Segmentation and Address Translation

Introduction

This shows the function used to map a logical address to a physical address for some segmentation schemes. Throughout this handout, an address in virtual memory is a pair (*segment*, *offset*) where *segment* is the segment number within the logical address space and *offset* the offset into that segment. We will assume the entire program is in memory, so no error handling is given; were this assumption false, the situation where the requested address were not in memory would need to be handled (by generating a segment fault and loading the necessary segment):

Segmentation

As with paging address translation with direct mapping, the segment table is stored in memory, and a pointer to its base in a register called the segment table base register. Let the segment table base register be called *st_base_register*, and let memory represent the main store of the computer. Then:

```
function NL_map((segment, offset)): physical_address;
begin
      NL_map := memory[st_base_register + segment] + offset;
end (* NL_map *)
```

In pictures: