18 marks **Question 9.** Mapped Files for OS/161

In this question you will outline an implementation of mapped files in OS/161. To support this, we introduce two new system calls:

```
void *mmap(char *filename, int length, int mode);
void *munmap(void *address, int length);
```

The `mmap` call opens the file specified by `filename` (with `mode` specifying if the access is read, write, or both) and creates a mapping between the first `length` bytes in the file and pages in the caller's virtual address space. If successful, it returns the virtual address of the mapped region. The `munmap` call removes the mapping beginning at `address` with `length` bytes.

Once a process has mapped a file, it can read and write the file's data using ordinary memory operations. For example:

```
main() {
    int *values;
    int i;
    values = (int *)mmap(‘‘mydata.txt’’, 100, READ | WRITE);
    if (values != NULL) {
        for (i=0; i < 100; i++)
            values[i] = values[i] + 1;
    }
    munmap(values, 100);
}
```

To simplify the problem, we will assume that a mapped file can only be used by one process at a time.

For each of the following questions, answer them descriptively at a high-level. You do not need to write pseudo-code, though you may if you wish. **This question has 6 parts (a-f). You should read all parts of the question and think about the overall problem before answering each individual part.**

3 marks **Part (a)** Briefly outline the steps needed to add these (or any) new system calls to OS/161.

CONTINUED ON PAGE 12

3 marks **Part (b)** Describe the changes you will make to the `vm_object` and/or `lpage` structure and any new data structures you will add to support mapped files.

3 marks **Part (c)** Describe the basic operation of the function that handles the `mmap` system call. Identify one error that the function should check for.

3 marks **Part (d)** Describe the basic operation of the function that handles the `munmap` system call.

continued on page 13

3 marks **Part (e)** Describe what should be done in `as_fault` and/or `lpage_fault` to handle TLB misses on virtual addresses that are mapped to a file.

3 marks **Part (f)** Describe what should be done in `lpage_evict` to handle the eviction of a page that is mapped to a file.

Total Marks = 100

continued on page 14