Homework #7

[ECE30021/ITP30002] Operating Systems

Mission

- Solve problem 1 and 2
  - Solve the problems on Ubuntu Linux (VirtualBox or UTM) using vim and gcc.
  - Problem 1 is a group assignment but Problem 2 is not.
    - Using AI is allowed for Problem 1 but not for Problem 2

- Submission
  - Submit a .tgz file containing hw7_1.c and hw7_2.c on LMS.

    "tar cvfz hw7_<*student_id*>.tgz hw7_1.c hw7_2.c"

    (Do not copy&past but type the above command.)
  - After compression, please check the .tgz file by decompressing in an empty directory.

    "tar xvfz hw7_<*student_id*>.tgz"

- Due date: PM 11:00:00, June 3rd

Honor Code Guidelines

- "과제"
  - 과제는 교과과정의 내용을 소화하여 실질적인 활용 능력을 갖추기 위한 교육활동이다. 학생은 모든 과제를 정직하고 성실하게 수행함으로써 과제에 의도된 지식과 기술을 얻기 위해 최선을 다해야 한다.
  - 제출된 과제물은 성적 평가에 반영되므로 공식적으로 허용되지 않은 자료나 도움을 획득, 활용, 요구, 제공하는 것을 포함하여 평가의 공정성에 영향을 미치는 모든 형태의 부정행위는 단호히 거부해야 한다.
  - 수업 내용, 공지된 지식 및 정보, 또는 과제의 요구를 이해하기 위하여 동료의 도움을 받는 것은 부정행위에 포함되지 않는다. 그러나, 과제를 해결하기 위한 모든 과정은 반드시 스스로의 힘으로 수행해야 한다.
  - 담당교수가 명시적으로 허락한 경우를 제외하고 다른 사람이 작성하였거나 인터넷 등에서 획득한 과제물, 또는 프로그램 코드의 일부, 또는 전체를 이용하는 것은 부정행위에 해당한다.
  - 자신의 과제물을 타인에게 보여주거나 빌려주는 것은 공정한 평가를 방해하고, 해당 학생의 학업 성취를 저해하는 부정행위에 해당한다.
  - 팀 과제가 아닌 경우 두 명 이상이 함께 과제를 수행하여 이를 개별적으로 제출하는 것은 부정행위에 해당한다.
  - 스스로 많은 노력을 한 후에도 버그나 문제점을 파악하지 못하여 동료의 도움을 받는 경우도 단순한 문법적 오류에 그쳐야 한다. 과제가 요구하는 design, logic, algorithm의 작성에 있어서 담당교수, TA, tutor 이외에 다른 사람의 도움을 받는 것은 부정행위에 해당한다.
  - 서로 다른 학생이 제출한 제출물간 유사도가 통상적으로 발생할 수 있는 정도를 크게 넘어서는 경우, 또는 자신이 제출한 과제물에 대하여 구체적인 설명을 하지 못하는 경우에는 부정행위로 의심받거나 판정될 수 있다.

Problem 1

- Mission: peer review of midterm problems as a group
  - Sign up study group
    - Section 01: https://docs.google.com/spreadsheets/d/1fIRP_15xVKxahA6bGDeifoBvW2upYjgkyi6eU0ch4MQ/edit?usp=sharing
    - Section 02: https://docs.google.com/spreadsheets/d/1JovWOmVhmVbF2A74NX_T7uXxEZrC-oRXlgHeK9r53tQ/edit?usp=sharing
  - Have a meeting to review HW#6
    - Review the solutions of other members and share comments to each other.
    - Each member update her/his solution to hw6_2 individually applying the comments.
      - Mark the modified parts by comments as the next page.
      - If you believe your previous solution was perfect, submit it again and put a comment "My previous solution was perfect, so I didn't modify it!" in the first line.

Problem 1

- Example of updated solution.

```
int main()
{
        int i = 0;

        // previous code
        /*
        <old code>

        */

        // updated code
        <new code>


        return 0;
}
```

Problem 2

- Implement the address translation mechanism of the paging system
  - Follow the instructions in hw7_2_problem.c
  - Implement and/or use the following structure and functions

```
typedef struct {
                int pagenum_bits;
                int offset_bits;
                int process_size;

                int page_size;
                int no_page;                // PTLR stores this value
                int offset_mask;

                int *frame;                 // array of frame numbers (f = frame[p]), PTBR store this address
} PageTable;

void PageTable_Create(PageTable *pt, int pagenum_bits, int offset_bits, int process_size); // allocate a page
table
void PageTable_Destroy(PageTable *pt);                 // deallocate page table
int PageTable_Load(PageTable *pt, char *filename);     // load page table from a file
int PageTable_Save(PageTable *pt, char *filename);     // save page table from a file
int PageTable_Init(PageTable *pt, int no_frame);       // randomly initialize a page table

int PageTable_Translate(PageTable *pt, int logical); // translate a logical address into the corresponding
physical address
```

Problem 2

- Example) ./a.out 3 2 16

```
pagenum_bits = 3, offset_bits = 2, process_size = 16
=== Page Table:                                          // the output of PageTable_Display()
pagenum_bits = 3
offset_bits = 2
process_size = 16
offset_mask = 0x0003
no_page = 4
frame[0] = 6
frame[1] = 2
frame[2] = 4
frame[3] = 1
=== PageTable Test:                       // the output of PageTable_Test()
[DEBUG] logical = 0x0000, page_no = 0x0000, offset = 0x0000, frame_no = 0x0006, physical = 0x0018
logical 000000 (0x0000) ==> physical 000024 (0x0018)
[DEBUG] logical = 0x0001, page_no = 0x0000, offset = 0x0001, frame_no = 0x0006, physical = 0x0019
logical 000001 (0x0001) ==> physical 000025 (0x0019)
[DEBUG] logical = 0x0002, page_no = 0x0000, offset = 0x0002, frame_no = 0x0006, physical = 0x001a
logical 000002 (0x0002) ==> physical 000026 (0x001a)
[DEBUG] logical = 0x0003, page_no = 0x0000, offset = 0x0003, frame_no = 0x0006, physical = 0x001b
logical 000003 (0x0003) ==> physical 000027 (0x001b)
[DEBUG] logical = 0x0004, page_no = 0x0001, offset = 0x0000, frame_no = 0x0002, physical = 0x0008
logical 000004 (0x0004) ==> physical 000008 (0x0008)
[DEBUG] logical = 0x0005, page_no = 0x0001, offset = 0x0001, frame_no = 0x0002, physical = 0x0009
logical 000005 (0x0005) ==> physical 000009 (0x0009)
```

Problem 2

```
[DEBUG] logical = 0x0006, page_no = 0x0001, offset = 0x0002, frame_no = 0x0002, physical = 0x000a
logical 000006 (0x0006) ==> physical 000010 (0x000a)
[DEBUG] logical = 0x0007, page_no = 0x0001, offset = 0x0003, frame_no = 0x0002, physical = 0x000b
logical 000007 (0x0007) ==> physical 000011 (0x000b)
[DEBUG] logical = 0x0008, page_no = 0x0002, offset = 0x0000, frame_no = 0x0004, physical = 0x0010
logical 000008 (0x0008) ==> physical 000016 (0x0010)
[DEBUG] logical = 0x0009, page_no = 0x0002, offset = 0x0001, frame_no = 0x0004, physical = 0x0011
logical 000009 (0x0009) ==> physical 000017 (0x0011)
[DEBUG] logical = 0x000a, page_no = 0x0002, offset = 0x0002, frame_no = 0x0004, physical = 0x0012
logical 000010 (0x000a) ==> physical 000018 (0x0012)
[DEBUG] logical = 0x000b, page_no = 0x0002, offset = 0x0003, frame_no = 0x0004, physical = 0x0013
logical 000011 (0x000b) ==> physical 000019 (0x0013)
[DEBUG] logical = 0x000c, page_no = 0x0003, offset = 0x0000, frame_no = 0x0001, physical = 0x0004
logical 000012 (0x000c) ==> physical 000004 (0x0004)
[DEBUG] logical = 0x000d, page_no = 0x0003, offset = 0x0001, frame_no = 0x0001, physical = 0x0005
logical 000013 (0x000d) ==> physical 000005 (0x0005)
[DEBUG] logical = 0x000e, page_no = 0x0003, offset = 0x0002, frame_no = 0x0001, physical = 0x0006
logical 000014 (0x000e) ==> physical 000006 (0x0006)
[DEBUG] logical = 0x000f, page_no = 0x0003, offset = 0x0003, frame_no = 0x0001, physical = 0x0007
logical 000015 (0x000f) ==> physical 000007 (0x0007)
```

Problem 2

- pagetable_example.txt

```
3          // bits for page number
2          // bits for offset
16         // process size
5 6 1 2    // page table (frame numbers)
```

Problem 2

- Example)./a.out 3 2 16 pagetable_example.txt // load page table from 'pagetable_example.txt'

```
pagenum_bits = 3, offset_bits = 2, process_size = 16
Loading page table from file pagetable_example.txt
=== Page Table:
pagenum_bits = 3
offset_bits = 2
process_size = 16
offset_mask = 0x0003
no_page = 4
frame[0] = 5
frame[1] = 6
frame[2] = 1
frame[3] = 2
=== PageTable Test:
[DEBUG] logical = 0x0000, page_no = 0x0000, offset = 0x0000, frame_no = 0x0005, physical = 0x0014
logical 000000 (0x0000) ==> physical 000020 (0x0014)
[DEBUG] logical = 0x0001, page_no = 0x0000, offset = 0x0001, frame_no = 0x0005, physical = 0x0015
logical 000001 (0x0001) ==> physical 000021 (0x0015)
[DEBUG] logical = 0x0002, page_no = 0x0000, offset = 0x0002, frame_no = 0x0005, physical = 0x0016
logical 000002 (0x0002) ==> physical 000022 (0x0016)
[DEBUG] logical = 0x0003, page_no = 0x0000, offset = 0x0003, frame_no = 0x0005, physical = 0x0017
logical 000003 (0x0003) ==> physical 000023 (0x0017)
```

Problem 2

```
[DEBUG] logical = 0x0004, page_no = 0x0001, offset = 0x0000, frame_no = 0x0006, physical = 0x0018
logical 000004 (0x0004) ==> physical 000024 (0x0018)
[DEBUG] logical = 0x0005, page_no = 0x0001, offset = 0x0001, frame_no = 0x0006, physical = 0x0019
logical 000005 (0x0005) ==> physical 000025 (0x0019)
[DEBUG] logical = 0x0006, page_no = 0x0001, offset = 0x0002, frame_no = 0x0006, physical = 0x001a
logical 000006 (0x0006) ==> physical 000026 (0x001a)
[DEBUG] logical = 0x0007, page_no = 0x0001, offset = 0x0003, frame_no = 0x0006, physical = 0x001b
logical 000007 (0x0007) ==> physical 000027 (0x001b)
[DEBUG] logical = 0x0008, page_no = 0x0002, offset = 0x0000, frame_no = 0x0001, physical = 0x0004
logical 000008 (0x0008) ==> physical 000004 (0x0004)
[DEBUG] logical = 0x0009, page_no = 0x0002, offset = 0x0001, frame_no = 0x0001, physical = 0x0005
logical 000009 (0x0009) ==> physical 000005 (0x0005)
[DEBUG] logical = 0x000a, page_no = 0x0002, offset = 0x0002, frame_no = 0x0001, physical = 0x0006
logical 000010 (0x000a) ==> physical 000006 (0x0006)
[DEBUG] logical = 0x000b, page_no = 0x0002, offset = 0x0003, frame_no = 0x0001, physical = 0x0007
logical 000011 (0x000b) ==> physical 000007 (0x0007)
[DEBUG] logical = 0x000c, page_no = 0x0003, offset = 0x0000, frame_no = 0x0002, physical = 0x0008
logical 000012 (0x000c) ==> physical 000008 (0x0008)
[DEBUG] logical = 0x000d, page_no = 0x0003, offset = 0x0001, frame_no = 0x0002, physical = 0x0009
logical 000013 (0x000d) ==> physical 000009 (0x0009)
[DEBUG] logical = 0x000e, page_no = 0x0003, offset = 0x0002, frame_no = 0x0002, physical = 0x000a
logical 000014 (0x000e) ==> physical 000010 (0x000a)
[DEBUG] logical = 0x000f, page_no = 0x0003, offset = 0x0003, frame_no = 0x0002, physical = 0x000b
logical 000015 (0x000f) ==> physical 000011 (0x000b)
```