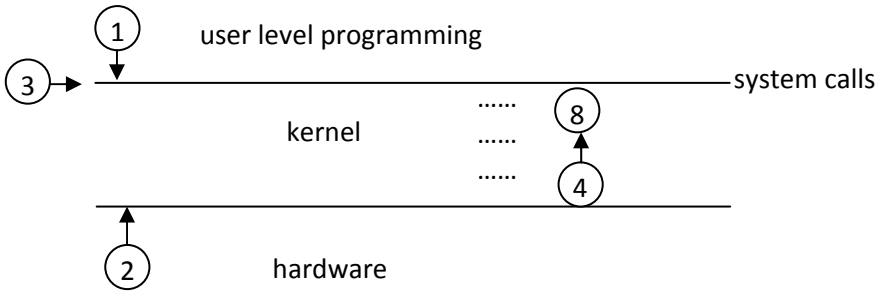
Operating System-Class 1-Jan. 26th

Notes by Weiwei Gong



Topics:

1. Operating system from outside
2. Hardware
3. System call mechanism
4. Processes
5. Kernel synchronization
6. Memory management
7. Interrupts
8. I/O systems
9. Networking
10. Disk

Setup

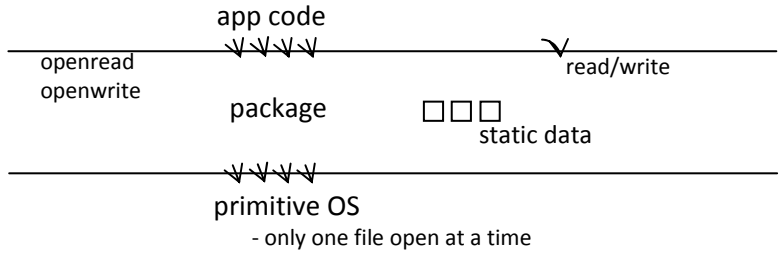
use ulab.cs.umb.edu(Solaris Unix system) and copy ~eoneil/.cshrc.

- has "online" SAPCs
  - stripped-down Linux "Tutor"
- download program written on Unix, then run on SAPC (like kernel environment)

SAPC(our OS): - we download Xinu OS (~64KB)

1). write user level code on Xinu
2). write program directly on SAPC, no Xinu – hardware programming
3). kernel programming: modify Xinu, Linux, etc.

Qual



Timer Example

Timer Package

API for package

void init()
void timeron(int timerno)
void timeroff(int timerno)
int gettime(int timerno)

Ex. for app use:

```
init();
timeron(0);
RUN_APP;
timeroff(0);
int interval = gettime(0);
```

get vs. put:

```
init();
for(…) {
    timeron(0);
    get(…);
    timeroff(0);
    timeron(1);
    put(…);
    timeroff(1);
}
get_time = gettime(0);
put_time = gettime(1);
```

Implementation of Timer Package

End time, on vs. off, time so far
for all timers: old_cpu(as we can do ?time)

```
# define  NTIMES 10;
struct timerdata{
    int runtime;
    int is_on;
}

static int old_cpu;
static struct timedatas[NTIMERS];

void init()
{
     ……
}
void timeron(int timerno)
{
    int i, newcpu, delcpu;
    newcpu = getcpu();
    delcpu = newcpu – old_cpu;
    /* add delcpu to all timers that is on */
    for (i = 0; i < NTIMERS; i++)
        if (timerdatas[i].is_on)
        timerdatas[i].runtime += delcpu;
    old_cpu = newcpu;
    timerdatas[timerno].is_on = TRUE;
}
```

should be in the .c file not .h file, not public

static: hide the data from outside of the file

helper function
static void updatetimer();
static hide this function from outside