OS MP2 Preliminary

B08902055 李英華

April 19, 2021

1 Print a Page Table

Explain the output of *vmprint* in terms of the figure and answer questions.

- The PTE bit printed indicates the conversion from virtual address (the left square) to the physical address and the corresponding flags. The physical address store contents such as text and data (in page 0) or file contents which is mapped to the virtual address space.
- Page 0: text and data.
- Page 2: start of stack.
- No, because page 1 is guard page. Guard page's PTE is invalid.

2 Generate a Page Fault

Explain why the page fault in question occurs.

- For memory management, it might take a long time for to allocate and map memory for a large memory request. What's more, some programs allocate more memory than they actually use or allocate memory well in advance of use. Therefore, OS implement lazy allocation. That is, sbrk() doesn't allocate physical memory, but just remembers which user addresses are allocated. When the process first tries to use any given page of lazily-allocated memory, the CPU generates a page fault, which the kernel handles by allocating physical memory and mapping it.
- What is page fault: MMU will transfer the virtual address to physical address. However, the page doesn't actually exist due to lazy allocation, and thus result in page fault.
- Why occurs in this case: When *echo hi*, shell fork child proces. *execcmd*() will then be called and do *malloc*() for the process. In *malloc*(), because it's the first time to call *malloc*(), $p == freep$, and thus call *morecore*(). In *morecore*(), *sbrk*() is called and return pointer p. However, the physical memory is not allocated, thus the line "$hp->s.size = nu;$" eventually cause page fault, if other function access the address according to $hp->s.size$ later.