Project 1 Report

Evan French

September 30, 2016

Description

This project familiarized me with the xv6 build environment and these topics:

1. Conditional compilation
2. Implementing system call tracing in the xv6 kernel
3. Implementing a new system call in xv6
4. Control character sequence handling by the xv6 console

Deliverables

The principle deliverables for this project are:

1. Demonstrated knowledge and use of conditional compilation
2. A simple system call tracing facility that is activated by conditional compilation
3. A new system call, **date()**, that demonstrates knowledge of how to implement a new system call in xv6
4. A new shell command, **date** that demonstrates a correct implementation of the **date()** system call.
5. A modification to an existing kernel routine for printing debug information regarding processes and activated by the existing sequence control – P, `procdump()`

Implementation

Trace

To create a trace of every **SYS call** it was suggested to modify the current **sys_call** function. Another suggestion was to create a new dispatch table of syscalls to define the names of each function. Using the previous dispatch table in **syscall.c** as a model I created a new table titled **sysnames**. Then inside the loop that was already being run inside the function **syscall()**,

I added a print statement which prints both the result of the dispatch table, and the return value stored in eax. Because the eax value is stored in the trap frame, we can easily read the return value of every function from **proc-¿tf-¿eax**. In order to achicve conditional compiliation the sysname array as well as the modifications to **syscall()** had to be wrapped in IFDEF/ENDIF statesments. The flag for these statements was given to me as **PRINT_CALLS** and so that is what I used in both the Makefile as well as syscall.c.

Date

To create a shell command who prints the current date and time when called I needed to see which system calls had access to the hardware timers. The recconmended system call was **cmostime()**. In the interest of isolation and security, as well as the goals for this assignment I created a new system call which would take a pointer to an rtc date object and populate it with data. In order to do this from the kernel **argptr()** must be used to verify that the argument passed to the system call is the correct type and belongs to user address space. Once this has been done the **cmostime** function can populate it with data.

Note: Most of this section is left for the student to fill in.

Modified Console Commmand

The xv6 console interprets certain control sequences as commands to the console. One such command, control−p, is used to "dump" the state of all active processes. The control sequence is recognized in `console.c` and a routine in `proc.c`, `procdump()`, is invoked to display information on the console. The routine is implemented in `proc.c` as that is where routines that need to use the process table, `ptable`, are located.

The `procdump` routine was modified to display the following information regarding active processes in xv6

1. put the items in a list beginning here

In addition, the process structure, `proc` defined in `proc.h` was modified by . . .

Testing Methodology

Modified Console Command

The modified console command output is

// put output here

This output demonstrates that the required fields

1. list them here again

are present in the modified console command. This output suffices to demonstrate that the required functionality is present and correctly displayed.

`date` Command

Testing the `date` command is tricky. While I can show that the output of the command is in the correct format, actually showing that the information is correct takes innovation. The method that I chose has four main steps:

1. Run the new xv6 `date` command
2. Quickly exit xv6 using the "control-a x" sequence
3. Issue the Linux `date` command
4. Compare the two outputs

The idea here is to show that the output of my xv6 `date` command produces a reasonable date as compared to the Linux `date` command.

< rest of test here >