

Assignment 8

1.I/O

Please read "Three Easy Pieces" Ch36 <https://pages.cs.wisc.edu/~remzi/OSTEP/fifile-devices.pdf>, and answer the following questions:

- (1) What are the pros and cons of polling and interrupt-based I/O?
- (2) What are the differences between PIO and DMA?
- (3) How to protect memory-mapped I/O and explicit I/O instructions from being abused by malicious user process?

A:(1)The pros of polling is simple and effective but meanwhile it has cons like low efficiency and inconvenience. The pros of interrupt-based I/O is efficient and minimize the CPU cost, but if CPU is of high-performance, then polling will be faster than handling interruption. Besides, interrupt-based I/O is not advised in the Internet.

(2) PIO needs CPU to process I/O request and move the data, but DMA can do data processing itself while CPU schedules another process next. When DMA has done its work, then CPU receives the interruption from DMA and schedules to handle the interruption.

(3) OS will set these I/O operations as privileged and user process perform I/O through OS, which means only OS can communicate with I/O devices.

2.Condition variable

Please implement the condition variable in ucore by the already implemented wait queue or semaphore.

Requirements:

- \1. Please complete the definition of the condition variable in condvar.h
- \2. Please implement the related functions of condition variables in condvar.c
- \3. We have used these functions in check_milk, please make sure your implementation can make check_milk run in valid order. (Please release annotations of check_milk in init_main in proc.c for testing, and annotate check_sync).

A: design idea: We implement condition variable by semaphore. In the struct of condition variable, sem of semaphore_t type is defined to use function up and down to implement wait and signal later. Besides, count of integer type is defined to record the number of processes blocked at this condition variable. What's more, another condvar type pointer named next is defined to cooperate with current condition variable to implement wait and signal. How to use this condvar type pointer? In function wait, when it's called, the current process needs to be added into waiting queue, so we will awake process blocked at condvar next. In function signal, when it's called, the current process first awake the process blocked at the same condvar then block it self at condvar next, because then the process awoken by current process will later awake the process blocked at its condvar next, which means it will awake the current process in the future.

Screen shot:

```

5
6 typedef struct condvar{
7 //=====your code=====
8     semaphore_t sem;
9     struct condvar *next;
10    int count;
11 } condvar_t;
12

```

```

6 void
7 cond_init (condvar_t *cvp) {
8 //=====your code=====
9     cvp->count = 0;
10    cvp->next = (condvar_t*) kmalloc(sizeof(condvar_t));
11    cvp->next->count = 0;
12    sem_init(&(cvp->sem),1);
13    sem_init(&(cvp->next->sem),0);
14 }
15
16 // Unlock one of threads waiting on the condition variable.
17 void
18 cond_signal (condvar_t *cvp) {
19 //=====your code=====
20     if(cvp->count > 0){
21         cvp->next->count++;
22     }
23     up(&(cvp->sem));
24     down(&(cvp->next->sem));
25     cvp->next->count--;
26 }

```

```

27
28 void
29 cond_wait (condvar_t *cvp, semaphore_t *mutex) {
30 //=====your code=====
31     cvp->count++;
32     if(cvp->next->count > 0){
33         up(&(cvp->next->sem));
34     }else{
35         up(mutex);
36     }
37     down(&(cvp->sem));
38     cvp->count--;
39
40 }

```

```
memory management: default_pmm_manager
physcial memory map:
  memory: 0x08800000, [0x80200000, 0x885fffff].
sched class: stride_scheduler
SWAP: manager = fifo swap manager
++ setup timer interrupts
you checks the fridge.
you eating 20 milk.
sis checks the fridge.
sis waiting.
sis waiting.
Mom checks the fridge.
Mom waiting.
Dad checks the fridge.
Dad eating 20 milk.
Dad checks the fridge.
Dad eating 20 milk.
you checks the fridge.
you eating 20 milk.
you checks the fridge.
you eating 20 milk.
Dad checks the fridge.
sis goes to buy milk...
sis comes back.
sis puts milk in fridge and leaves.
sis checks the fridge.
sis waiting.
Dad tell mom and sis to buy milk
you checks the fridge.
you eating 20 milk.
you checks the fridge.
you eating 20 milk.
```

```
Dad checks the fridge.
Dad eating 20 milk.
Dad checks the fridge.
Dad eating 20 milk.
you checks the fridge.
you eating 20 milk.
you checks the fridge.
Mom goes to buy milk...
Mon comes back.
Mom puts milk in fridge and leaves.
Dad checks the fridge.
Dad eating 20 milk.
Mom checks the fridge.
Mom waiting.
you tell mom and sis to buy milk
Dad checks the fridge.
Dad eating 20 milk.
QEMU: Terminated
```

```
xjn12110714@xjn12110714-virtual-machine:~/Desktop/Assignment8$ █
```

