3. (12 points) Page Tables

Suppose you have a 64-bit virtual and physical address space with 4KB pages. For all addresses, the most significant 8 hex digits will always be 0, so just indicate the least significant 8 hex digits. Below, is a six-row portion of the page table. All the blank Valid, Dirty, and Reference bits are 0.

The free pages start at physical address 0x00204 0000 and go up. The OS allocates free pages in ascending order, i.e. page 0x00204, 0x00205, etc.

Below the page table is a sequence of virtual memory accesses. Show the page table after this sequence of memory accesses. The address is the memory address to which the LDUR instruction is reading from or the STUR instruction is writing to; we are not concerned with the memory addresses at which these instructions are located at in this question. For each virtual address, translate it to a physical address and indicate if a page fault occurred with that access.

Leave unused entries in the page table blank. However, for non-blank entries in the page table, *all* columns must be completed; i.e., you must fill in the 0's rather than leave an entry blank if anything else is non-zero on the row. If you want to change an existing value in the page table, cross out the old value and add the new one beside it.

Page Table

Index	V	D	R	Page
00100	1	0	1	00205
00101	1	~~0~~ 1	1	00200
00102	1	0	1	00204
00103	1	1	0	00201
00104				
00105	~~0~~ 1	1	1	~~00202~~ 00206

Instruction	Virtual Address	Physical Address	Page fault?
LDUR	0x00101 340	0x00200340	NO
STUR	0x00103 408	0x00201408	NO
STUR	0x00102 888	0x00204888	YES
LDUR	0x00100 010	0x00205010	YES
LDUR	0x00105 560	0x00206560	YES
STUR	0x00101 108	0x00200108	NO