

Problem statement 2: System Call to Concatenate Strings

Question:

You need to write a new system call in the Linux kernel that concatenates two strings provided by the user and returns the concatenated result. The task involves:

- 1. Writing the Kernel Module to Implement the New System Call:**
 - **Implement the system call handler that takes two strings as input and returns their concatenation.**
- 2. Writing a User-Space Program:**
 - **Utilize this new system call to concatenate two user-provided strings and print the result.**

Requirements:

- 1. Kernel Module (System Call):**
 - **Implement the system call in the kernel.**
 - **Define a new system call number.**
 - **Handle string copying from user space to kernel space and vice versa.**
 - **Concatenate the strings and return the result.**
- 2. User-Space Program:**
 - **Accept two strings from the user.**
 - **Call the new system call with the provided strings.**
 - **Print the concatenated result.**

Steps involved in system call implementation:

- First we need to install / download the kernel source

Using command - `sudo tar -xvf linux-6.9.2.tar -C /usr/src/`

`tar`- Tar stores and extracts files from a tape or disk archive.

`-x` ->extract files from an archive

`-v` ->requested using the `-verbose` option, when extracting archives

`-f` ->file archive; use archive file or device archive

`-C` ->extract to the directory specified after it. (in this case `/usr/src/`)

Now, change the directory to where the files are extracted: `cd /usr/src/linux-6.9.2`

- Here creating a new directory for syscall which has a kernel file that concatenate the string.

Using - `mkdir string_concat`

Then create a kernel code program so that it send concatenated string to the user using `gedit` / `vim`.

Using command - `gedit string_concat/string.c`

The code that I had written is

```
#include <linux/kernel.h>
#include <linux/syscalls.h>
#include <linux/printk.h>
SYSCALL_DEFINE0(string_concat)
{
    printk(KERN_INFO "concatng Successfull\n");
    char destination_string[100];
    char source_string1[] = "Hello, ";
    char source_string2[] = "world!";
    int combined_length = snprintf(destination_string, sizeof(destination_string), "%s%s",
    source_string1, source_string2);
    if (combined_length >= 0) {
        printk(KERN_INFO "Concatenated string: %s\n", destination_string);
    } else {
        printk(KERN_ERR "Error during string concatenation\n");
    }
    return 0;
}
```

Here we are writing an API `SYSCALL_DEFINE0(string_concat)` which takes '0' arguments and does the concatenate part and I used '`snprintf`' which is used especially when dealing with limited buffer sizes.

- Creating a makefile for the kernel code using `gedit string_concat/Makefile`

Having a simple command in that `obj-y :=string.o`

- Now we have to include the system call prototype into `syscalls.h` and give an entry into the syscall table as follow

Command -> `sudo gedit include/linux/syscalls.h`

In this declaration will be taken place by writing `asmlinkage long sys_string_concat(void);`

Command for entry in table -> `sudo gedit arch/x86/entry/syscalls/syscall_64.tbl`

In this at somepoint below write as given below

```
index_number common string_concat sys_string_concat
```

- Now its time for the original kernel Makefile use the command `sudo gedit Makefile`
In this you can find "core-y =" There we have to assign our subdirectory just make it
Like this

```
ifeq ($(KBUILD_EXTMOD),)
"edit this line core-y:= put your custom folder name "
core-y                := helloworld/

drivers-y             :=
libs-y                := lib/
endif # KBUILD_EXTMOD
```

- Now use the command `sudo make menuconfig` ...where to configure the options and functionalities of kernel before building it.
- Now you have to disable some trusted keys and revocation keys because ...By disabling that It wont check/verify the revocation status of keys used to sign software updates.
- As kernel maintains list of revoked public keys used by signing software packages. when a key is comprised or deemed untrustworthy it gets added to revocation list.
- By checking revocation list during software update your system can ensure the authenticity of packages being installed. So packages signed with revocation key are rejected
This can be done using following commands :

```
sudo scripts/config --disable SYSTEM_REVOCATION_KEYS
sudo scripts/config --disable SYSTEM_TRUSTED_KEYS
```

- Now type `sudo make -j4`...to build the software
Here -j represents no. of parallel processes to use in building the software
And 4 represents make use of 4 processes.
- `sudo make modules_install install -j4` for installing modules
- `sudo update-grub` For updating the GRUB(Grand Unified Bootloader).The GRUB will be responsible for displaying the boot menu when you start your compuer and allowing you to choose which OS to boot into.
- Now its time to restart your machine or reboot to check if your syscall is working

Use this command `sudo reboot` Then your system gets restarted and your can check your syscall .

- After reboot you can create a directory in Home and write a user code where we can check the syscall is implemented or not. The user code is as follows

```

#include<linux/kernel.h>
#include<sys/syscall.h>
#include<stdio.h>
#include<unistd.h>
int main(){
    long int i=syscall(463);
    if(i==0){
        printf("Syscall successfull\n");
    }else{
        printf("Syscall unsuccessful\n");
    }
    return 0;
}

```

Here '463' represents the syscall line number in syscall table

- So after writing the user code compile using gcc string_concat.cThen an executable will be created by the name "a.out".
- Now time to run the executable and check the "dmesg" for the successful concatenation of our strings.

Block diagram would be as follows:

