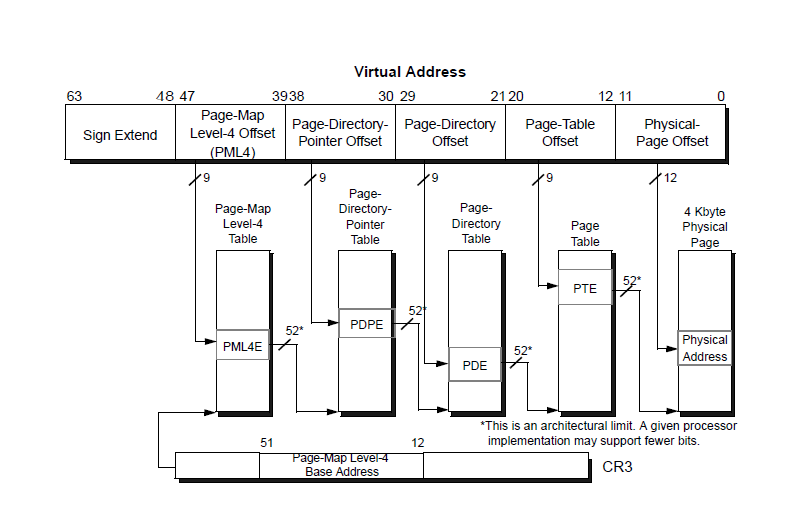
**Project #5. Memory management**

作業系統使用 Paging 機制分隔各個程式使用的記憶體，每個程式能有相同的虛擬記憶體空間 ( Virtual Memory Address )，但對應到不同的實體記憶體位置( Physical Memory Address )，以此達到對記憶體的管理。虛擬與實體的對應就紀錄在 **Page table** 中，而 CPU 中有一個 **CR3** (Control Register 3) 指向 Page table 的所在位置，告訴 CPU 要去何處查表，在 x86\_64 的系統中是透過下圖流程進行轉換。



Example :

**VA 0x5625c4a67456 -> PA 0x234b80456**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **VA (in binary)** | 010101100 | 010010111 | 000100101 | 001100111 | 10001010110 | | | | |
| **Layer** | **PML4 Index** | **PDPT Index** | **PD index** | **PT index** | **Page offset** |
| **Binary** | 0 1010 1100 | 0 1001 0111 |  |  | 1000 1010 110 |
| **Hex** | 0xac | 0x97 |  |  | 0x456 |

**CR3 value : 0x18d396000**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Table Base Address** | **Table Entry Address** | **Entry Content** |
| **PML4** | 0x18d396000 | 0x18d396560 (0xac \* 8) | 0x8000000843ee2067 |
| **PDPT** | 0x843ee2000 | 0x843ee24b8 (0x97 \* 8) | 0x787857067 |
| **PD** | 0x787857000 |  |  |
| **PT** |  |  | (以此類推) |

## Environment Preparation

本次作業請使用上一次 Project #4 的虛擬機器

將作業檔案下載到虛擬機器裡面並執行以下指令

$ make

$ sudo make install

主要內容都在 **main.c** 這個檔案中，裡面有兩個 **TODO** 需要各位同學完成

完成後執行以下指令即可編譯與執行程式

$ gcc -o main main.c

$ ./main

main.c 裡有三個助教提供的 Function ，分別為

獲取 CR3 值：

uint64\_t get\_cr3\_value ()

讀取一**實體記憶體位置**的值：

uint64\_t read\_physical\_address (uint64\_t physical\_address)

將值寫入一**實體記憶體位置**：

void write\_physical\_address (uint64\_t physical\_address, uint64\_t value)

請使用這三個 Function 來完成查找 Page table 與修改 Page table entry

可自行新增 Function，但請勿修改到 main.c 中 TODO以外的程式碼

## Tasks

(40%) 1. **main** 程式中有兩個指標 **x, y** 分別指向兩個字串，透過修改 Page Table Entry (PTE) ，讓 **y** 指向的實體記憶體位置變成 **x** 的位置，此時印出 **y** 的字串內容應與 **x** 相同。

(40%) 2. 將 **y** 指向的實體記憶體位置還原，此時印出 **y** 的字串內容應與最初的字串內容相同

(10％) 3. 將你其中一次查找過程中的各層內容記錄下來（完成下表格）

**VA [ ] -> PA [ ]**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **VA (in binary)** |  | | | | |
| **Layer** | **PML4 Index** | **PDPT Index** | **PD index** | **PT index** | **Page offset** |
| **Binary** |  |  |  |  |  |
| **Hex** |  |  |  |  |  |

**CR3 value : [ ]**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Table Base Address** | **Table Entry Address** | **Entry Content** |
| **PML4** |  |  |  |
| **PDPT** |  |  |  |
| **PD** |  |  |  |
| **PT** |  |  |  |

(10％) 4. 請列舉 3 個你查找到的 entry content 中，除了 Base Address 以外的 Bit 代表什麼意思

程式輸出範例：

Before

x : This is OS homework 5.

y : You have to modify my page table.

After modifying page table

x : This is OS homework 5.

y : This is OS homework 5.

After modifying string y

x : When you modify y, x is modified actually.

y : When you modify y, x is modified actually.

After recovering page table of y

x : When you modify y, x is modified actually.

y : You have to modify my page table.

## Submission Rules

1. 請完成以上問題繳交此檔案，存成PDF並命名為 {student\_id}.pdf 例：0123456.pdf

2. 請繳交修改的程式碼，並命名為 {student\_id}.c 例：0123456.c

**\* 總共上傳兩個檔案，不需要壓縮，檔案名稱未照規定將扣10分**