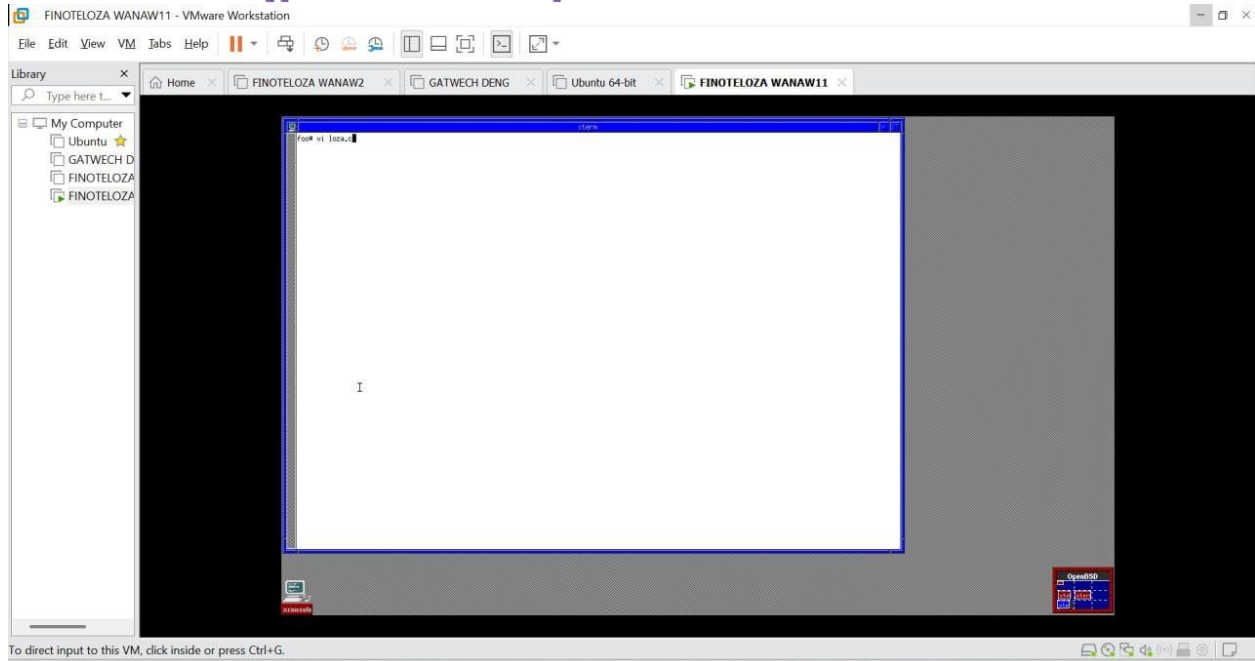


System call implementation

The `madvise()` system call provides the kernel with hints about memory usage patterns, helping optimize system performance.

here I have to type `vi loza.c` to open c-editor

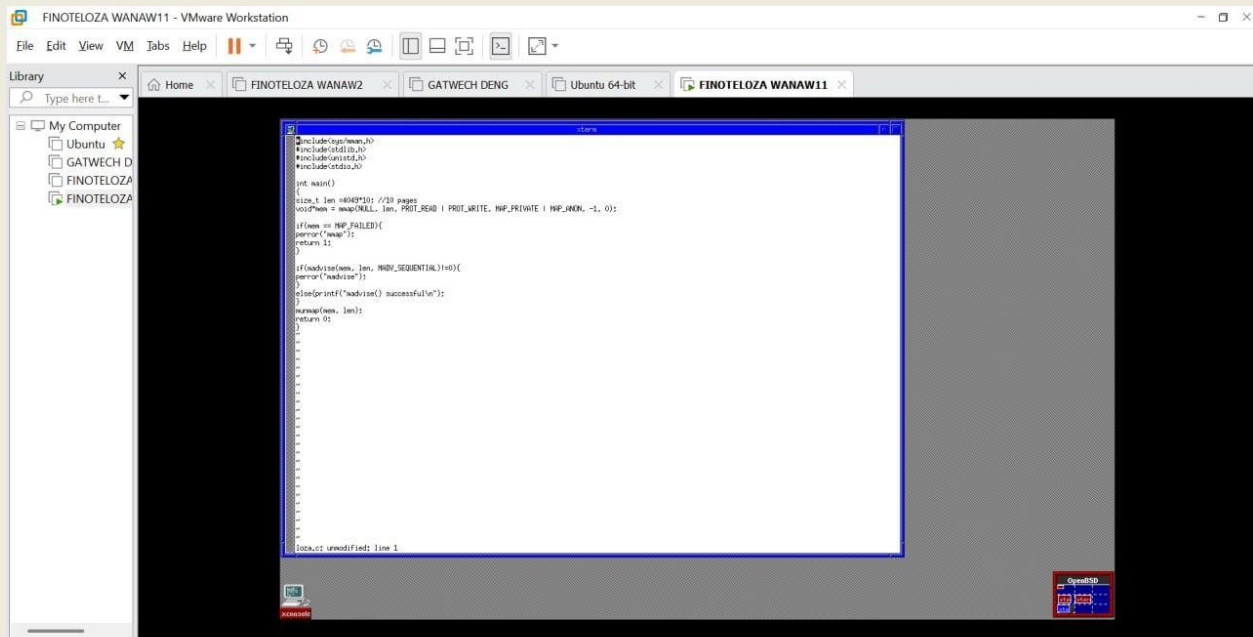


Parameters:

- `addr`: Starting address of the memory region
- `len`: Length in bytes
- `behav`: Expected access pattern (e.g., `MADV_SEQUENTIAL`, `MADV_RANDOM`)

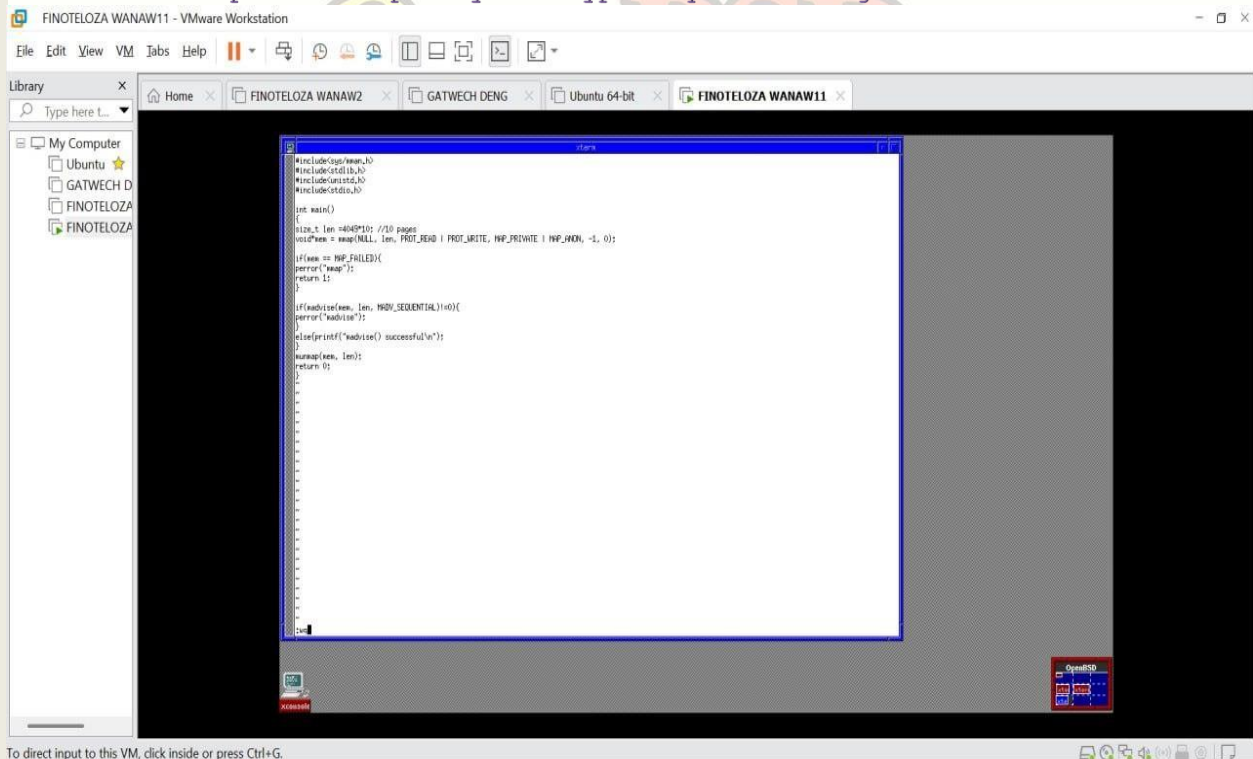
Function Prototype:

```
int madvise(void *addr, size_t len, int behav);
```



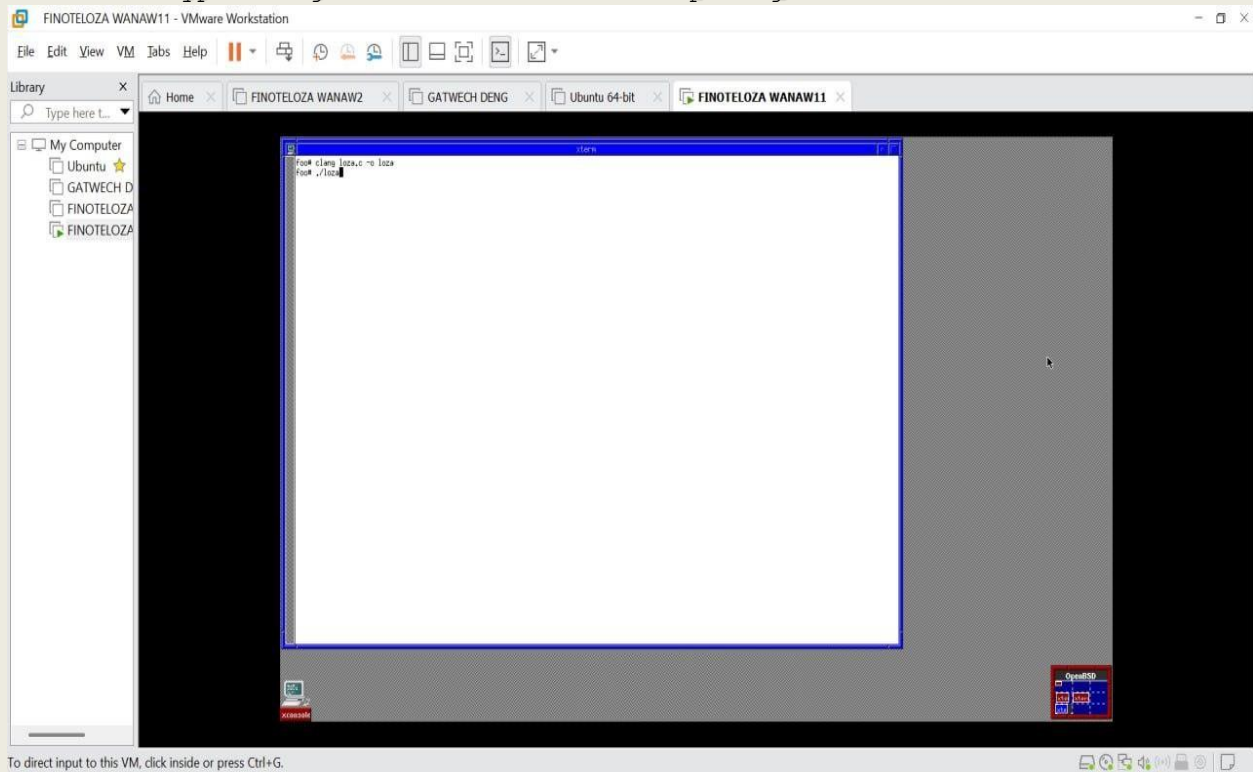
To direct input to this VM, click inside or press Ctrl+G.

Now I have to press escape key and type :wq for saving

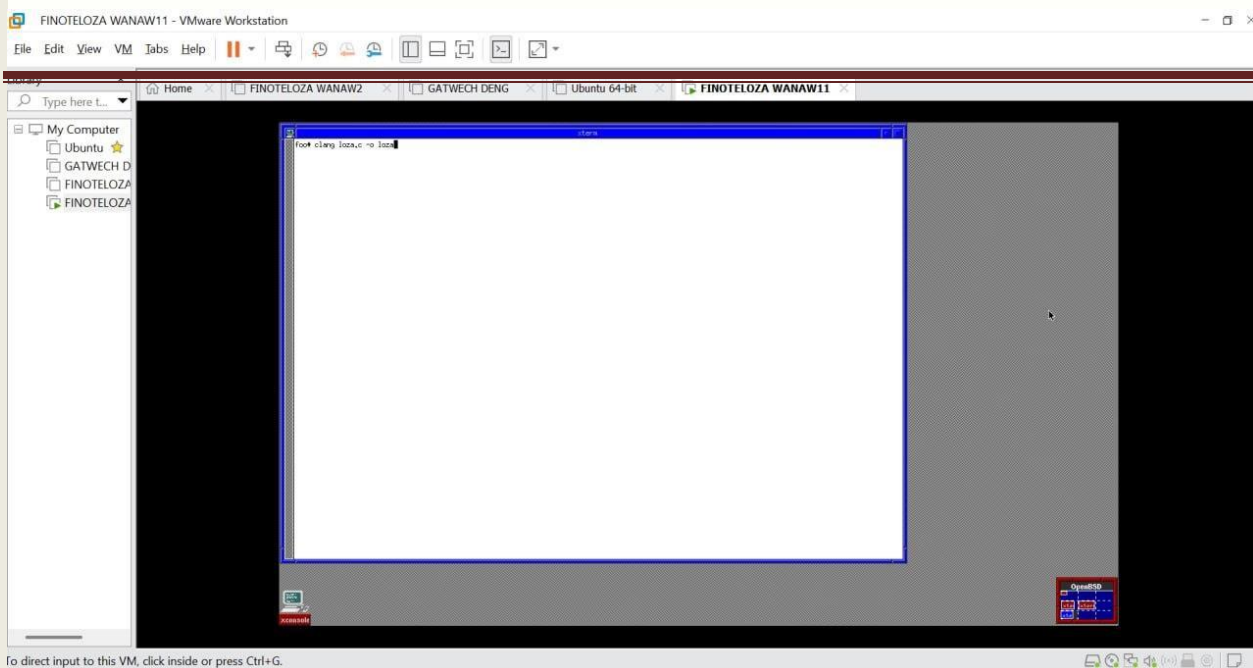


To direct input to this VM, click inside or press Ctrl+G.

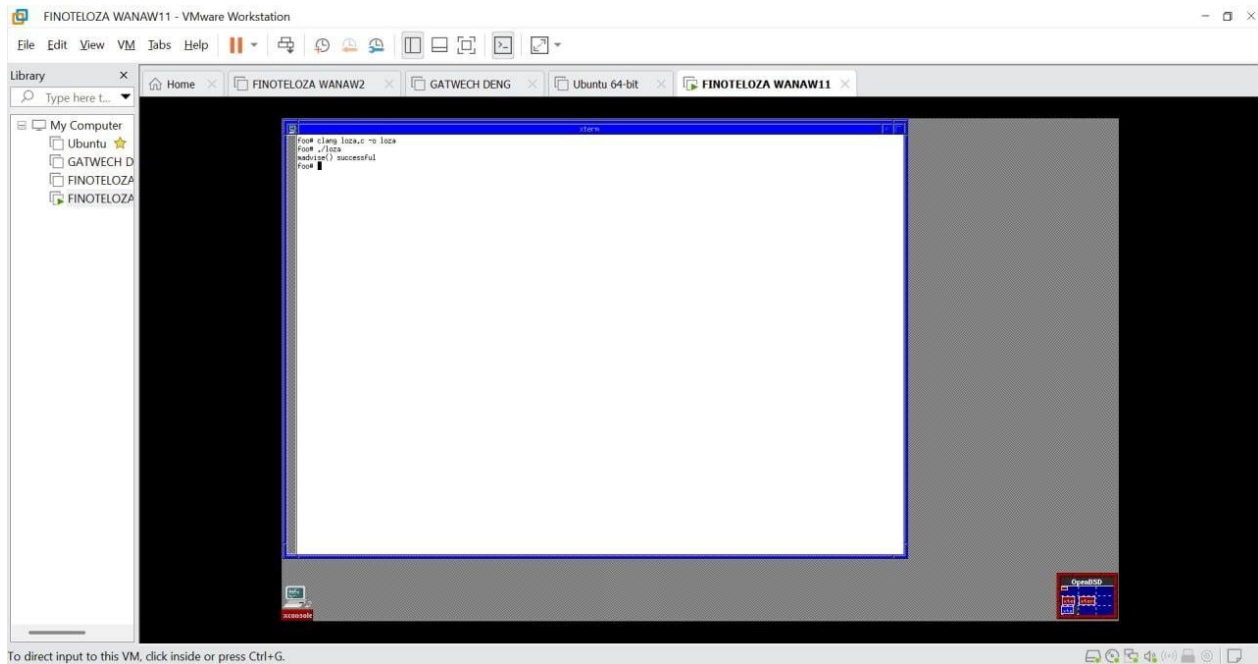
I have to type `clang loza.c -o Loza` for compiling



I have to type `./loza` for running



Now it is successful (madvise() successful).



Internals:

OpenBSD validates memory regions, applies flags, and informs the VM system how to manage memory paging and prefetching more effectively.



OpenBSD