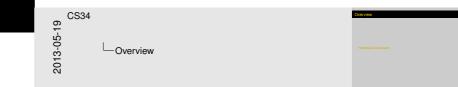


Overview



Homework Discussion

We should have talked about this last Thursday:

- 1. What does getpid do?
- 2. What does stime do?
- 3. What is interesting about readdir?

- 4. How many system calls does Linux have?
- 5. What did you run strace on, and what did you learn?



We should have talked about this last Thursday:

1. What does getpid do?

Returns the current process ID

- 2. What does stime do?
- 3. What is interesting about readdir?

- 4. How many system calls does Linux have?
- 5. What did you run strace on, and what did you learn?



We should have talked about this last Thursday:

1. What does getpid do?

Returns the current process ID

2. What does stime do?

Sets the time of day. What might that involve?

3. What is interesting about readdir?

- 4. How many system calls does Linux have?
- 5. What did you run strace on, and what did you learn?



#### Questions About Un

We should have talked about this last Thursday: 1. What does get pild do? Returns the current process ID 2. What does ntime do? Sets the time of day. What might that involve? 3. What is interesting about readitir?

How many system calls does Linux have?

What did you run strace on, and what did you learn

We should have talked about this last Thursday:

1. What does getpid do?

Returns the current process ID

2. What does stime do?

Sets the time of day. What might that involve?

3. What is interesting about readdir?

"This is not the function you are interested in."

- 4. How many system calls does Linux have?
- 5. What did you run strace on, and what did you learn?



#### Questions About Ur

We should have talked about this last Thursday: 1. What does extract do? Returns the current process ID 2. What does a time do? Seat the time of day. What might that involve? 3. What is interesting about readsts? "This is not the function you are interested in."

How many system calls does Linux have?

What did you run strace on, and what did you learn

We should have talked about this last Thursday:

1. What does getpid do?

Returns the current process ID

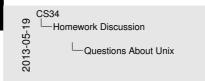
2. What does stime do?

Sets the time of day. What might that involve?

3. What is interesting about readdir?

"This is not the function you are interested in." ... nor is getdents.

- 4. How many system calls does Linux have?
- 5. What did you run strace on, and what did you learn?



#### Questions About Un

We should have talked about this last Thursday: 1. What does over 100 Hatura the current process ID 2. What does artime do? Seas the time of day. What might that involve? 3. What is interesting should rewards? "This is not the function you are interested in." ... And is get denta.

What did you run strace on, and what did you learn

We should have talked about this last Thursday:

1. What does getpid do?

Returns the current process ID

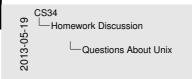
2. What does stime do?

Sets the time of day. What might that involve?

3. What is interesting about readdir?

"This is not the function you are interested in." ... nor is getdents.

- 4. How many system calls does Linux have? About 440.
- 5. What did you run strace on, and what did you learn?



#### uestions About Un

We should have talked about this last Thursday: 1. What does excite 407 Returns The curved process ID 2. What does a time 407 2. What is interesting about recent? This is not be include your amount in the excite This is not be include your amount in the excite ... Add sprif dents. ... Add sprif dents.

5. What did you run strace on, and what did you learn



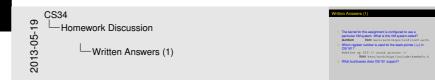
- 1. The kernel for this assignment is configured to use a particular VM system. What is this VM system called?
- 2. Which register number is used for the stack pointer (sp) in OS/161?

3. What bus/busses does OS/161 support?



- The kernel for this assignment is configured to use a particular VM system. What is this VM system called? dumbvm from kern/arch/mips/conf/conf.arch
- 2. Which register number is used for the stack pointer (sp) in OS/161?

3. What bus/busses does OS/161 support?



- The kernel for this assignment is configured to use a particular VM system. What is this VM system called? dumbvm from kern/arch/mips/conf/conf.arch
- 2. Which register number is used for the stack pointer (sp) in OS/161?

#define sp \$29 /\* stack pointer \*/
 from kern/arch/mips/include/asmdefs.h

3. What bus/busses does OS/161 support?



- The kernel for this assignment is configured to use a particular VM system. What is this VM system called? dumbvm from kern/arch/mips/conf/conf.arch
- 2. Which register number is used for the stack pointer (sp) in OS/161?

#define sp \$29 /\* stack pointer \*/
 from kern/arch/mips/include/asmdefs.h

3. What bus/busses does OS/161 support?

LAMEbus from kern/arch/mips/include/bus.h

Homework Discussion

## Written Answers (2)

4. What is the difference between splhigh and spl0?



5. Why do we use typedefs like u\_int32\_t instead of simply saying int?

6. What must be the first thing in the process control block?

4. What is the difference between splhigh and spl0? splhigh() sets spl to the highest value, disabling all interrupts

 ${\tt spl0}$  () sets spl to 0, enabling all interrupts

from kern/arch/mips/include/spl.h

5. Why do we use typedefs like u\_int32\_t instead of simply saying int?

6. What must be the first thing in the process control block?



4. What is the difference between splhigh and spl0? splhigh() sets spl to the highest value, disabling all interrupts

spl0() sets spl to 0, enabling all interrupts

from kern/arch/mips/include/spl.h

5. Why do we use typedefs like u\_int32\_t instead of simply saying int?

To make sure that we really get a 32-bit unsigned integer (unsigned int depends on the platform) *related to* kern/arch/mips/include/types.h

6. What must be the first thing in the process control block?



4. What is the difference between splhigh and spl0? splhigh() sets spl to the highest value, disabling all interrupts

spl0() sets spl to 0, enabling all interrupts

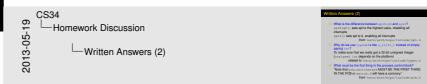
from kern/arch/mips/include/spl.h

5. Why do we use typedefs like u\_int32\_t instead of simply saying int?

To make sure that we really get a 32-bit unsigned integer (unsigned int depends on the platform) *related to* kern/arch/mips/include/types.h

6. What must be the first thing in the process control block? "Note that pcb\_switchstack MUST BE THE FIRST THING IN THE PCB or switch.S will have a coronary."

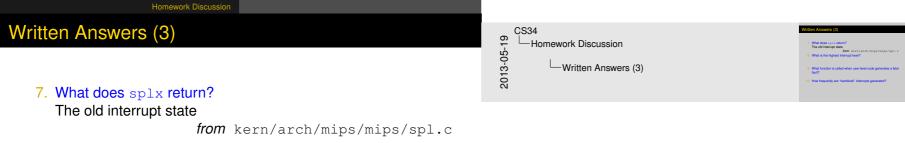
from kern/arch/mips/include/pcb.h





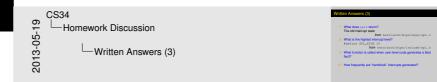
8. What is the highest interrupt level?

- 9. What function is called when user-level code generates a fatal fault?
- 10. How frequently are "hardclock" interrupts generated?



8. What is the highest interrupt level?

- 9. What function is called when user-level code generates a fatal fault?
- 10. How frequently are "hardclock" interrupts generated?



#### 7. What does splx return? The old interrupt state

from kern/arch/mips/mips/spl.c

8. What is the highest interrupt level?

#define SPL\_HIGH 15

from kern/arch/mips/include/spl.h

- 9. What function is called when user-level code generates a fatal fault?
- 10. How frequently are "hardclock" interrupts generated?



for kern/arch/mips/mips/spl.c

from kern/arch/mips/include/spl.h

7. What does splx return? The old interrupt state

from kern/arch/mips/mips/spl.c

8. What is the highest interrupt level?

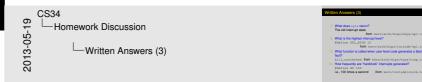
#define SPL\_HIGH 15

from kern/arch/mips/include/spl.h

9. What function is called when user-level code generates a fatal fault?

kill\_curthread *from* kern/arch/mips/mips/trap.c

10. How frequently are "hardclock" interrupts generated?



# 7. What does splx return? The old interrupt state

from kern/arch/mips/mips/spl.c

8. What is the highest interrupt level?

#define SPL\_HIGH 15

from kern/arch/mips/include/spl.h

9. What function is called when user-level code generates a fatal fault?

kill\_curthread *from* kern/arch/mips/mips/trap.c

10. How frequently are "hardclock" interrupts generated? #define HZ 100 i.o. 100 times a second from kern (include (cleak)

i.e., 100 times a second from kern/include/clock.h

11. What functions comprise the standard interface to a VFS device?

12. How many characters are allowed in an SFS volume name?

13. What is the standard interface to a file system (i.e., what functions must you implement to implement a new file system)?



11. What functions comprise the standard interface to a VFS device?

d\_open, d\_close, d\_io, d\_ioctl

from kern/include/dev.h

12. How many characters are allowed in an SFS volume name?

13. What is the standard interface to a file system (i.e., what functions must you implement to implement a new file system)?



11. What functions comprise the standard interface to a VFS device?

d\_open, d\_close, d\_io, d\_ioctl

from kern/include/dev.h

- 12. How many characters are allowed in an SFS volume name?
   #define SFS\_VOLNAME\_SIZE 32 /\* max length of
   volume name \*/ from kern/include/kern/sfs.h
- 13. What is the standard interface to a file system (i.e., what functions must you implement to implement a new file system)?

