1. Memory Manager (Weight: 70%)

- [x] **Stack Allocation:** Implemented.
  - **Details:** The kernel stack is statically allocated in `boot.S` (16KB). For user processes, a 4KB stack is dynamically allocated using `kalloc` in `process.c` during `process_create`.
- [x] **Heap Allocation:** Implemented.
  - **Details:** A "Bump Allocator" is implemented in `memory.c`. The `kalloc()` function returns a pointer to the `next_free_address` and increments it, ensuring 4KB alignment.
- [ ] **Stack/Heap Deallocation:** Not Implemented.
- [ ] **Optimized Memory Allocation:** Not Implemented.

2. Process Manager (Weight: 70%)

- [x] **Process Table:** Implemented.
  - **Details:** A fixed-size array `pcb_t process_table[MAX_PROC]` is defined in `process.c` to store the context of up to 10 processes.
- [x] **Process Creation:** Implemented.
  - **Details:** The `process_create()` function initializes the PCB, allocates a stack, sets up the initial stack frame (instruction pointer, return address), and marks the state as `READY`.
- [x] **State Transition:** Implemented.
  - **Details:** Processes transition between `READY`, `CURRENT`, and `TERMINATED`. These states are defined in `process.h` and managed within the `schedule()` function in `process.c`.
- [x] **Process Termination:** Implemented.
  - **Details:** A `process_exit()` function is implemented. It sets the process state to `TERMINATED` and calls the scheduler. The return address of every process is set to this function so it is called automatically upon completion.
- [ ] **Utility Functions / IPC:** Not Implemented.

3. Scheduler (Weight: 70%)

- [x] **Clear Scheduling Policy:** Implemented.
  - **Details:** A **Round Robin** policy is strictly followed. The `schedule()` function iterates through the process table `(next_pid + 1) % proc_count` to find the next `READY` task.
- [x] **Context Switch:** Implemented.

    - **Details:** Low-level context switching is handled in assembly (`switch.S`). It saves the callee-saved registers (`ebx`, `esi`, `edi`, `ebp`) and switches the stack pointer (`esp`).
- [ ] **Configurable Time Quantum:** Not Implemented.
    - *Note:* The OS currently uses **Cooperative Multitasking** (via `yield()`), so there is no preemptive time quantum.
- [ ] **Aging:** Not Implemented.