1차과제 출력본

app_oslab.c

```
#include <unistd.h>
#include <stdio.h>

#define my_stack_push 335
#define my_stack_pop 336

int main(){
  int a;
  int index;

  for(index = 1; index <=3; index++){
    a = syscall(my_stack_push, index);
    printf("push: ");
    printf("%d\n", a);
  }

  for(index=3; index>=1; index--){
    a = syscall(my_stack_pop, index);
    printf("pop: ");
    printf("%d\n", a);
  }


  a = syscall(my_stack_push, 1);
  printf("push: ");
  printf("%d\n", a);

  a = syscall(my_stack_push, 2);
  printf("push: ");
  printf("%d\n", a);

  a = syscall(my_stack_push, 2);
  printf("push: ");
  printf("%d\n", a);


  return 0;
}
```

my_stack_syscall.c

```
#include <linux/syscalls.h>
#include <linux/kernel.h>
#include <linux/linkage.h>

#define MAXSIZE 500

int stack[MAXSIZE];
int top = -1;

int is_already_stacked; //the indictor variable which show if there is the same element as input 'a' in the stack
int pop_element; //the poped element
int stack_index; //index variable for 'for loop'

SYSCALL_DEFINE1(oslab_push, int, a){ //my push syscall function

  printk(KERN_INFO "[System Call] oslab_push(): ");
  is_already_stacked =0;

  for (stack_index = 0; stack_index <= top; stack_index++){ //check if there is the same element as input 'a' in the stack
    if(a == stack[stack_index]){
      is_already_stacked = 1;
      break;
    }
  }


  if (is_already_stacked == 0){ //if there is no same elemet, then check whether stack is full
    if (top<MAXSIZE){
      //stack is not full, then push element 'a'.

      top++;
      stack[top] = a;
      printk("push %d\n", a);

      //print stack
      printk("Stack Top -----------------\n");
      for (stack_index=top; stack_index>-1;stack_index--){
        printk("%d\n",stack[stack_index]);
      }

      printk("Stack Bottom --------------\n");
      return a;
    }
    else{
      //stack is full
      printk("Push %d", a);

      printk("Stack is full\n");

      //print stack
      printk("Stack Top -----------------\n");
      for (stack_index=top; stack_index>-1;stack_index--){
        printk("%d\n",stack[stack_index]);
      }

      printk("Stack Bottom --------------\n");



      return a;
    }
  }
  else{
    //element 'a' is in the stack
    printk("Push %d", a);

    printk("element overlap\n");

    //print stack
    printk("Stack Top -----------------\n");
    for (stack_index=top; stack_index>-1;stack_index--){
      printk("%d\n",stack[stack_index]);
    }

    printk("Stack Bottom --------------\n");

    return a;
  }
}

SYSCALL_DEFINE0(oslab_pop){ //my pop syscall function
  printk(KERN_INFO "[System Call] oslab_pop(): ");

  if (top>-1){ //check if the stack is empty

    //pop element
    pop_element = stack[top];
    top--;

    printk("pop %d\n", pop_element);

    //print stack
    printk("Stack Top -----------------\n");
    for (stack_index=top; stack_index>-1; stack_index--){
      printk("%d\n",stack[stack_index]);
    }

    printk("Stack Bottom --------------\n");

    return pop_element;
  }
  else{
    // if stack is empty, then return -1
    printk("stack is empty");

    //print stack
    printk("Stack Top -----------------\n");
    for (stack_index=top; stack_index>-1;stack_index--){
      printk("%d\n",stack[stack_index]);
    }

    printk("Stack Bottom --------------\n");

    return -1;
  }
}
```

result.txt

```
song@song-VirtualBox:~$ sudo ./app_oslab
push: 1
push: 2
push: 3
pop: 3
pop: 2
pop: 1
push: 1
push: 2
push: 2
song@song-VirtualBox:~$

song@song-VirtualBox:~$ sudo dmesg
...(중략)...
[   99.304725] [System Call] oslab_push():
[   99.304726] push 1
[   99.304727] Stack Top -----------------
[   99.304728] 1
[   99.304728] Stack Bottom --------------
[   99.304787] [System Call] oslab_push():
[   99.304806] push 2
[   99.304806] Stack Top -----------------
[   99.304807] 2
[   99.304807] 1
[   99.304808] Stack Bottom --------------
[   99.304810] [System Call] oslab_push():
[   99.304810] push 3
[   99.304811] Stack Top -----------------
[   99.304811] 3
[   99.304811] 2
[   99.304812] 1
[   99.304812] Stack Bottom --------------
[   99.304814] [System Call] oslab_pop():
[   99.304814] pop 3
[   99.304815] Stack Top -----------------
[   99.304815] 2
[   99.304816] 1
[   99.304816] Stack Bottom --------------
[   99.304818] [System Call] oslab_pop():
[   99.304818] pop 2
[   99.304819] Stack Top -----------------
[   99.304819] 1
[   99.304819] Stack Bottom --------------
[   99.304821] [System Call] oslab_pop():
[   99.304821] pop 1
[   99.304822] Stack Top -----------------
[   99.304822] Stack Bottom --------------
[   99.304824] [System Call] oslab_push():
[   99.304824] push 1
[   99.304825] Stack Top -----------------
[   99.304825] 1
[   99.304825] Stack Bottom --------------
[   99.304827] [System Call] oslab_push():
[   99.304828] push 2
[   99.304828] Stack Top -----------------
[   99.304828] 2
[   99.304842] 1
[   99.304843] Stack Bottom --------------
[   99.304845] [System Call] oslab_push():
[   99.304845] Push 2
[   99.304846] element overlap
[   99.304846] Stack Top -----------------
[   99.304846] 2
[   99.304847] 1
[   99.304847] Stack Bottom --------------
song@song-VirtualBox:~$
```