Linux Kernel Device Driver Development – Part 1

Theoretical material

Practical Implementation

Practical Exercise

Constant update of Course content

Linux Kernel Device Driver Development – Part 1

Course Content

- ❖ Installing the Linux on Virtual machine.
- ❖ Introduction to Linux Kernel
- ❖ Code compilation
- ❖ Makefile creation
- ❖ Hello world Module
- ❖ Module utilities
- ❖ Character Driver
- ❖ Creating /sys and /proc entries using kernel module
- ❖ Submitting your First patch to Linux kernel source.

Code Compilation

Process of converting the high level language to Machine readable language

General Compilation Involves

- Pre-processing : Includes the header files and expands Macros
- Compiling : Pre-processed source code is converted to assembly code
- Assemble : Converts the assembly code to object code
- Linking : All object codes are linked to libraries and executable binary is created.

Final output from the compilation process is architecture dependent code and can be executed only on the architecture it is compiled against.

Pre-processing

- Input file : <filename>.c
- Output file : <filename>.i
- Command :

```
gcc   – E   hello.c   – o    hello.i
```

Compiling

- Input file : <filename>.i
- Output file : <filename>.S
- Command :

```
gcc   – S   hello.i   – o    hello.S
```

Assemble

- Input file : <filename>.S
- Output file : <filename>.o
- Command ::

Click to add text

```
as   hello.S   – o    hello.o
```

```
gcc – c hello.S - o hello.o
```

Linking

- Input file : <filename>.o
- Output file : <filename>
- Command :

```
gcc   hello.o   – o   hello
```

Single Step using GCC

- Input file : <filename>.c
- Output file : <filename>.out
- Command :

```
gcc hello.c –o hello.out
```