Operating Systems

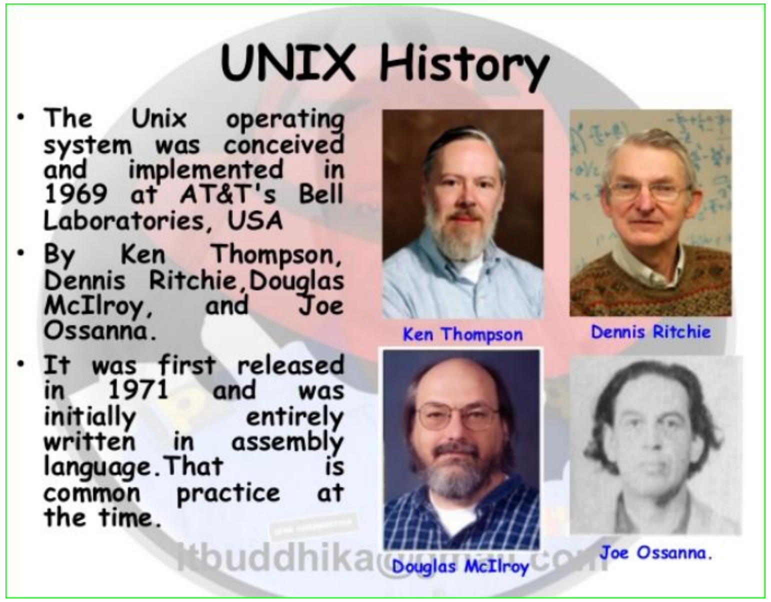
Mini Projects

Juhyung Park

arter97@dgist.ac.kr

Xv6

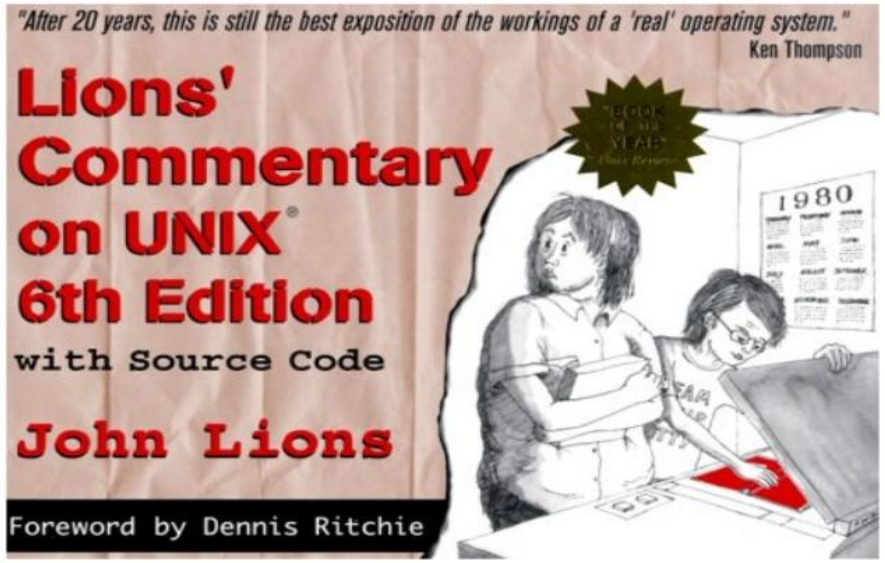
- Unix-like teaching operating system developed by MIT
- Reimplementation of v6 for a modern x86–based multiprocessor using ANSI C.
- Provide basic interface introduced by Ken Thompson and Dennis Ritchie's Unix operating system, as well as mimicking Unix's internal design



UNIX History

- The Unix operating system was conceived and implemented in 1969 at AT&T's Bell Laboratories, USA
- By Ken Thompson, Dennis Ritchie, Douglas McIlroy, and Joe Ossanna.
- It was first released in 1971 and was initially entirely written in assembly language. That is common practice at the time.

Ken Thompson
Dennis Ritchie
Douglas McIlroy
Joe Ossanna.



"After 20 years, this is still the best exposition of the workings of a 'real' operating system."
Ken Thompson

Lions' Commentary on UNIX® 6th Edition with Source Code

John Lions

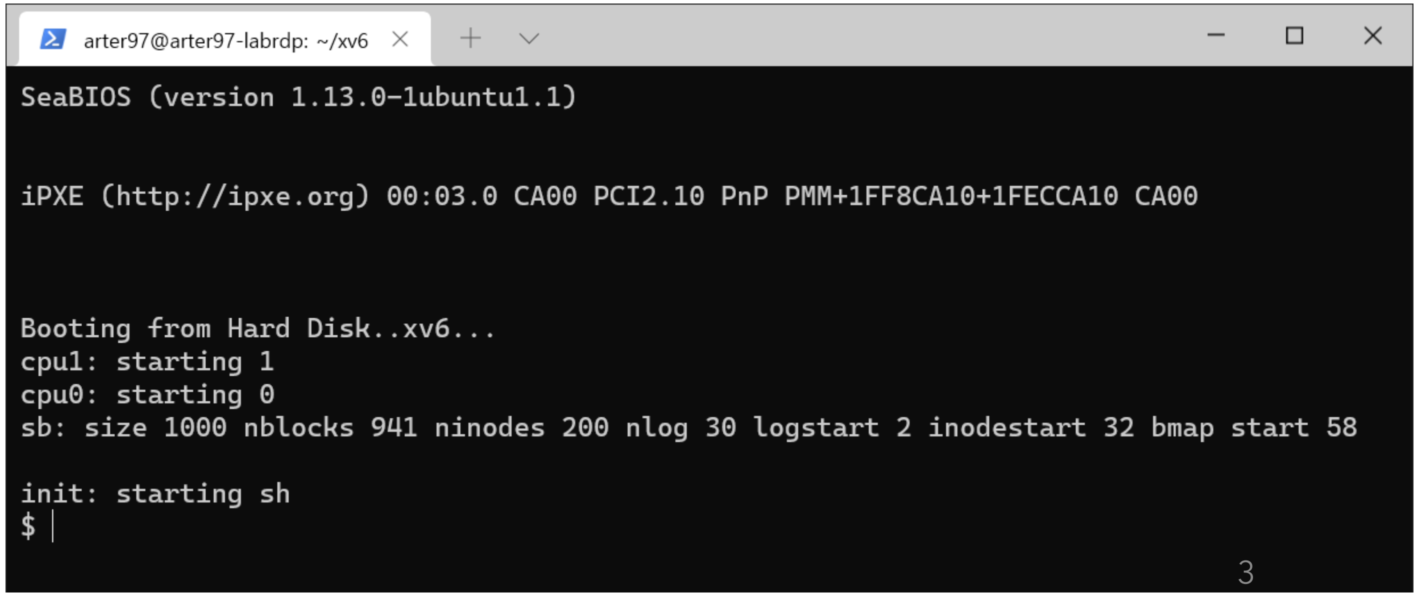
Foreword by Dennis Ritchie

Xv6 Installation

- Type:

```
# git clone https://github.com/dgist-datalab/xv6
cd xv6
git fetch
git checkout miniprj-2024
make qemu-nox -j
```

