104062203 陳涵宇

1. Problem Set

1. A simplified view of thread states is Ready, Running, and Blocked, where a thread is either ready and waiting to be scheduled, is running on the processor, or is blocked (for example, waiting for I/O). This is illustrated in Figure 9.30. Assuming a thread is in the Running state, answer the following questions, and explain your answer:
a. Will the thread change state if it incurs a page fault? If so, to what state will it change?
b. Will the thread change state if it generates a TLB miss that is resolved in the page table? If so, to what state will it change?
c. Will the thread change state if an address reference is resolved in the page table? If so, to what state will it change?

a. Yes, it will change to 'blocked' state. During that time, the OS will do lots of things such as checking whether the page is really invalid or is in the disk, find a free frame, etc. After all these are done, the process will finally restart and the state is switched from 'blocked' to 'running'.

b. The answer will be different according to the result of finding in the page table. If page fault happens, then the answer is same as (a). If page hit, then the TLB will be updated and the process will keep executing and thus stays in 'running' state.

c. No, since the page needed is already in the main memory, there is no need to do I/O operation, and no state changing, either.

2. What is the copy-on-write feature, and under what circumstances is its use beneficial? What hardware support is required to implement this feature?

When a page is shared by multiple processes, and one of them tries to write it, then the page will be copied. It is beneficial when a child process is supposed to have full access to the parent processes' memory space, like fork(). The required hardware support is the system needs to check if a page is write-protected. If so, then a trap would occur and the OS can solve it.

3. What is the cause of thrashing? How does the system detect thrashing? Once it detects thrashing, what can the system do to eliminate this problem?

Thrashing occurs if the allocation is less than the minimum number of pages that a process needs, then the process will continuously page fault. To detect thrashing, OS can compare the CPU utilization to the level of multiprogramming. If thrashing happens, OS can reduce process in the system to decrease the degree of multiprogramming.