

**Lab report**

|  |  |
| --- | --- |
| **Course**: | Operating System Principle |
| **Semester**: | 2nd semester of the academic year **2020-2021** |
| **Major**: | Software Engineering |
| **Class**: | 2019 |
| **Student Name**: | 吴嘉诚 |
| **Student ID:** | 222019321062111 |
| **Teacher:** | ZHAO, Hengjun (赵恒军) |

**School of Computer and Information Science**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | | Virtual memory management: paging in x86-64 | | | |
| Date | | May, 2020 | Type | | √ Confirmatory  □Design  √ Comprehensive |
| 1. **Objective & Requirements**    1. Understanding the paging mechanism for virtual memory management    2. Understanding the support for paging in x86-64 architecture with Linux operating system    3. Reviewing the compiling and loading of linux kernel module and the structure of process control block in linux | | | | | |
| 1. **Experimental environment (**platform and software**)**   Virtualbox + Ubuntu | | | | | |
| 1. **Experimental content and design** (Main Content, Procedure, Codes and Results) 2. Tasks for this lab 3. Use lscpu, getconf LONG\_BIT and getconf PAGESIZE to certify that your CPU, operating system is working in 64bit and the page size is set to 4KB. 4. Create a process in which a local integer variable is defined. Then run this process to print out the virtual address of this local variable and the process ID, as well as all the offsets for looking up page tables. Note that a while(1) statement should be appended to the end of this process to keep it running. 5. Compile the cr3 module and load it. Then you will get the physical address of the first level page table. 6. Compile the phy\_mem kernel module and load it into memory. Then compile the read\_phy\_mem program, preparing for reading physical memory. 7. Compile and run read\_physical.c as super user (sudo) with the physical address obtained in c) and the offsets in b) as the input arguments. Then you will get the physical address for the second level page table. 8. Continue the e) step until you reach the final physical address that holding the local integer variable defined in b). If you have performed all the above steps correctly, then the integer value read from this final physical address will be identical to the one defined in b). 9. Thus you finally verify the paging mechanism for virtual memory management under x86-64 with Linux. 10. Please provide your procedure to perform the tasks and source codes. 11. Configuration        1. Compile and run      1. Make and insert module(and pass pid) to get cr3      1. Compile and load into memory      1. Continue reading | | | | | |
| 1. **Result analysis and discussion**（Analysis of experimental results and summing up the harvest and the existing problems） | | | | | |
| Comments & Evaluation | Content & Design (A-E) | | |  | |
| Procedure & Codes (A-E) | | |  | |
| Results (A-E) | | |  | |
| Analysis & Discussion (A-E) | | |  | |
| Score (A-E):  Feedback comments: | | | | |