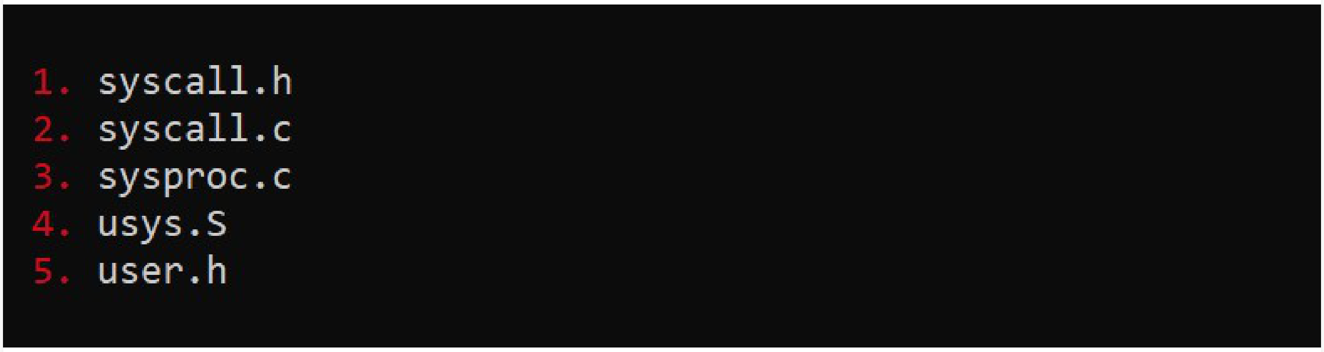
Xv6 Operating System -adding a new system call

Difficulty Level : Medium • Last Updated : 21 Aug, 2020

Prerequisite – Xv6 Operating System -add a user program

In last post we got to know how to add user program in Xv6 Operating System. Now here you will see how to add new system call in Xv6 Operating System.

Adding new system call to xv6 :
A system call is way for programs to interact with operating system. A computer program makes system call when it makes request to operating system's kernel. System calls are used for hardware services, to create or execute process, and for communicating with kernel services, including application and process scheduling.
Overview :
You are going to implement system call called getyear which will return 1975 from kernel always. Unix version 6 was released in that year. In order to define your own system call in xv6, you need to make changes to 5 files. Namely, these files are as follows.
In order to define your own system call in xv6, you need to make changes to 5 files. Namely, these files are as follows.



```
1. syscall.h
2. syscall.c
3. sysproc.c
4. usys.S
5. user.h
```

Create system call to return year Unix version 6 was released :

You could start working from syscall.h file where number is assigned to every system call in this Xv6 system. As you can see, there are 21 system calls already defined in this file. Let’s go ahead and add following line to reserver system call number for your own system call.

```
#define SYS_getyear 22
```

Next, you need to add pointer to system call in syscall.c file. This file contains an array of function pointers which uses above-defined numbers (indexes) as pointers to system calls which are defined in different location. In order to add your custom system call, add following line to this file.

```
[SYS_getyear] sys_getyear
```

What changes happen here ?

This means, when system call occurred with system call number 22, function pointed by function pointer sys_getyear will be called. So, you have to implement this function. However, this file is not place you are going to implement it. You will just put function prototype here inside this file.

So, find suitable place inside this file and add following line. You can see that all other 21 system call functions are defined similarly.

e function prototype which needs to be added to syscall.c file is as follows.

```
extern int sys_getyear(void)
```

Next, you will implement system call function. In order to do this, open sysproc.c file where system call functions are defined.

```
//return the year of which
//Unix version 6 was released
```

Related Articles

```
return 1975;
}
```

Now you have just two little files to edit and these files will contain interface for your user program to access system call. Open file called usys.S and add line below at the end.

```
SYSCALL(getyear)
```

Next,open file called user.h and add following line. This is function that user program will be calling. As you know now, there’s no such function implemented in system. Instead, call to below function from user program will be simply mapped to system call number 22 which is defined as SYS_getyear preprocessor directive. The system knows what exactly is this system call and how to handle it.

```
int getyear(void);
```

If you have completed all above procedure, you have successfully added new system call to xv6. However, in order to test functionality of this, you would need to add user program which calls this system call.

The user program could be as follows:

```
#include "types.h"
#include "stat.h"
#include "user.h"

int main(void)
{
printf(1, "Note: Unix V6 was released in year %d\n", getyear());
```

```
    exit();
}
```

In order to add this user program to xv6, you need to follow above steps for adding user program. At last, run user program in qemu window which can be obtained by running command make qemu on terminal.

After executing everything successfully, you will get on terminal.
Output:

```
"Note:Unix V6 was released in year 1975"
```

Attention reader! Don't stop learning now. Get hold of all the important CS Theory concepts for SDE interviews with the **CS Theory Course** at a student-friendly price and become industry ready.

Like 0

Next

Xv6 Operating System -add a user program

RECOMMENDED ARTICLES

Page : 1 2 3

01 Xv6 Operating System -add a user program
11, Aug 20

02 Difference between system call and library call
10, Jun 21

03 Remote Procedure Call (RPC) in Operating System

05 Implementation of sleep (system call) in OS
01, Jul 21

06 Linux system call in Detail
20, Jun 21

07 System Protection in Operating System
21, Aug 19

30, Aug 17

04 Introduction of System Call
05, Jan 18

08 User View Vs Hardware View Vs System View of Operating System
23, Sep 19

Article Contributed By :

sambhav228
@sambhav228

Vote for difficulty

Current difficulty : Medium

Easy Normal Medium Hard Expert

Article Tags : Operating Systems

Practice Tags : Operating Systems

Improve Article Report Issue

Writing code in comment? Please use ide.geeksforgeeks.org, generate link and share the link here.

Load Comments

5th Floor, A-118,
Sector-136, Noida, Uttar Pradesh - 201305

feedback@gee ▲ eeks.org