- Uses Git to download Xv6 source code
- Build & run with QEMU



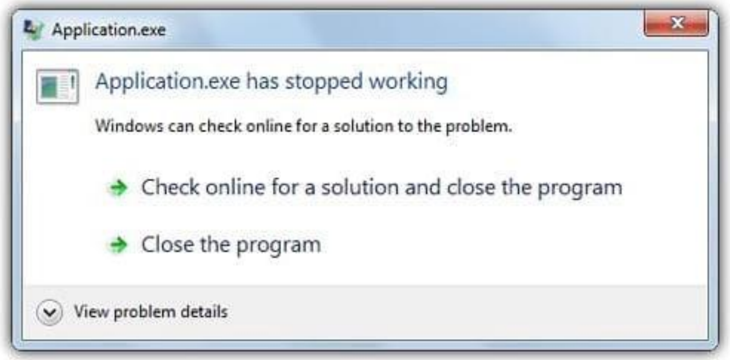
```
SeaBIOS (version 1.13.0-1ubuntu1.1)


iPXE (http://ipxe.org) 00:03.0 CA00 PCI2.10 PnP PMM+1FF8CA10+1FECCA10 CA00



Booting from Hard Disk..xv6...
cpu1: starting 1
cpu0: starting 0
sb: size 1000 nblocks 941 ninodes 200 nlog 30 logstart 2 inodestart 32 bmap start 58

init: starting sh
$
```

Xv6

- Caution
  - You're modifying a real operating system
  - Conventional assumptions made with userspace programming can't be applied
  - Standard C functions may not be available: stdio.h, malloc, printf, etc
    - Xv6 kernel: replacement functions may be available: kalloc, cprintf
  - Xv6 userspace is not POSIX-compliant
    - Some popular function's usage and behavior may differ: `printf(…) -> printf(1, …)`
  - In case of errors – instead of a segmentation fault, you'll encounter a total (virtual machine) system failure

Kernel is not kind

- Instead of a segmentation fault, you’ll encounter a total (virtual machine) system failure

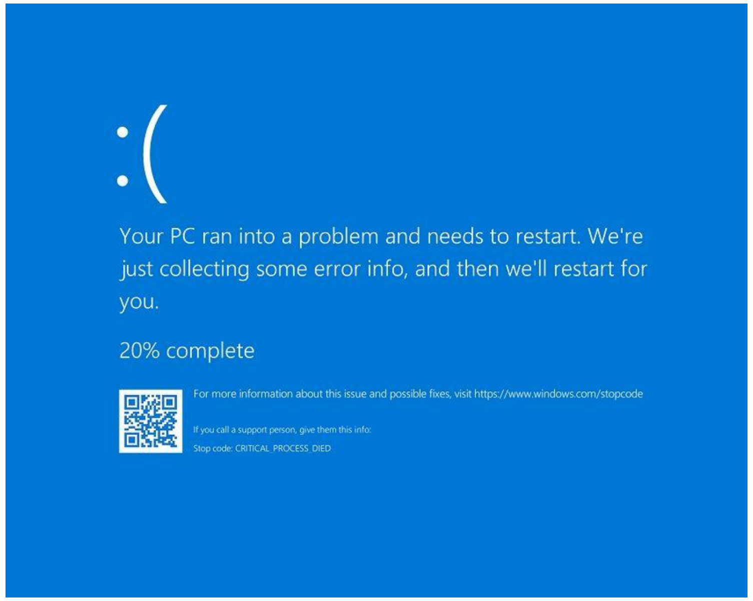


Application.exe

Application.exe has stopped working

Windows can check online for a solution to the problem.

▲ Coding mistakes in regular applications



:(

Your PC ran into a problem and needs to restart. We're just collecting some error info, and then we'll restart for you.

20% complete

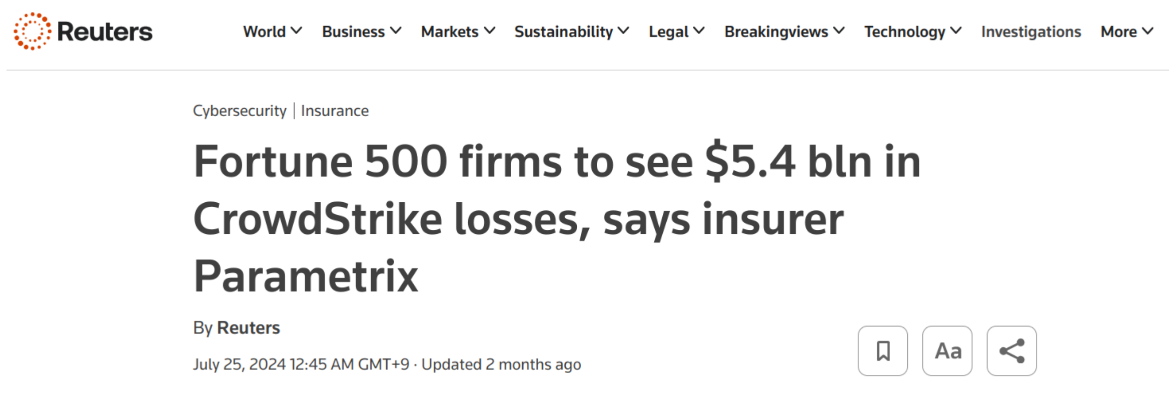
For more information about this issue and possible fixes, visit https://www.windows.com/stopcode

If you call a support person, give them this info:

Stop code: CRITICAL_PROCESS_DIED

▲ Coding mistakes in the kernel

Don’t be like CrowdStrike

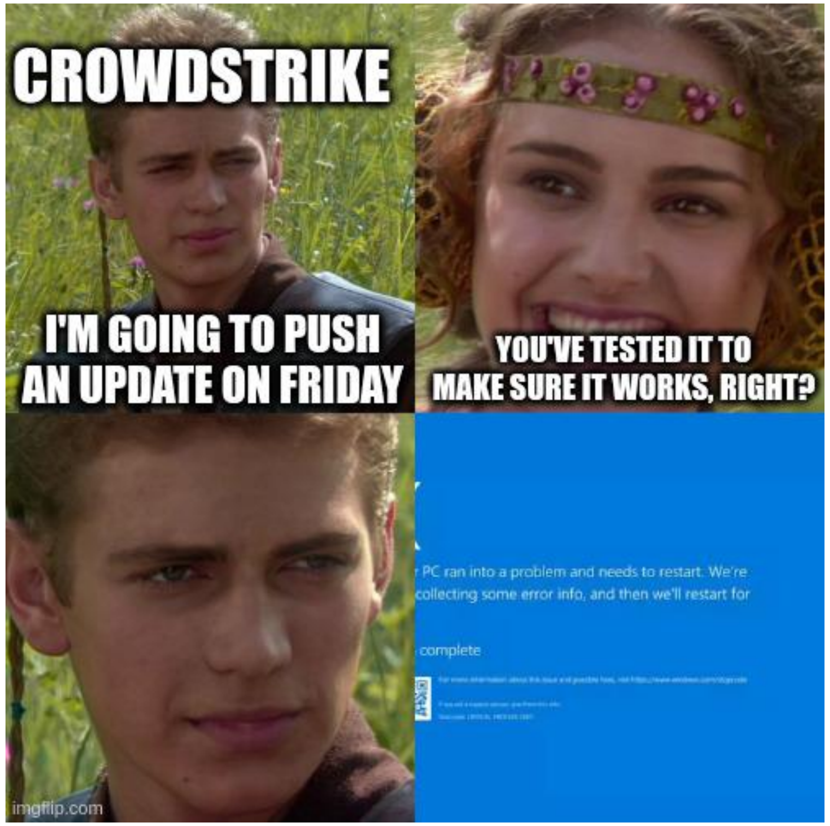


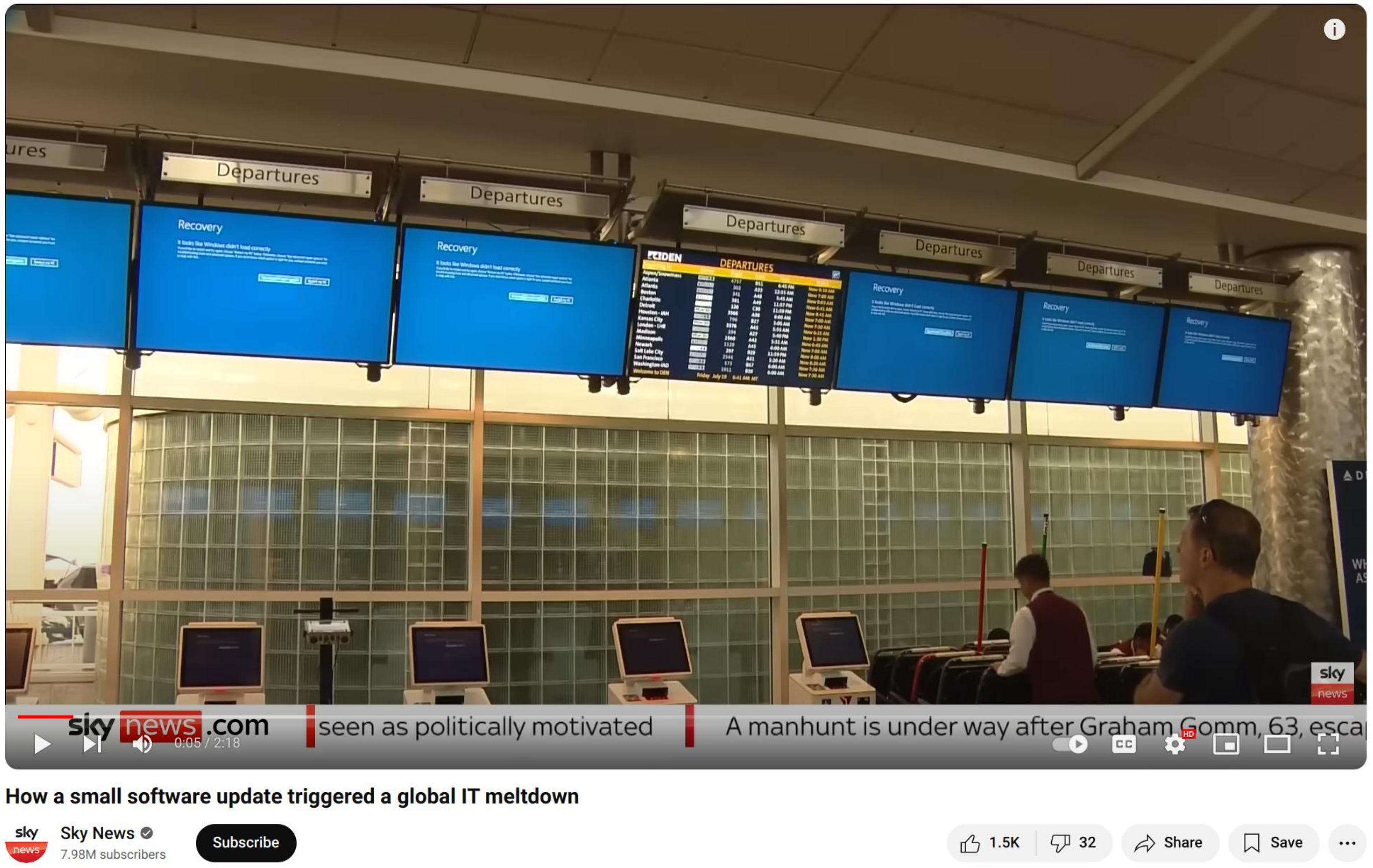
Cybersecurity | Insurance

Fortune 500 firms to see $5.4 bln in CrowdStrike losses, says insurer Parametrix

By Reuters

July 25, 2024 12:45 AM GMT+9 · Updated 2 months ago

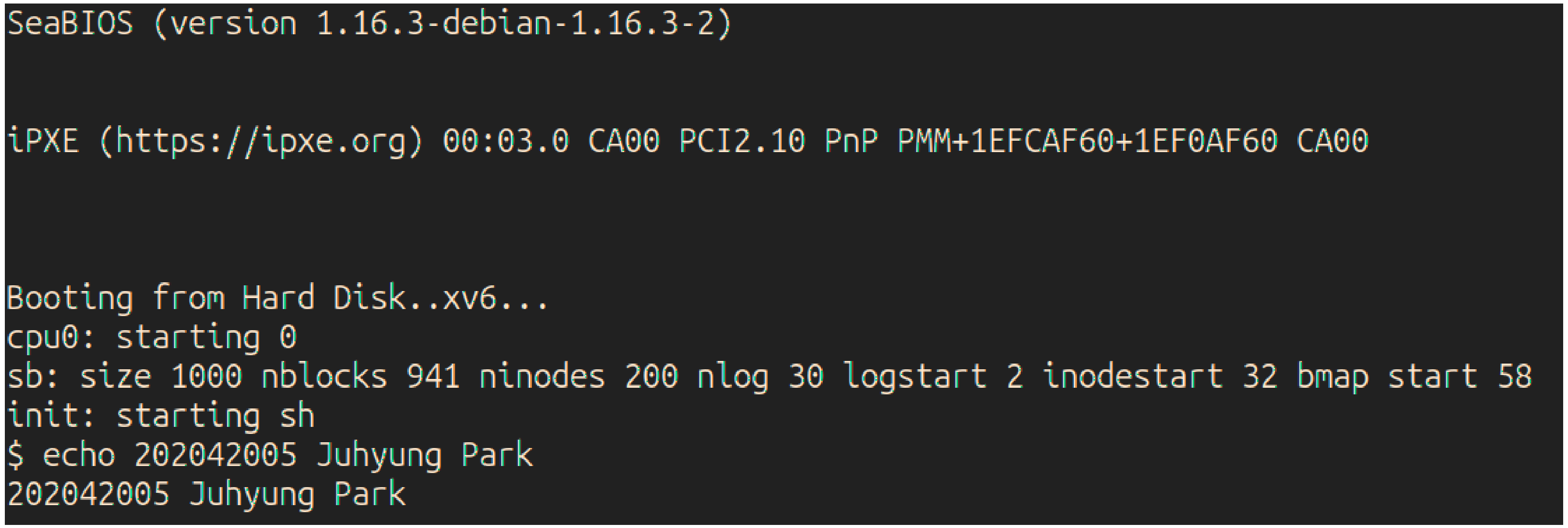

CROWDSTRIKE
I'M GOING TO PUSH AN UPDATE ON FRIDAY
YOU'VE TESTED IT TO MAKE SURE IT WORKS, RIGHT?
PC ran into a problem and needs to restart. We're collecting some error info, and then we'll restart for
complete
imgflip.com



How a small software update triggered a global IT meltdown

Mini Project #1 - Print process information upon termination

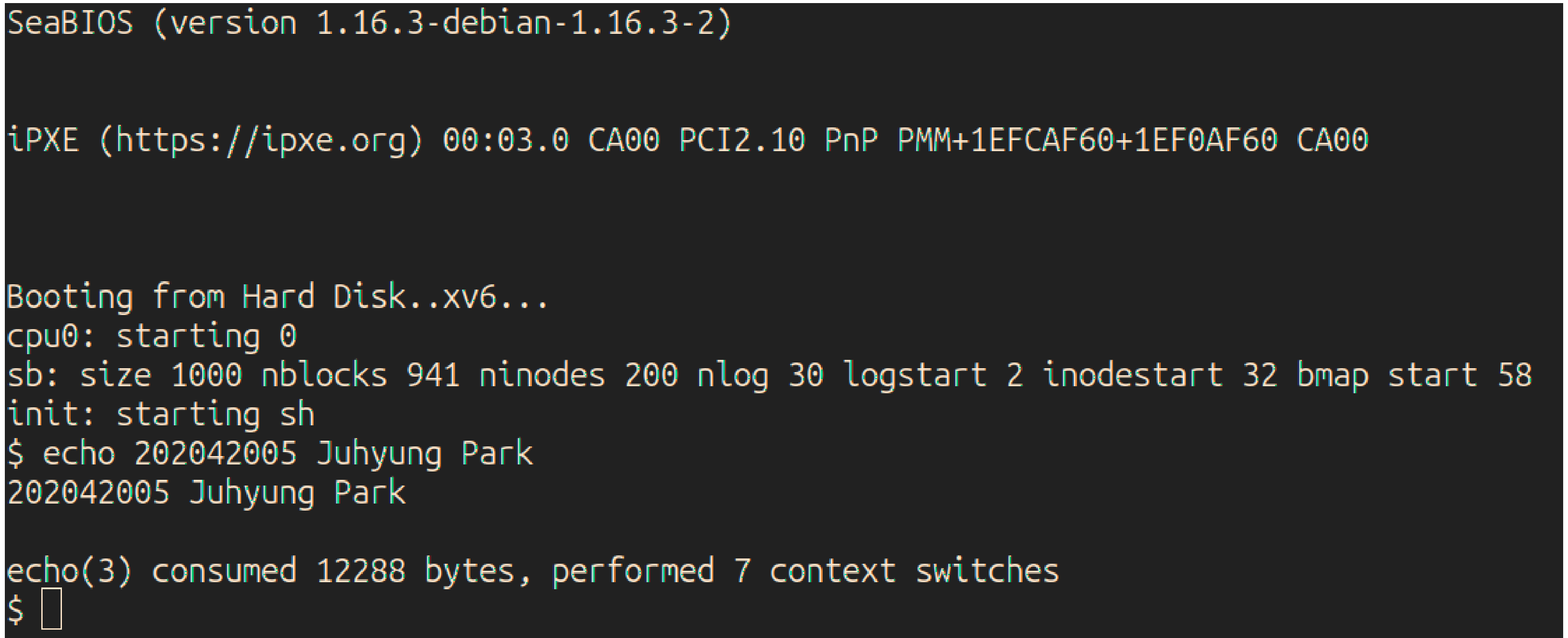
- Run some programs



```
SeaBIOS (version 1.16.3-debian-1.16.3-2)


iPXE (https://ipxe.org) 00:03.0 CA00 PCI2.10 PnP PMM+1EFCAF60+1EF0AF60 CA00



Booting from Hard Disk..xv6...
cpu0: starting 0
sb: size 1000 nblocks 941 ninodes 200 nlog 30 logstart 2 inodestart 32 bmap start 58
init: starting sh
$ echo 202042005 Juhyung Park
202042005 Juhyung Park
```

Mini Project #1 - Print process information upon termination

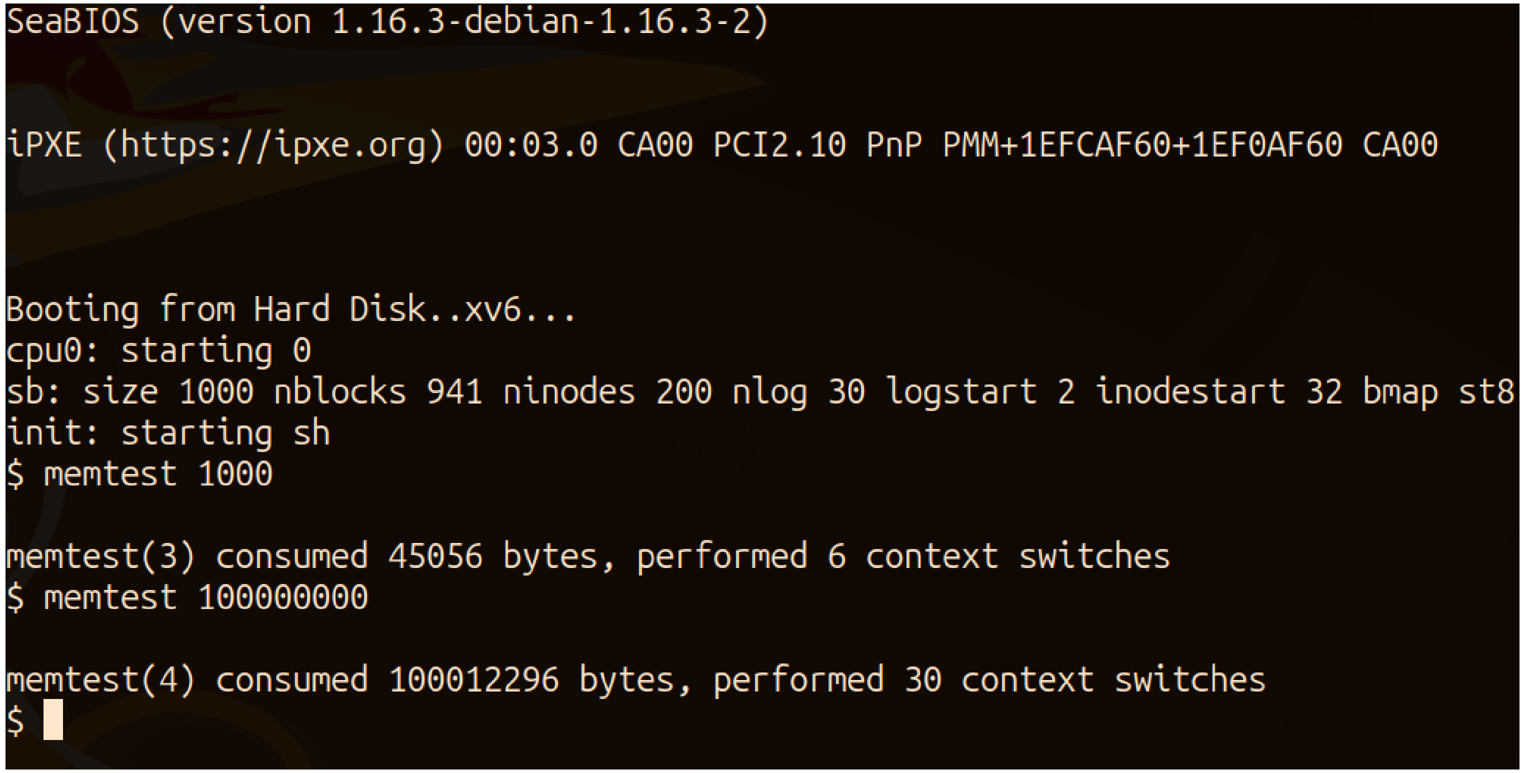
- Run some programs
- **… and also print process information**
    - Process name, process ID, process memory size, number of context switches
    - "`<Process name>(<PID>) consumed <process memory size> bytes, performed <N of context switches> context switches`"



```
SeaBIOS (version 1.16.3-debian-1.16.3-2)


iPXE (https://ipxe.org) 00:03.0 CA00 PCI2.10 PnP PMM+1EFCAF60+1EF0AF60 CA00



Booting from Hard Disk..xv6...
cpu0: starting 0
sb: size 1000 nblocks 941 ninodes 200 nlog 30 logstart 2 inodestart 32 bmap start 58
init: starting sh
$ echo 202042005 Juhyung Park
202042005 Juhyung Park

echo(3) consumed 12288 bytes, performed 7 context switches
$ 
```

Mini Project #1 - Print process information upon termination

- Confirm the reported process memory size changes accordingly with `memtest`
  - `memtest: malloc(atoi(argv[1]))`
- Bigger memory allocation generally takes longer



```
SeaBIOS (version 1.16.3-debian-1.16.3-2)


iPXE (https://ipxe.org) 00:03.0 CA00 PCI2.10 PnP PMM+1EFCAF60+1EF0AF60 CA00



Booting from Hard Disk..xv6...
cpu0: starting 0
sb: size 1000 nblocks 941 ninodes 200 nlog 30 logstart 2 inodestart 32 bmap st8
init: starting sh
$ memtest 1000

memtest(3) consumed 45056 bytes, performed 6 context switches
$ memtest 100000000

memtest(4) consumed 100012296 bytes, performed 30 context switches
$ 
```

Mini Project #1 - Print process information upon termination

- Objectives of this project
    - Find the termination/exit point of a process from the kernel code and modify it
    - Find the function responsible for context switches from the scheduler code and modify it
    - Find the PCB (Process Control Block) structure in Xv6 and understand it
    - Modify the PCB to keep track of the number of context switches
    - Print relevant information from PCB

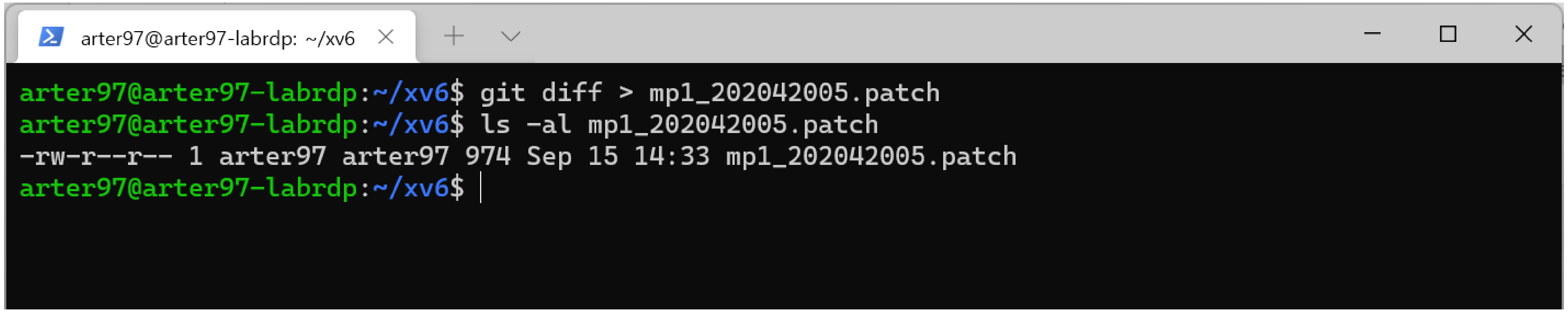
- Where to look and write code:
    - proc.c, proc.h

- How to print:

```
// Print the number of context switches (cswitch)
cprintf("\n%s(%d) consumed %d bytes, performed %d context switches\n", …);
```

Mini Project #1 - Print process information upon termination

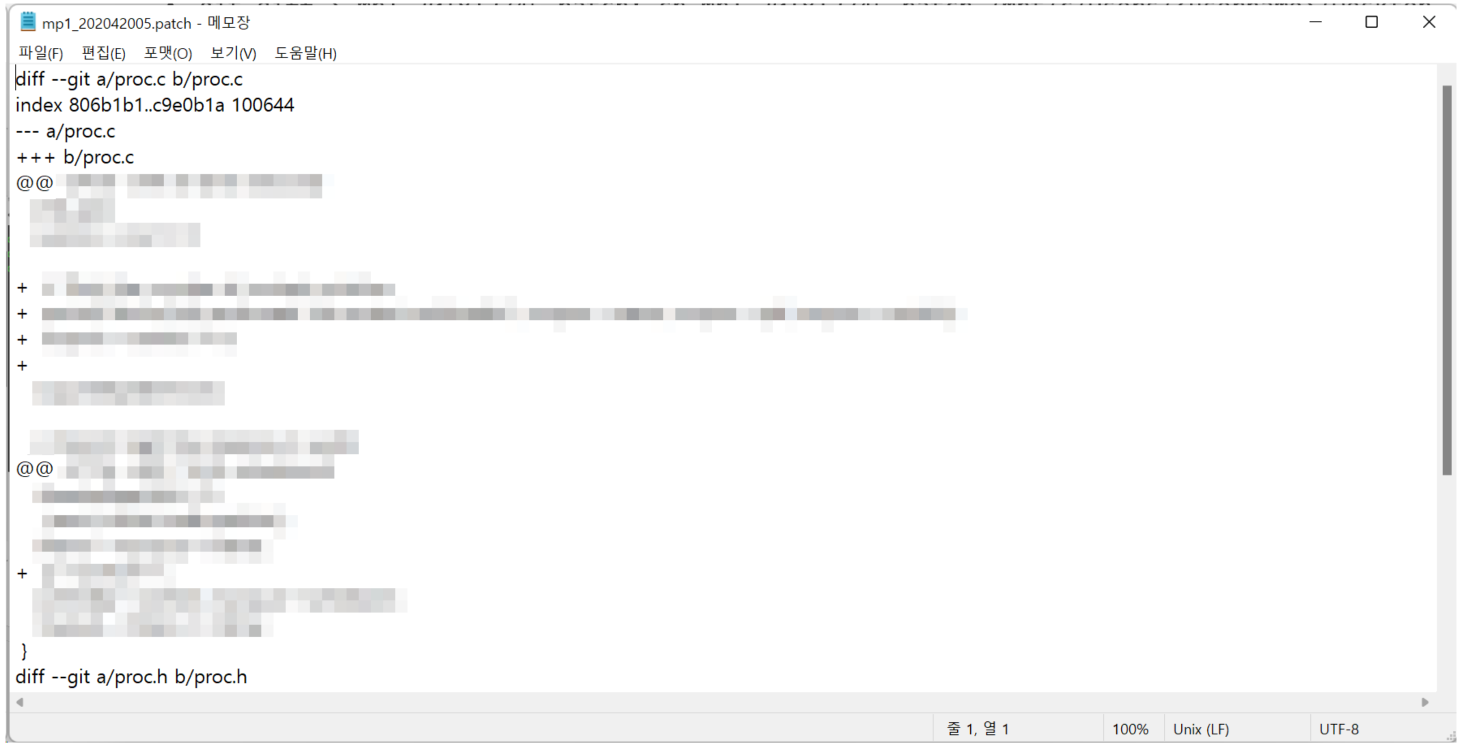
- Hand-in procedure
  - mp1_201812345.patch
    - Run the following command and upload mp1_201812345.patch
      - `git diff > mp1_201812345.patch`



```
arter97@arter97-labrdp:~/xv6$ git diff > mp1_202042005.patch
arter97@arter97-labrdp:~/xv6$ ls -al mp1_202042005.patch
-rw-r--r-- 1 arter97 arter97 974 Sep 15 14:33 mp1_202042005.patch
arter97@arter97-labrdp:~/xv6$
```

Mini Project #1 - Print process information upon termination

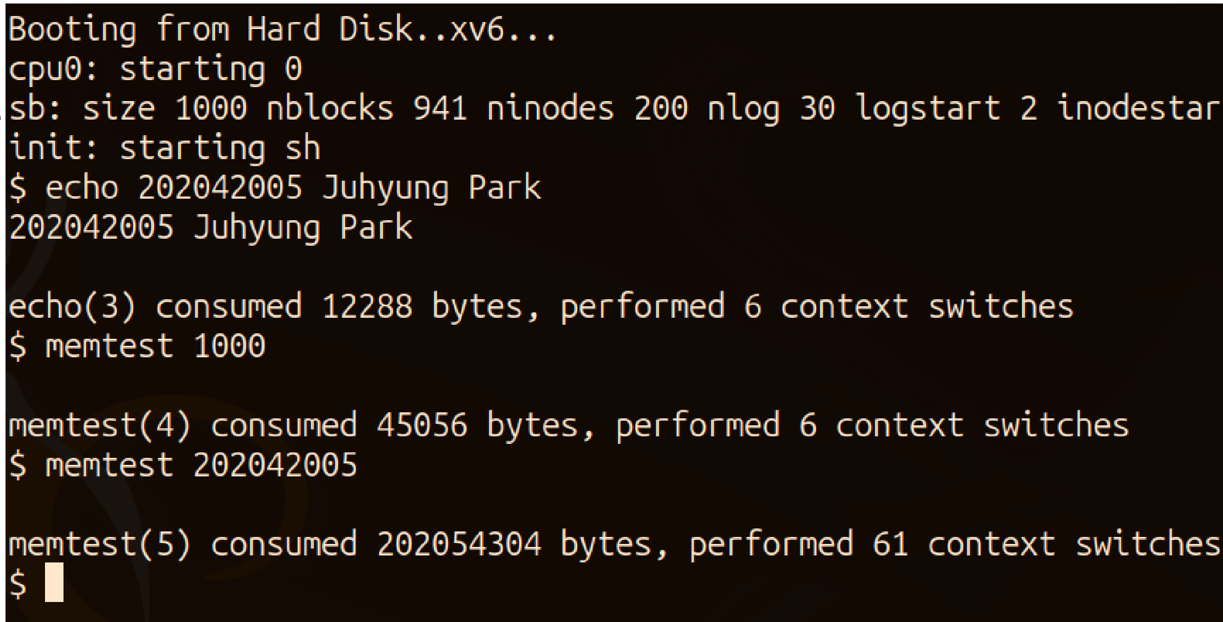
- Hand-in procedure
  - mp1_201812345.patch
    - Run the following command and upload mp1_201812345.patch
      - `git diff > mp1_201812345.patch`
    - Check the patch file with Notepad and confirm your modifications are in the patch file


mp1_202042005.patch - 메모장
파일(F) 편집(E) 포맷(O) 보기(V) 도움말(H)
줄 1, 열 1
100%
Unix (LF)
UTF-8

```
diff --git a/proc.c b/proc.c
index 806b1b1..c9e0b1a 100644
--- a/proc.c
+++ b/proc.c
@@ [redacted]
[redacted]

+ [redacted]
+ [redacted]
+ [redacted]
+
[redacted]

[redacted]
@@ [redacted]
[redacted]
+ [redacted]
[redacted]
 }
diff --git a/proc.h b/proc.h
```

Mini Project #1 - Print process information upon termination

- Hand-in procedure
  - mp1_201812345.patch
    - Run the following command and upload mp1_201812345.patch
      - `git diff > mp1_201812345.patch`
    - Check the patch file with Notepad and confirm your modifications are in the patch file
  - mp1_201812345.txt: 1-2 line short report
    - Explain why bigger memtest incurs more context switches
  - mp1_201812345.jpg: Screenshot
    - echo <Student ID> <Your name in English>
    - 2 memtest commands:
      - `memtest 1000`
      - `memtest <Student ID>`
  - Deadline: 2024.09.25 (Wed) 23:59
  - Do this before you move on to mp2!



```
Booting from Hard Disk..xv6...
cpu0: starting 0
sb: size 1000 nblocks 941 ninodes 200 nlog 30 logstart 2 inodestar
init: starting sh
$ echo 202042005 Juhyung Park
202042005 Juhyung Park

echo(3) consumed 12288 bytes, performed 6 context switches
$ memtest 1000

memtest(4) consumed 45056 bytes, performed 6 context switches
$ memtest 202042005

memtest(5) consumed 202054304 bytes, performed 61 context switches
$ 
```

Mini Project #2 - bash-like command prompt

- Xv6's command prompt only shows "$"

```
init: starting sh
$ cd test
$
```

- Bash's command prompt is more complex

```
root@arter97-x1:/#
```

  - Username
  - Hostname
  - Current working directory

Mini Project #2 - bash-like command prompt

- Xv6's command prompt only shows "$"

```
init: starting sh
$ cd test
$
```

- Bash's command prompt is more complex

```
root@arter97-x1:/#
```

  - Username - Xv6 only uses root, hardcode it!
  - Hostname - User-customizable
  - Current working directory - Skip

Mini Project #2 - bash-like command prompt

- Xv6's command prompt only shows "$"

```
init: starting sh
$ cd test
$
```

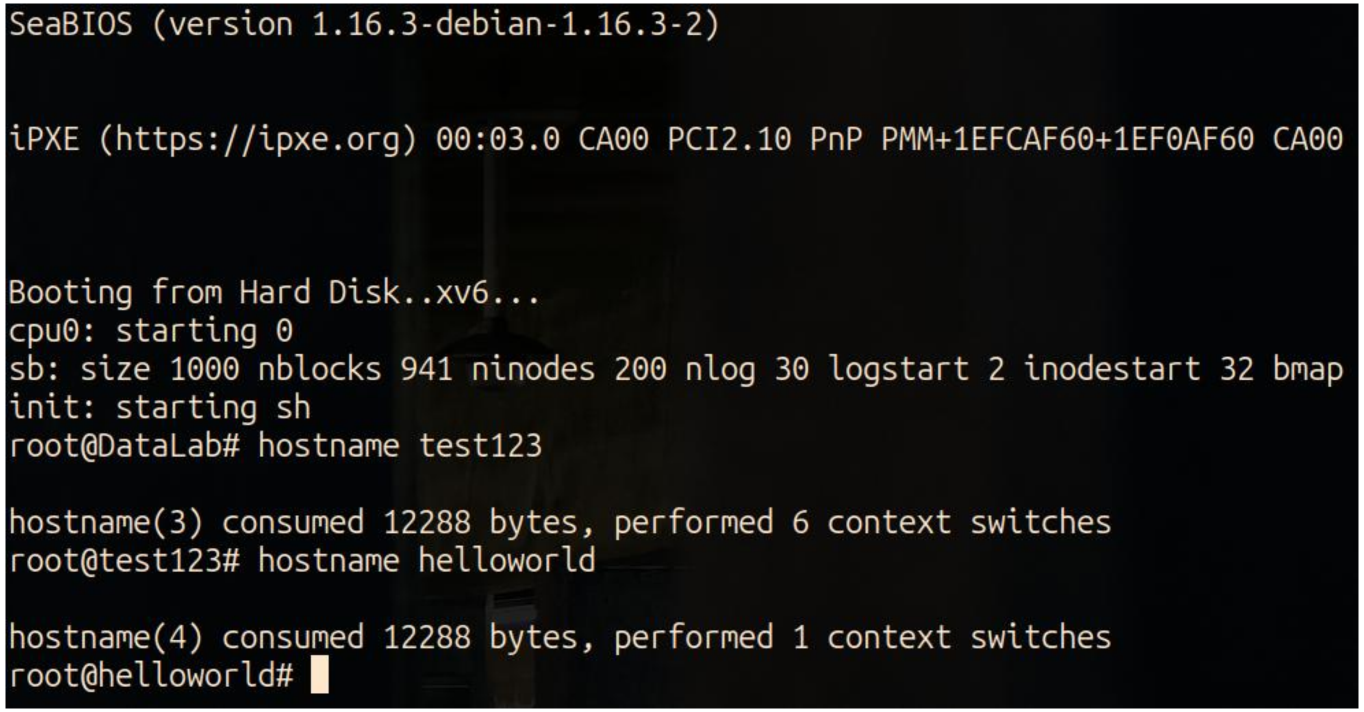
- Bash's command prompt is more complex

```
root@arter97-x1:/#
```

- Username - Xv6 only uses root, hardcode it!
    - Hardcode "root" - sh.c
- Hostname - User-customizable
    - **Implement getter and setter for hostname**
- Current working directory - Skip

Mini Project #2 - bash-like command prompt

- Objectives of this project
  - Find the system-call table and register your own system-call number
  - Implement your own system-call
  - Understand how the kernel-space and the user-space exchanges data
    - Kernel stores the current hostname: char hostname[64] = “DataLab”;



