2010 Summer Answers [OS II]

Question 1

A process can be defined as an instance of a running program, or as the context associated with a program in execution.

a) Present the components of the process context and their roles [10 marks]

- Process ID
- Parent Process ID
- Real user ID – The ID of the user who started this process.
- Priority
- Current Directory – The start directory for looking up relative pathnames.

b) What is a child process and how can it be created? [5 marks]

- A process can create a child process, identical to it, by calling fork() – Unix function. As the kernel creates a copy of the caller, two processes will return from this call.
- The parent and the child will continue to execute asynchronously, competing for CPU time shares.
- Generally users want the child to compute something different than the parent. The fork() command returns the child ID to the parent, while it returns 0 to the child itself. For this reason, fork() is placed inside an if test.
- Example:

```
int i;
if( fork() ) { //must be the parent
        for( i = 0; i < 1000; i++ )
                print( Parent, i );
} else { //must be the child
        for( i = 0; i < 1000; i++ )
                print( Child, i )
}
```

c) Explain the advantage of using threads, pointing out elements of a thread context. [5 marks]

A thread is known as a lightweight process. A process can have one or more threads.

- Threads provide concurrency in a program. This can be exploited in multicore computers.
- Concurrency corresponds to many programs internal structure.

- If a thread calls exit(), the whole process, including all its threads will terminate.
- If a thread is more time consuming than others, all other threads will starve of CPU time.

Question 2

While memory addresses are physical addresses, program addresses are virtual ones. The Memory Management Unit translates physical addresses into virtual addresses.

a) **Explain the three methods used for address translation using graphic presentation [10 marks]**
   - Base registers
     - The virtual address is added to the content of a base register. The result is the physical address.
   - Segmentation
     - In this case, different memory segments store different parts of the program: code, data, and stack. Each segment will have separate base and limit register.
   - Paging
     - Memory is separated into areas of 2^k bytes, called page frames. If virtual memory addresses are n bits in length, there can be 2^{n-k} pages. The first n-k bits of the virtual address point to an entry in a page table. Page table entries hold, along with other information, a page frame number, which added to the last n bits of virtual address, make up a physical address.

b) **A good solution to organise the virtual space is to use tow-level page tables. Considering a 32-bit virtual address and pages of size 2^{11} = 2048 B, show the two-level page table model, commenting on its benefits. [5 marks]**

c) **Present the complete set of fields in a page table entry (PTE). [5 marks]**

Question 3

Memory allocation to process is an important function of the operating system.

a) **Explain the Buddy system of memory allocation. [10 marks]**
   - The buddy memory allocation algorithm's main feature is the splitting of a free block of memory into two equally sized "buddy" blocks, where one is returned for allocation and the other is kept free. These blocks can be combined again at a later stage to form the original sized free block.
   - All free memory blocks have a size of some power of 2. If the smallest block size equal or greater than a memory request cannot be found, a block of twice that size is split into two buddies. The first buddy is offered for allocation.

b) **What is swapping? [5 marks]**
   -

c) **In the context of swapping, present two page replacement policies, pointing out their features. [10 marks]**
   - First In First Out starts from the idea that pages are used for a finite amount of times after which they become "old". The page selected here is the one that has been in

memory the longest. Implementation is done by a queue – all new pages are added to the tail of the queue.

- Second chance is an extension of FIFO: when a page if pulled off the head of the queue, the accessed (A) bit is examined. If its 0, the page is swapped out, else the bit is cleared and the page is reinserted at the tail of the queue.

Question 4

The device driver is the software that runs the device controller.

a) **Consider a scenario where several process, denoted as A, B,...request disk operations. What are the operations executed by the driver to satisfy these requests? [10 marks]**
b) **What is the purpose of the interrupt handler? [5 marks]**