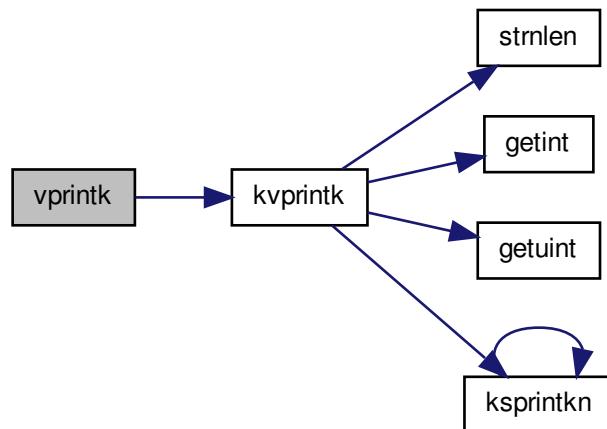
**4.37.2.5 int printk ( const char \* *format*, ... )**

Here is the call graph for this function:



4.37.2.6 vprintfk ( const char \* *format*, va\_list *ap* )

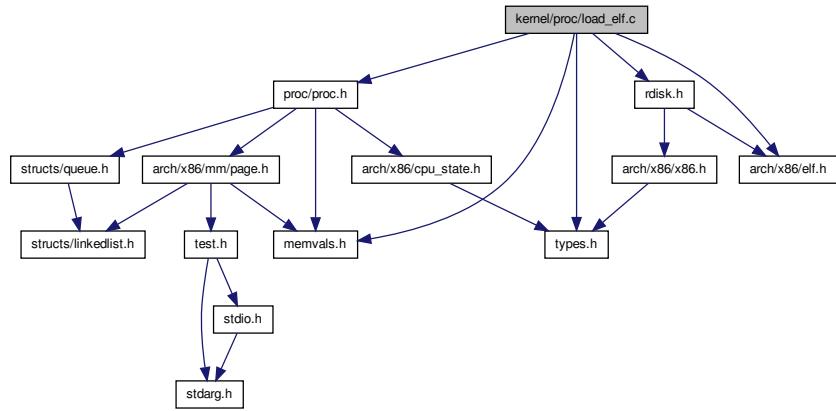
Here is the call graph for this function:



## 4.38 kernel/proc/load\_elf.c File Reference

```
#include <types.h>  #include <arch/x86/elf.h>  #include
<memvals.h>  #include <proc/proc.h>  #include <rdisk.h> x
```

Include dependency graph for load\_elf.c:



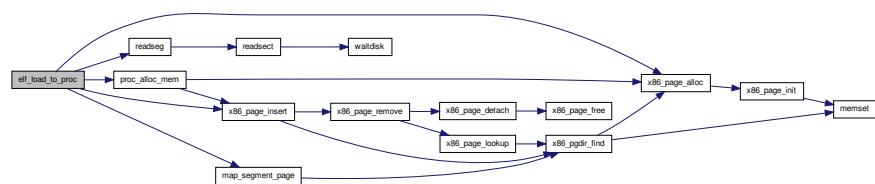
## Functions

- `uint32_t elf_load_to_proc (proc_t *proc, uint32_t offset)`

### 4.38.1 Function Documentation

#### 4.38.1.1 `uint32_t elf_load_to_proc ( proc_t *proc, uint32_t offset )`

Here is the call graph for this function:



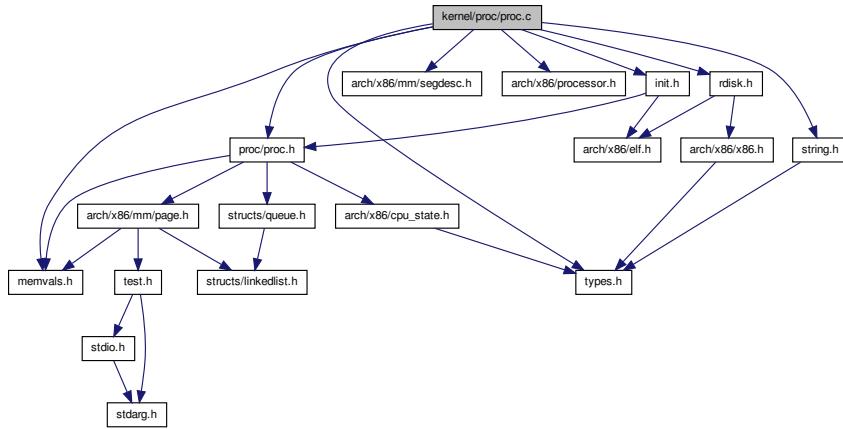
## 4.39 kernel/proc/proc.c File Reference

```

#include <types.h> #include <memvals.h> #include <proc/proc.h>
#include <arch/x86/mm/segdesc.h> #include <arch/x86/processor.h>
#include <string.h> #include <init.h> #include <rdisk.h>

```

h> Include dependency graph for proc.c:



## Functions

- void `init_proc_table` (void)
- `uint32_t create_proc (proc_t **proc_s)`
- `uint32_t proc_alloc_mem (proc_t *proc, void *va, uint32_t len)`
- `uint32_t init_proc0 ()`
- void `init_proc` (void)
- void `test_lifo` (void)
- void `switch_address_space (proc_t *proc_to_run)`

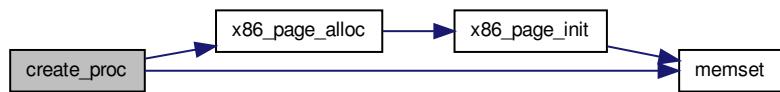
## Variables

- `pde_t * global_pgdir`
- struct `Proc_List` `empty_procs`
- `proc_t * proc_table`
- struct `Proc_Lifo` `running_procs`

### 4.39.1 Function Documentation

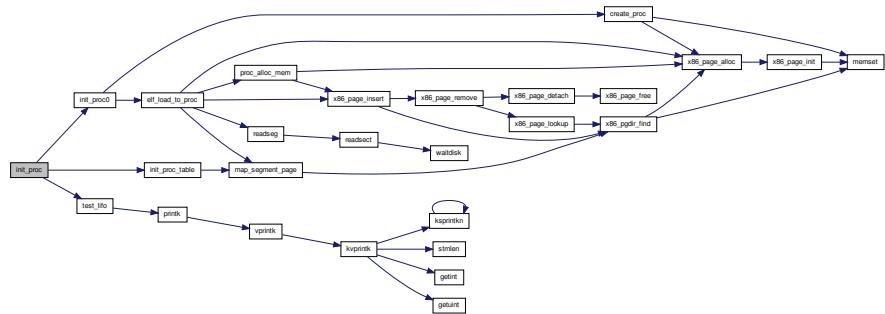
#### 4.39.1.1 uint32\_t create\_proc( proc\_t \*\*proc\_s )

Here is the call graph for this function:



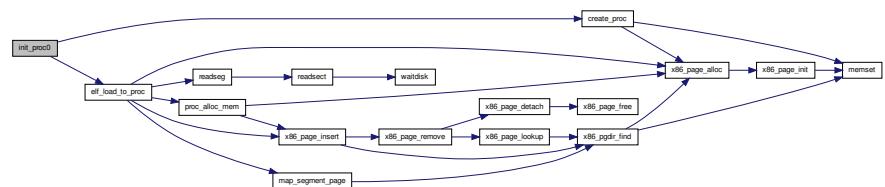
#### 4.39.1.2 void init\_proc( void )

Here is the call graph for this function:



#### 4.39.1.3 uint32\_t init\_proc0( )

Here is the call graph for this function:



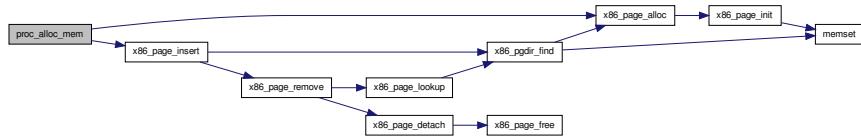
## 4.39.1.4 void init\_proc\_table ( void )

Here is the call graph for this function:



## 4.39.1.5 uint32\_t proc\_alloc\_mem ( proc\_t \* proc, void \* va, uint32\_t len )

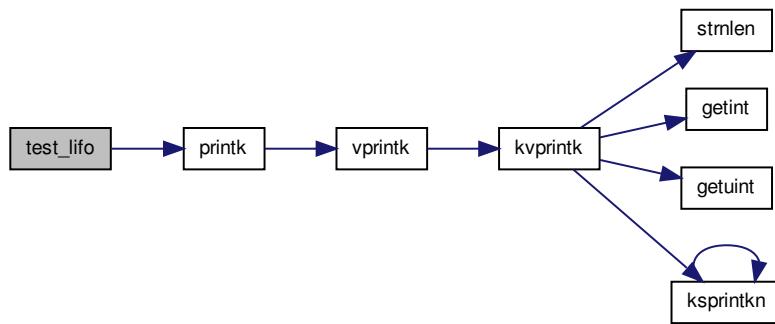
Here is the call graph for this function:



## 4.39.1.6 void switch\_address\_space ( proc\_t \* proc\_to\_run )

## 4.39.1.7 void test\_lifo ( void )

Here is the call graph for this function:



### 4.39.2 Variable Documentation

4.39.2.1 `struct Proc_List empty_procs`

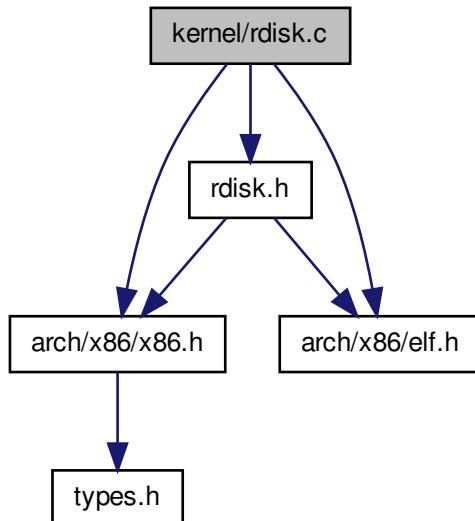
4.39.2.2 `pde_t* global_pgdir`

4.39.2.3 `proc_t* proc_table`

4.39.2.4 `struct Proc_Lifo running_procs`

## 4.40 kernel/rdisk.c File Reference

```
#include <rdisk.h>    #include <arch/x86/x86.h>    #include
<arch/x86/elf.h> Include dependency graph for rdisk.c:
```



## Functions

- `void readsect (void *, uint32_t)`
- `void readseg (uint32_t, uint32_t, uint32_t)`
- `void waitdisk (void)`

#### 4.40.1 Function Documentation

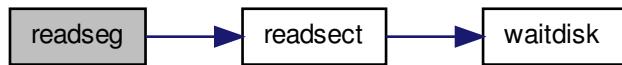
4.40.1.1 void readsect ( void \* *dst*, uint32\_t *offset* )

Here is the call graph for this function:



4.40.1.2 void readseg ( uint32\_t *va*, uint32\_t *count*, uint32\_t *offset* )

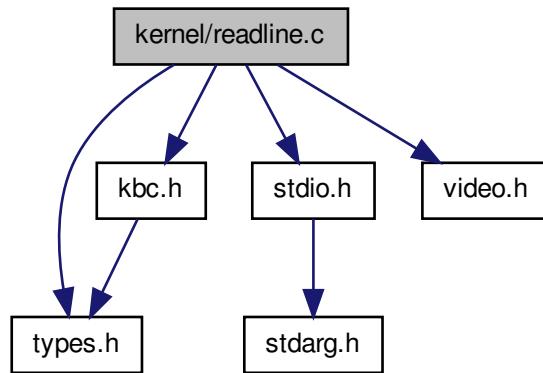
Here is the call graph for this function:



4.40.1.3 void waitdisk ( void )

## 4.41 kernel/readline.c File Reference

```
#include <types.h> #include <kbc.h> #include <stdio.h> x  
#include <video.h> Include dependency graph for readline.c:
```



### Defines

- `#define BUflen 1024`

### Functions

- `char * readline (const char *towrite)`

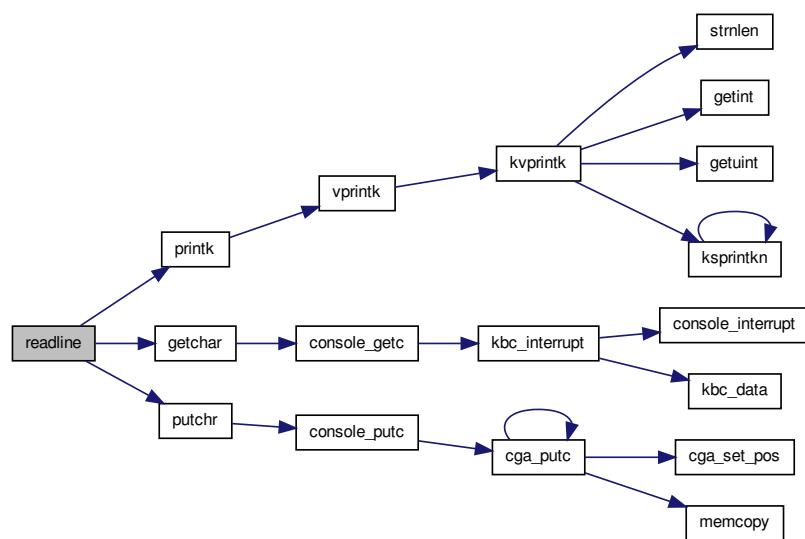
#### 4.41.1 Define Documentation

4.41.1.1 `#define BUflen 1024`

#### 4.41.2 Function Documentation

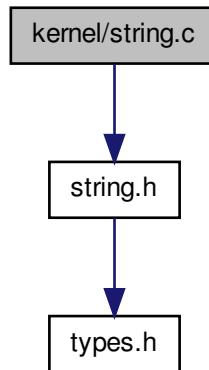
4.41.2.1 `char* readline( const char * touse )`

Here is the call graph for this function:



## 4.42 kernel/string.c File Reference

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



### Functions

- `uint32_t strnlen (const char *str, uint32_t count)`
- `void * memcpy (void *dst, const void *src, uint32_t count)`
- `void * memset (void *ptr, int c, uint32_t count)`

### 4.42.1 Function Documentation

4.42.1.1 `void* memcpy ( void * dst, const void * src, uint32_t count )`

4.42.1.2 `void* memset ( void * ptr, int c, uint32_t count )`

4.42.1.3 `uint32_t strnlen ( const char * str, uint32_t count )`

## 4.43 kernel/tmp/init\_elf.c File Reference

### Functions

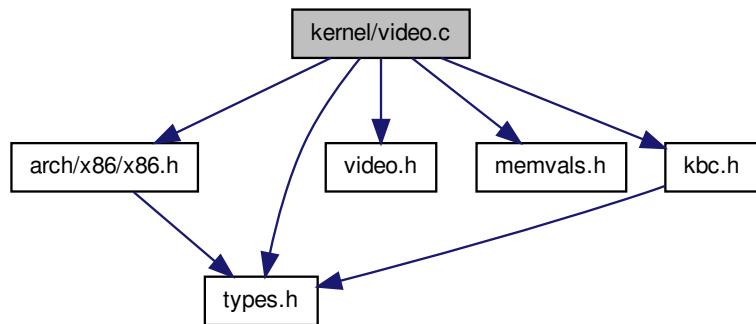
- `int main (void)`

#### 4.43.1 Function Documentation

4.43.1.1 `int main ( void )`

### 4.44 kernel/video.c File Reference

```
#include <arch/x86/x86.h>  #include <types.h>  #include
<video.h> #include <memvals.h> #include <kbc.h> Include de-
pendency graph for video.c:
```



### Functions

- `uint16_t cga_get_pos (void)`
- `void cga_set_attr (uint16_t c)`
- `void cga_clear (void)`
- `void cga_set_pos (uint16_t pos)`
- `void cga_init (void)`
- `void cga_putc (int c)`
- `cga_putstr (char *c)`

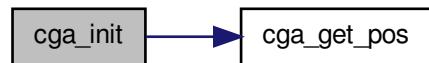
#### 4.44.1 Function Documentation

**4.44.1.1 void cga\_clear( void )**

Here is the call graph for this function:

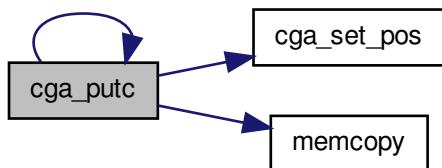
**4.44.1.2 uint16\_t cga\_get\_pos( void )****4.44.1.3 void cga\_init( void )**

Here is the call graph for this function:



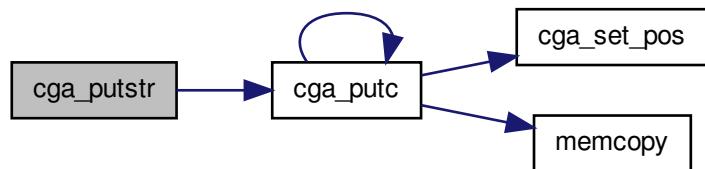
## 4.44.1.4 void cga\_putc( int c )

Here is the call graph for this function:



## 4.44.1.5 cga\_putstr( char \* c )

Here is the call graph for this function:



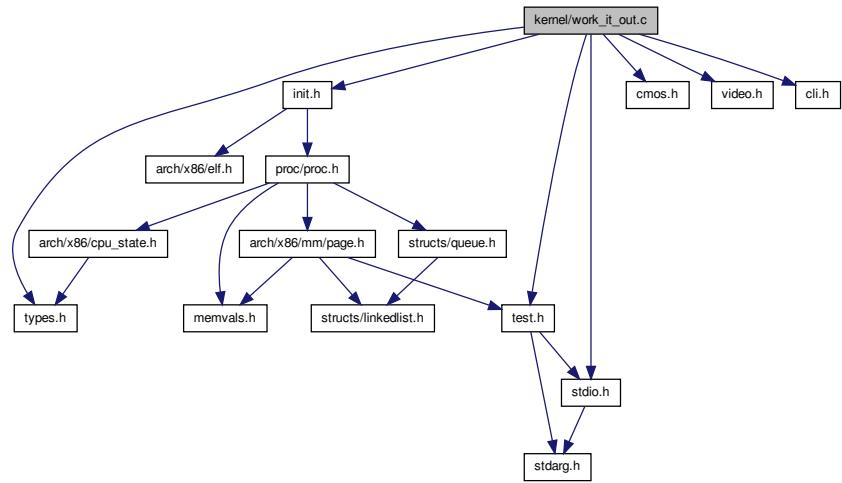
## 4.44.1.6 void cga\_set\_attr( uint16\_t c )

## 4.44.1.7 void cga\_set\_pos( uint16\_t pos )

## 4.45 kernel/work\_it\_out.c File Reference

```
#include <types.h> #include <cmos.h> #include <video.h>
#include <cli.h> #include <stdio.h> #include <test.h> x
```

```
#include <init.h> Include dependency graph for work_it_out.c:
```



## Functions

- void [work\\_it\\_out](#) (void)

### 4.45.1 Function Documentation

#### **4.45.1.1 void work\_it\_out( void )**

Here is the call graph for this function:

