Short Examples Demonstrating Relevant API's

During the implementation of you **in-core Pseudo File-System**, you will most likely use the following services of the following Linux library and system calls:

- shm_open()
  File backed shared memory object. This shared object can be mapped into a process address space and manipulated like regular memory. Changes to the memory object are preserved via a **file**. The default location is **/dev/shm/file_name**.
- mmap(), munmap()
  map files (shared memory object) into memory (RAM) of a process. Once a file is mapped into a processes address space, you can manipulate the contents of the shared object similar to regular memory (with pointers).
- struct/union
  - structures allow defining new data types, i.e. grouping multiple data-types under one name.
  - Union's allow *reinterpreting* a block of bytes as a specific type. Usage may be helpful when accessing various sub-fields in blocks (inode-bitmap block, data-bitmap block, data block, inode blocks).
    The size of a union with multiple types is the size to accommodate the largest data type in the union.
- creating a library (an archive)
  - example demonstrating how to create a static library. The static library is linked with other files that want to use functions in the library
- example header file
  This is an example header file that contains some common types that maybe useful when implementing you pseudo file-system.

Demonstration Code

- Shared Memory Objects (shm_open() and mmap()/munmap())
  - **sharedMemory.zip**
- Creating a static library (an archive)
  - **exampleMakeingStaticLibrary.zip**
- Struct vs Union usage

    - example demonstrating using a union (accessing fixed block sizes) vs using a structure. **union_structs.zip**
- starter header file for pseudo file-system
    - **pseudoFileSystem.h**