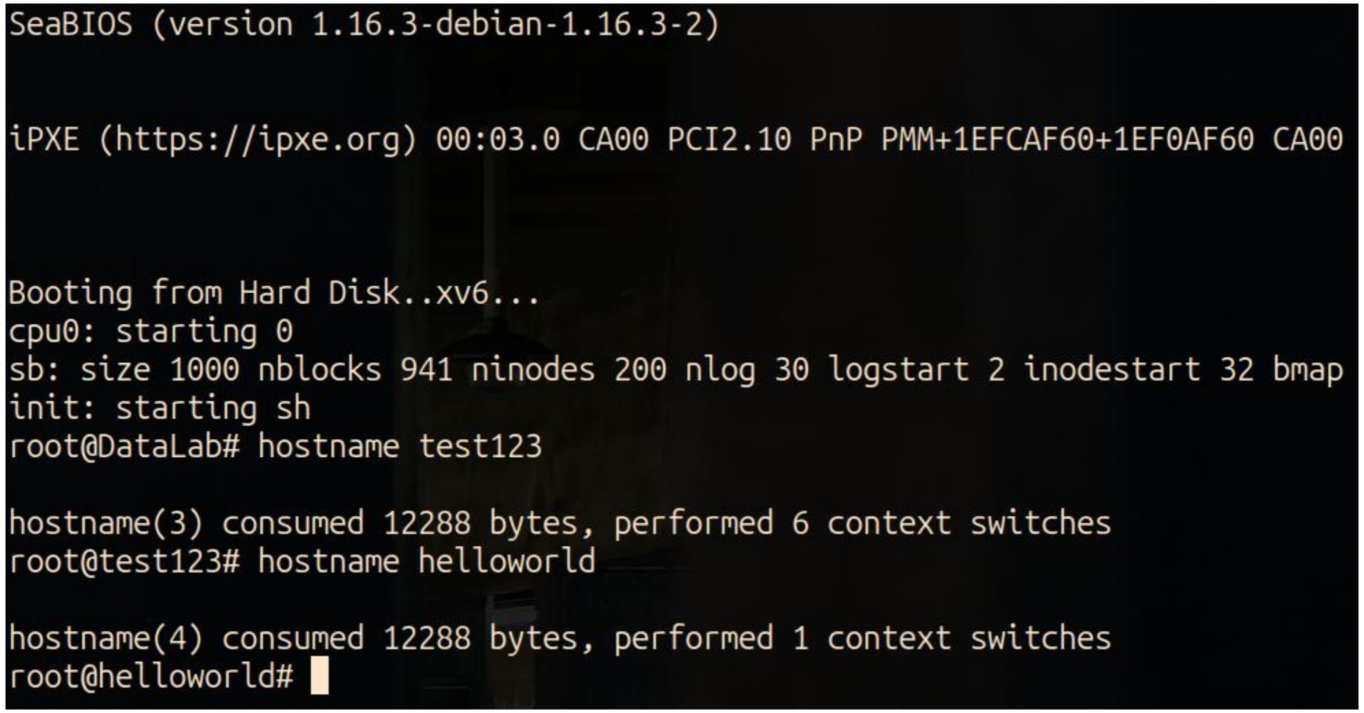
```
SeaBIOS (version 1.16.3-debian-1.16.3-2)


iPXE (https://ipxe.org) 00:03.0 CA00 PCI2.10 PnP PMM+1EFCAF60+1EF0AF60 CA00



Booting from Hard Disk..xv6...
cpu0: starting 0
sb: size 1000 nblocks 941 ninodes 200 nlog 30 logstart 2 inodestart 32 bmap
init: starting sh
root@DataLab# hostname test123

hostname(3) consumed 12288 bytes, performed 6 context switches
root@test123# hostname helloworld

hostname(4) consumed 12288 bytes, performed 1 context switches
root@helloworld# █
```

Mini Project #2 - bash-like command prompt

- Where to look and write code:
  - **syscall.c, syscall.h**: Function prototype declaration, syscall table insertion
    - `extern int sys_gethostname(void), extern int sys_sethostname(void)`
  - **sysproc.c/sysfile.c**: System-call implementation
    - `char hostname[64] = "DataLab";`
    - Hint: Reference other sys_*() to find out how to retrieve arguments (argint()/argptr()/argstr())
  - **user.h**: Function prototype declaration for user-space programs
    - `int gethostname(char *)`
      - Copies kernel's hostname to argument
    - `int sethostname(const char *)`
      - Copies user's hostname from argument to kernel's hostname
  - **usys.S**: Entry point of the system-call
  - **sh.c**: Shell prompt
    - getcmd(): Change the format to "`printf(2, "root@%s# ", hostname);`"

Mini Project #2 - bash-like command prompt

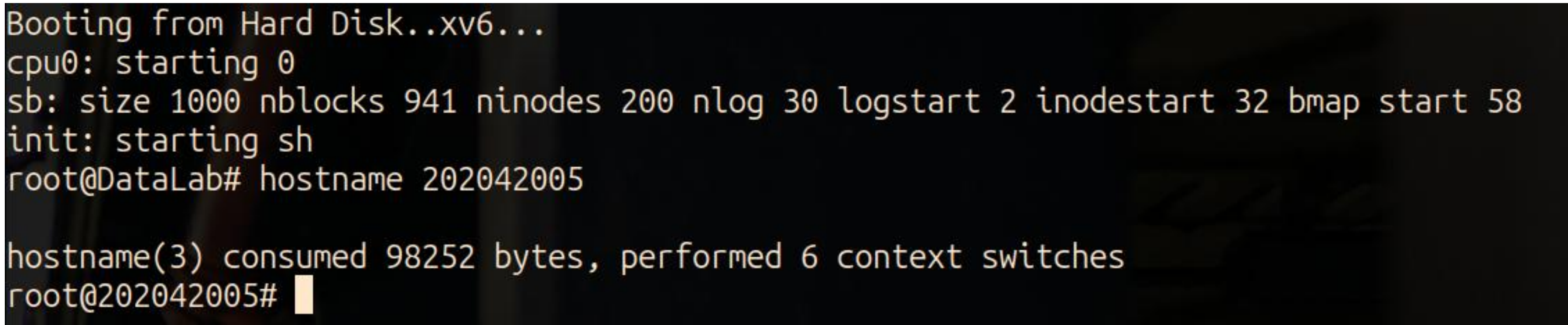
- **hostname.c**: Program that changes hostname
  - Usage: hostname [new hostname]
    - e.g., `hostname test123`
  - Needs hostname system-calls to be implemented
    - So, hostname compilation is disabled by default
    - Enable it by de-commenting _**hostname** from "Makefile" after you've finished implementing system-calls!



```
SeaBIOS (version 1.16.3-debian-1.16.3-2)


iPXE (https://ipxe.org) 00:03.0 CA00 PCI2.10 PnP PMM+1EFCAF60+1EF0AF60 CA00



Booting from Hard Disk..xv6...
cpu0: starting 0
sb: size 1000 nblocks 941 ninodes 200 nlog 30 logstart 2 inodestart 32 bmap
init: starting sh
root@DataLab# hostname test123

hostname(3) consumed 12288 bytes, performed 6 context switches
root@test123# hostname helloworld

hostname(4) consumed 12288 bytes, performed 1 context switches
root@helloworld# █
```

Mini Project #2 - bash-like command prompt

- Hand-in procedure
  - mp2_201812345.patch
    - Run the following command and upload mp2_201812345.patch
      - `git diff > mp2_201812345.patch`
    - Check the patch file with Notepad and confirm your modifications are in the patch file
    - mp1 changes can be included in the patch
    - **Warning: TAs will check against C mistakes/errors**
  - mp2_201812345.jpg
    - `hostname <Student ID>`
  - Deadline: 2024.09.25 (Wed) 23:59



```
Booting from Hard Disk..xv6...
cpu0: starting 0
sb: size 1000 nblocks 941 ninodes 200 nlog 30 logstart 2 inodestart 32 bmap start 58
init: starting sh
root@DataLab# hostname 202042005

hostname(3) consumed 98252 bytes, performed 6 context switches
root@202042005# █
```

Finally…

<u>Do NOT hesitate</u> to ask questions!

…but start early