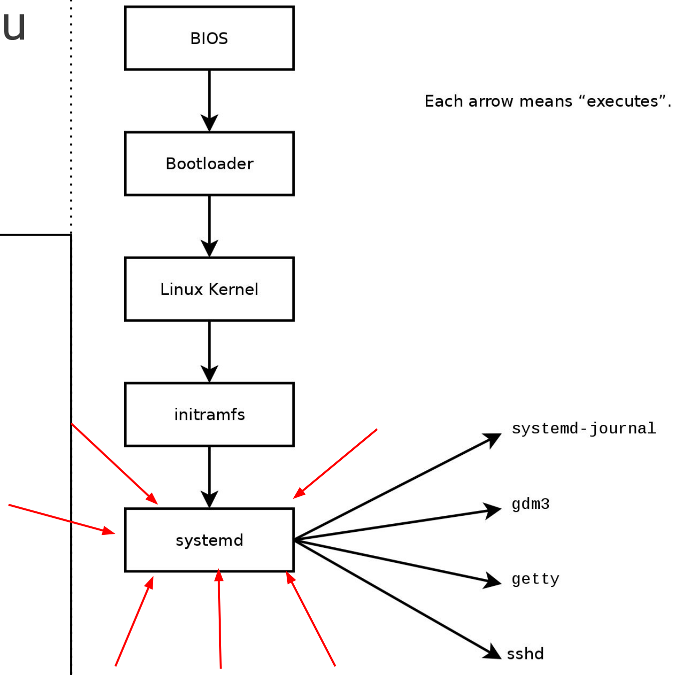
timedatectl: set-time

Alex Richardson & Joshua Patterson

timedatectl

The `timedatectl` command in Linux allows you to query and change the system clock and its settings. It comes as part of systemd

```
$ timedatectl

  Local time: Thu 2018-09-21 16:08:56 CEST

  Universal time: Thu 2018-09-21 14:08:56 UTC

  RTC time: Thu 2018-09-21 14:08:56

  Time zone: Europe/Warsaw (CEST, +0200)
```



Our Planned Modification:

```
$ timedatectl --12hr

>> 2 MAY 2019 4:20 pm




$ timedatectl settime “2 MAY 2100 1:00 pm”

>> 2 MAY 2100 1:00 pm
```

- We wanted to add a way to view the time in 12hr format
- We wanted to add a way to change the system time in 12hr format

Creating the Module

Issues:

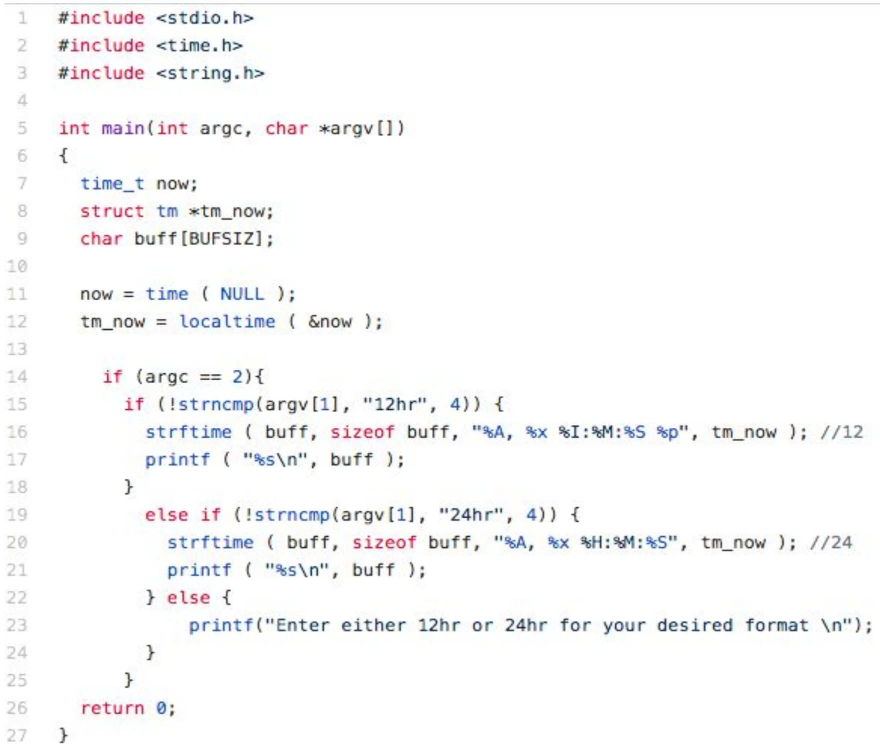
-A kernel module is not an application so there is no `main()` function.

-There is no printf function, just printk

-formatting issues

conversion specifications

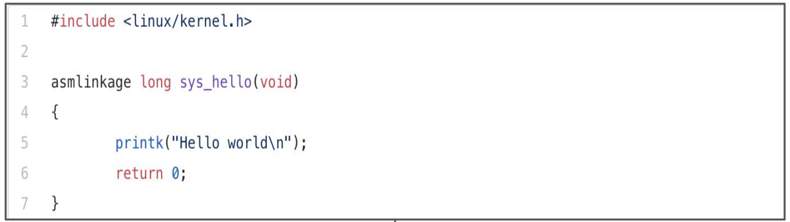
%A = Day of the week	%S = Seconds
%x = Month, Day, Year	%p = AM/PM indicator
%I = 12 Hour “Hour”	
%M= Minutes	



```c
#include <stdio.h>
#include <time.h>
#include <string.h>

int main(int argc, char *argv[])
{
  time_t now;
  struct tm *tm_now;
  char buff[BUFSIZ];

  now = time ( NULL );
  tm_now = localtime ( &now );

    if (argc == 2){
      if (!strncmp(argv[1], "12hr", 4)) {
        strftime ( buff, sizeof buff, "%A, %x %I:%M:%S %p", tm_now ); //12
        printf ( "%s\n", buff );
      }
        else if (!strncmp(argv[1], "24hr", 4)) {
          strftime ( buff, sizeof buff, "%A, %x %H:%M:%S", tm_now ); //24
          printf ( "%s\n", buff );
        } else {
          printf("Enter either 12hr or 24hr for your desired format \n");
        }
      }
  return 0;
}
```

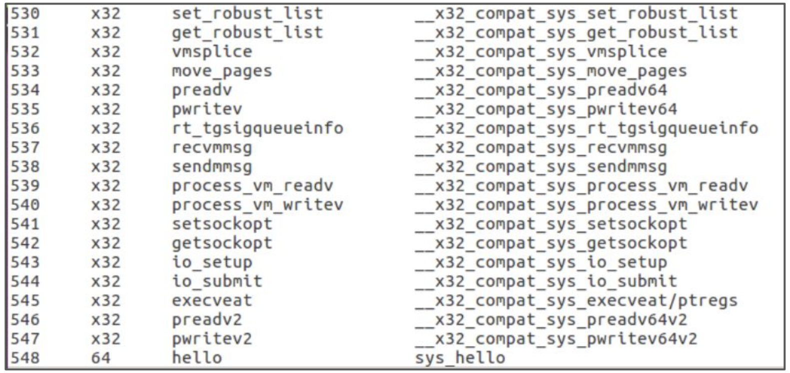
The **strftime**() function formats the broken-down time *tm* according to the format specification *format*

Creating a Hello World Module



```c
#include <linux/kernel.h>

asmlinkage long sys_hello(void)
{
        printk("Hello world\n");
        return 0;
}
```

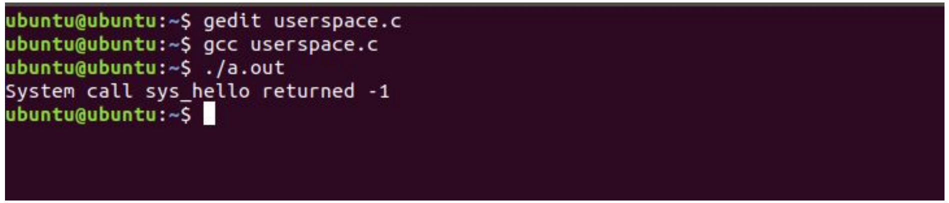
Added to system call table



```
530     x32     set_robust_list         __x32_compat_sys_set_robust_list
531     x32     get_robust_list         __x32_compat_sys_get_robust_list
532     x32     vmsplice                __x32_compat_sys_vmsplice
533     x32     move_pages              __x32_compat_sys_move_pages
534     x32     preadv                  __x32_compat_sys_preadv64
535     x32     pwritev                 __x32_compat_sys_pwritev64
536     x32     rt_tgsigqueueinfo       __x32_compat_sys_rt_tgsigqueueinfo
537     x32     recvmmsg                __x32_compat_sys_recvmmsg
538     x32     sendmmsg                __x32_compat_sys_sendmmsg
539     x32     process_vm_readv        __x32_compat_sys_process_vm_readv
540     x32     process_vm_writev       __x32_compat_sys_process_vm_writev
541     x32     setsockopt              __x32_compat_sys_setsockopt
542     x32     getsockopt              __x32_compat_sys_getsockopt
543     x32     io_setup                __x32_compat_sys_io_setup
544     x32     io_submit               __x32_compat_sys_io_submit
545     x32     execveat                __x32_compat_sys_execveat/ptregs
546     x32     preadv2                 __x32_compat_sys_preadv64v2
547     x32     pwritev2                __x32_compat_sys_pwritev64v2
548     64      hello                   sys_hello
```

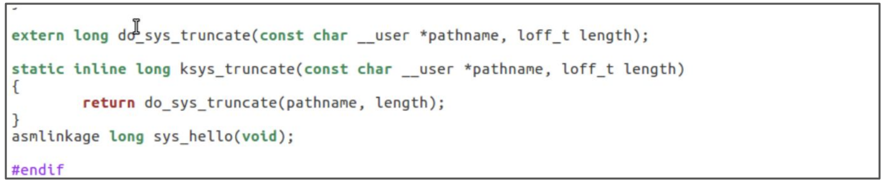
Potential Issues from accessing a system call from userspace:

1. Passing control to the particular point in kernel with switching processor from user mode to kernel mode and returning it back switching processor back to the user mode.
2. Specifying of id of the requested kernel service.
3. Passing of parameters for the requested service.
4. Capturing the result of the service.



```
ubuntu@ubuntu:~$ gedit userspace.c
ubuntu@ubuntu:~$ gcc userspace.c
ubuntu@ubuntu:~$ ./a.out
System call sys_hello returned -1
ubuntu@ubuntu:~$ 
```

Added to system call header



```c
extern long do_sys_truncate(const char __user *pathname, loff_t length);

static inline long ksys_truncate(const char __user *pathname, loff_t length)
{
        return do_sys_truncate(pathname, length);
}
asmlinkage long sys_hello(void);

#endif
```

[redacted]