Demo

1. Inserting/Removing new module to Linux kernel

Step 1: lsmod

List all kernel modules that are currently loaded. Output in three columns name, size and where the module is being used.

dmesg: To check kernel log buffer

sudo dmesg -c: To clear the kernel buffer

The following program (named simple.c, available with the source code and Makefile) illustrates a very basic kernel module that prints appropriate messages when the kernel module is loaded and unloaded.

```
#include <linux/init.h>
#include <linux/kernel.h>
#include <linux/module.h>

/* This function is called when the module is loaded. */
int simple_init(void)
{
   printk(KERN_INFO "Loading Kernel Module\n");

   return 0;
}

/* This function is called when the module is removed. */
void simple_exit(void)
{
   printk(KERN_INFO "Removing Kernel Module\n");
}

/* Macros for registering module entry and exit points. */
module_init(simple_init);
module_exit(simple_exit);

MODULE_LICENSE("GPL");
MODULE_DESCRIPTION("Simple Module");
MODULE_AUTHOR("SGG");
```

Step 2: make

Builds the project

The file **simple.ko** represents the compiled kernel module.

Kernel modules are loaded using the **insmod** command

Step 3: sudo insmod simple.ko

Step 4: lsmod | grep "sim*"

To check whether the module has loaded, enter the lsmod command and search for the module simple.

Step 5: dmesg

To check whether in kernel buffer we have "Loading Kernel Module"

Removing the kernel module involves invoking the **rmmod** command:

Step 6: sudo rmmod simple

Step 7: lsmod | grep "simple"

We should not have this in list of modules now.

Step 8: dmesg

To check kernel buffer having message "Removing Kernel Module"

Step 9: sudo dmesg -c

Displays and clear the kernel buffer

2. **Basic fork implementation**

Given below is multi_fork.c **Step 1**: gcc multi_fork.c -o fork **Step 2**: ./fork

Guess: How many processes will be created?

```
#include <stdio.h>
#include <unistd.h>

int main()
{
    /* fork a child process */
    fork();

    /* fork another child process */
    fork();

    /* and fork another */
    fork();

    return 0;
}
```

3. Pipes

Given below is unix_pipe.c

```
#include <sys/types.h>
#include <stdio.h>
#include <string.h>
#include <unistd.h>

#define BUFFER_SIZE 25
#define READ_END 0
#define WRITE_END 1

int main(void)
{
   char write_msg[BUFFER_SIZE] = "Greetings";
   char read_msg[BUFFER_SIZE];
   int fd[2];
   pid_t pid;
```

```
    /* create the pipe */
    if (pipe(fd) == -1) {
       fprintf(stderr,"Pipe failed");
       return 1;
    }

    /* fork a child process */
    pid = fork();

    if (pid < 0) { /* error occurred */
       fprintf(stderr, "Fork Failed");
       return 1;
    }

    if (pid > 0) { /* parent process */
       /* close the unused end of the pipe */
       close(fd[READ_END]);

       /* write to the pipe */
       write(fd[WRITE_END], write_msg, strlen(write_msg)+1);

       /* close the write end of the pipe */
       close(fd[WRITE_END]);
    }
    else { /* child process */
       /* close the unused end of the pipe */
       close(fd[WRITE_END]);

       /* read from the pipe */
       read(fd[READ_END], read_msg, BUFFER_SIZE);
       printf("read %s",read_msg);

       /* close the read end of the pipe */
       close(fd[READ_END]);
    }

    return 0;
}
```

Step 1: gcc unix_pipe.c -o pipe **Step 2**: ./pipe

Note: https://drive.google.com/drive/folders/1xB9VVn5XZAg4-wvdPxl7l_c8jv5RMpeK?usp=sharing

Demos can be downloaded from here.

CS3500 - Operating System, August 2022

Lab 0: Introduction

Understand the Linux commands from the supporting reading material and try to do the following using Command Line Interface.

Practice Question 1

- Login to the system with your user account and open the terminal!
- What is the current/working directory just by looking the prompt?
- Check the name of your working directory with pwd command?
- List the content of your home directory
- Create a subdirectory called `Practice` in the directory Home
- Create a file linux.txt using cat command and write some text in it.
- View the file linux.txt with more command! Which effect do the keys CR and SPACE have? Also use a different text editor to view the file.
- Write a hello world C program in Practice folder and execute it.
- Create a copy of the Practice folder as Copy_Practice and change its ownership to just execute for the current user using chmod.
- Remove the Practice folder and rename Copy_Practice as Practice_renamed

Practice Question 2: Searching with grep

- Go on the following page: https://plants.ensembl.org/Oryza_sativa/Info/Index using your internet navigator
- Copy the URL of the rice genome annotation file (gff format, all chromosomes) that we will use to download the file directly on the server
- Go to the Bank directory and type the following command:

```
wget gff_url
```

- After checking the content of your current directory, what have you done with the `wget`command?
- Decompress the gff with the command `gzip -d file.gz`
- Display the firsts and lasts lines of the gff file
- Print the lines with the word `gene` in the gff file
- Count the number of genes
- Search for the nbs-lrr genes
- Count lines without the word “putative”