## Chapter 3 kernel Facilities

**Kernel Threads**

From the console, type: ps –ef

The kernel threads are surrounded by []:

[kthreadd], [ksoftirqd/0], [khelper], [kblockd], etc.

“**bottom half**”: the second-half of the code within the interrupt handler that can be deferred

On SMP, one instance of ksoftirqd is created per CPU; ksoftirq/n, where n = CPU number.

**Create a kthread**

pid = kernel\_thread(mythreadd, NULL, CLONE\_FS| CLONE\_AND);

CLONE\_SIGHAND: signal handler be shared

**Linked Lists**

// list head defined in <linux/list.h>

struct list\_head {

struct list\_head \*next, \*prev;

};

// define my own data list structure; embed the list\_head as a member

struct my\_list{

struct list\_head list;

int data;

};

// 2 ways to create an instance of my\_list

struct my\_list mylist;

INIT\_LIST\_HEAD(&mylist.list);

OR

LIST\_HEAD(mylist);

struct list\_head \*pos, \*q;

// create a new item to the list

struct my\_list \*tmp;

tmp= (struct my\_list \*)malloc(sizeof(struct my\_list), GFP\_KERNEL);

tmp->data = i + 10;

// add the new item to the head list, mylist

list\_add(&(tmp->list), &(mylist.list));

// traverse the list

struct list\_head \*pos;

list\_for\_each(pos, &mylist.list){

tmp= list\_entry(pos, struct my\_list, list);

printf("to= %d from= %d\n", tmp->data);

}