Project Overview - Yalnix Operating System Kernel

The Yalnix Operating System Kernel is a project where we designed and implemented a kernel system from scratch. The kernel is initialized and manages processes, memory, and system calls while handling interrupts and traps. This kernel was designed to simulate real operating system behavior on virtual hardware, and the entire project was implemented in C.

Key features:

1. **Kernel Initialization and Process Management**:
    - The kernel is initialized in *yalnix.c*, which sets up process control blocks (PCBs), page tables, and memory allocation systems. It creates the idle and init processes, which serve as the foundational processes for the system.
2. **Memory Management**:
    - The kernel implements virtual memory using region 1 for kernel operations and region 0 for user processes. Page tables are dynamically managed to allocate and free physical memory as processes run and terminate.
3. **Trap and Interrupt Handling**:
    - The kernel handles various types of interrupts and traps, including clock interrupts, I/O traps, and system calls through the trap vector table. Non-kernel traps are managed in *trap.c*, while system calls are handled in *kernel.c*.
    - Context switching is managed by *MySwitchFunc()* function, we use Round Robbin for process scheduling.
4. **I/O Management**:
    - The kernel manages terminal I/O through buffer structures and interrupts, handling both input and output in a line-buffered manner.
5. **Linked List Structures**:
    - Helper functions in *linked_list.c* manage process queues, memory allocation lists, and other dynamic data structures through custom linked list implementations.

Explanation of Files:

- ***yalnix.c***: Contains the kernel's initialization code, the KernelStart function, and the context switching mechanism.
- ***trap.c***: Handle the Trap/Interrupt handlers for everything besides TRAP_KERNEL (system call).
- ***kernel.c***: Handle the Trap/Interrupt calls that may be specified from a TRAP_KERNEL (system call) interrupt.
- ***helper.c***: Implements helper functions for process creation, memory management, and program loading.
- ***linked_list.c***: Provides queue management and linked list operations for managing processes and memory allocation.
- ***function.h***: Contains the function prototypes, data structures (PCB, memory management, and process queues), and global variables necessary for the kernel's operation.

Remarks:

- This project is designed and provided by Rice University, TX, USA. I completed this project during my exchange.
- The project was a collaborative effort between two people.
- Only the *load.template* and the project_description.pdf are provided for this project. We build everything else from scratch.
- The final score of this project is in the top 5 percent in our class.