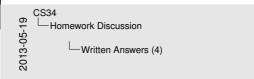


11. What functions comprise the standard interface to a VFS device?

d\_open, d\_close, d\_io, d\_ioctl

from kern/include/dev.h

- 12. How many characters are allowed in an SFS volume name?
   #define SFS\_VOLNAME\_SIZE 32 /\* max length of
   volume name \*/ from kern/include/kern/sfs.h
- 13. What is the standard interface to a file system (i.e., what functions must you implement to implement a new file system)?



Written Answers (4)		
11. What functions comprise the standard interface to a VFS device?		
d_open, d_close, d_io, d_ioctl #dm kern/include/dev.h		
<ol> <li>How many characters are allowed in an SFS volume name? #define SFS_VOLNAME_SITE 32 /* max length of volume name */ Nom kern/include/kern/sfs.h</li> </ol>		
13. What is the standard interface to a file system (i.e., what functions must you implement to implement a new file system)?		



CS34 Homework Discussion With Arasee (5) Written Answers (5) Written Asswers (5)

14. What function puts a thread to sleep?

15. How large are OS/161 pids?

16. What operations can you perform on a vnode?



#### 14. What function puts a thread to sleep?

15. How large are OS/161 pids?

16. What operations can you perform on a vnode?



#### 

typedef int32\_t pid\_t; /\* Process ID \*/ 32 bits/4 bytes form kern/include/kern/types.t 6. What operations can you perform on a vnode?

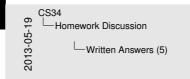
#### 14. What function puts a thread to sleep?

#### 15. How large are OS/161 pids?

typedef int32\_t pid\_t; /\* Process ID \*/

32 bits / 4 bytes from kern/include/kern/types.h

16. What operations can you perform on a vnode?



#### itten Answers (5)

What function puts a thread to skep?
 void thread\_sleep(const void \*addr);
 form from kern/include/thread.h
 How large are OS/161 pids?
 transfer (srilt = nid tr (A Process ID \*/

32 bits / 4 bytes from kern/include/kern/types.

16. What operations can you perform on a woode?

rine, ioci, ital, ata, gettype, tryseek, faync, mmap, runcate, namefile, creat, symlink, mkdir, link, emove, rmdir, renae, lookup, lookparent form kern/include/vnode.h

#### 14. What function puts a thread to sleep?

#### 15. How large are OS/161 pids?

typedef int32\_t pid\_t; /\* Process ID \*/

32 bits / 4 bytes from kern/include/kern/types.h

#### 16. What operations can you perform on a vnode?

open, close, reclaim, read, readlink, getdirentry, write, ioctl, stat, gettype, tryseek, fsync, mmap, truncate, namefile, creat, symlink, mkdir, link, remove, rmdir, rename, lookup, lookparent

from kern/include/vnode.h



17. What is the maximum path length in OS/161?

18. What is the system call number for a reboot?

19. Where is STDIN\_FILENO defined?



#### 17. What is the maximum path length in OS/161?

/\* Longest full path name \*/
#define PATH\_MAX 1024
 from kern/include/kern/limits.h

#### 18. What is the system call number for a reboot?

19. Where is STDIN\_FILENO defined?



#### 17. What is the maximum path length in OS/161?

#### 18. What is the system call number for a reboot?

**19. Where is STDIN\_FILENO defined?** 



#### 17. What is the maximum path length in OS/161?

/\* Longest full path name \*/ #define PATH\_MAX 1024

from kern/include/kern/limits.h

#### 18. What is the system call number for a reboot?

#define SYS\_reboot 8 /\* Reboot system \*/
 from kern/include/kern/callno.h
(C.f., RB\_REBOOT in kern/include/kern/unistd.h)

**19. Where is STDIN\_FILENO defined?** 



#### 21. Is it okay to initialize the thread system before the scheduler? Why or why not?

22. What is a zombie?

#### 20. What does kmain() do?

Kernel main. (Boot up, then fork the menu thread, wait for a reboot request, and then shut down.)

*from* kern/main/main.c

21. Is it okay to initialize the thread system before the scheduler? Why or why not?

22. What is a zombie?



#### 20. What does kmain() do?

Kernel main. (Boot up, then fork the menu thread, wait for a reboot request, and then shut down.)

*from* kern/main/main.c

21. Is it okay to initialize the thread system before the scheduler? Why or why not?

Yes. The scheduler bootstrap just creates the run queue, and the thread bootstrap just initializes the first thread.

22. What is a zombie?



#### 20. What does kmain() do?

Kernel main. (Boot up, then fork the menu thread, wait for a reboot request, and then shut down.)

*from* kern/main/main.c

21. Is it okay to initialize the thread system before the scheduler? Why or why not?

Yes. The scheduler bootstrap just creates the run queue, and the thread bootstrap just initializes the first thread.

#### 22. What is a zombie?

"Zombies are threads/processes that have exited but not been fully deleted yet." *from* kern/thread.c



Written Answers (7)				
20.	What does kmain () do?			
	Kernel main. (Boot up, then fork the menu thread, wait for a reboot request, and then shut down.) from kern/main/main.			
	<ol> <li>Is it okay to initialize the thread system before the schedule Why or why not? Yes. The scheduler bootstrap just creates the run queue, at the thread bootstrap just initializes the first thread.</li> </ol>			
	What is a zombie? "Zombies are threads/processes that have exited but not been fully deleted yet." from kern/thread.thread.			
	How large is the initial run queue?			

#### 20. What does kmain() do?

Kernel main. (Boot up, then fork the menu thread, wait for a reboot request, and then shut down.)

*from* kern/main/main.c

21. Is it okay to initialize the thread system before the scheduler? Why or why not?

Yes. The scheduler bootstrap just creates the run queue, and the thread bootstrap just initializes the first thread.

#### 22. What is a zombie?

"Zombies are threads/processes that have exited but not been fully deleted yet." *from* kern/thread.c

#### 23. How large is the initial run queue?

```
runqueue = q_create(32);
```

from kern/thread/scheduler.c



Homework Discussion

### Written Answers (8)

24. Can an array represented by a struct array be resized?

25. What does a device name in OS/161 look like?

26. What does a raw device name in OS/161 look like?

27. What lock protects the vnode reference count?



25. What does a device name in OS/161 look like?

26. What does a raw device name in OS/161 look like?

27. What lock protects the vnode reference count?



### 25. What does a device name in OS/161 look like?

The name of a device is always *device*:, such as lhd0: *from* kern/fs/vfs/device.c

26. What does a raw device name in OS/161 look like?

27. What lock protects the vnode reference count?



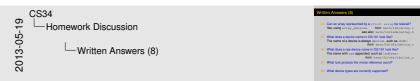
### 25. What does a device name in OS/161 look like?

The name of a device is always *device*:, such as lhd0: *from* kern/fs/vfs/device.c

## 26. What does a raw device name in OS/161 look like?

The name with raw appended, such as lhd0raw:
 from kern/fs/vfs/vfslist.c

- 27. What lock protects the vnode reference count?
- 28. What device types are currently supported?



### 25. What does a device name in OS/161 look like?

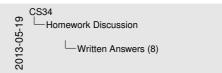
The name of a device is always *device*:, such as lhd0: *from* kern/fs/vfs/device.c

## 26. What does a raw device name in OS/161 look like?

The name with raw appended, such as lhd0raw:
 from kern/fs/vfs/vfslist.c

### 27. What lock protects the vnode reference count?

vn\_countlock from kern/fs/vfs/vnode.c



Written Answers (8)
<ol> <li>Can an array represented by a struct stray be resized? Yes, using array_stsize. Some kern/lib/array.c Sole abo kern/lib/array.1</li> <li>What does a device name in CSF10 look Nam?</li> </ol>
The name of a device is always devicer, such as 1hd0: from kern/fs/vfs/device.c
<ol> <li>What does a raw device name in OS/161 look like? The name with raw appended, such as 1hd0raw: from kern/fs/vfs/vfslist.c</li> </ol>
27. What lock protects the wnode reference count? vn_countlock from kern/fs/vfs/vnode.c

### 25. What does a device name in OS/161 look like?

The name of a device is always *device*:, such as lhd0: *from* kern/fs/vfs/device.c

### 26. What does a raw device name in OS/161 look like?

The name with raw appended, such as lhd0raw:
 from kern/fs/vfs/vfslist.c

### 27. What lock protects the vnode reference count?

vn\_countlock *from* kern/fs/vfs/vnode.c

28. What device types are currently supported?

Block & character devices. from kern/fs/vfs/device.c



Written Answers (8)		
		ted by a struct array be resized? taize. #000 kern/lib/array. see also kern/include/array.
		ame in OS/161 look like? is always devicet, such as 1hd0: form kern/fs/vfs/device.
		pended, such as lhd0raw: from kern/fs/vfs/vfslist.
	What lock protects the vn_count lock	whode reference count? Nom kern/fs/vfs/vhode.

Block & character devices. from kern/fs/vfs/device.