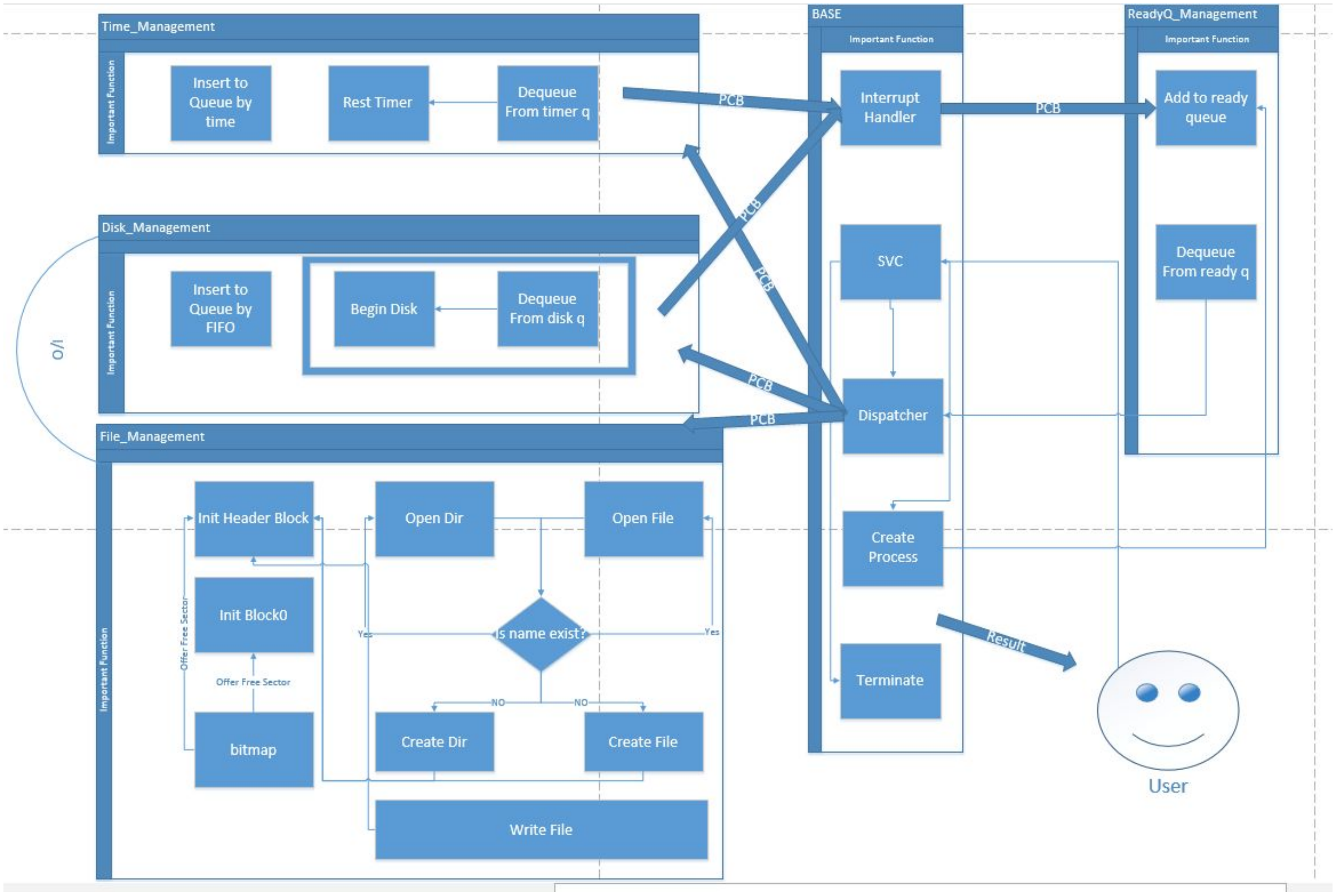
Architectural Document

A . what is included

What have done	Details
Create process and pcb	Put it on ready queue directly
PCB scheduling for timer queue	Insert new pcb by comparing wake time
PCB scheduling for disk queue	FIFO policy, head is currently writing or reading one
PCB scheduling for ready queue	Insert new pcb by comparing priority
Dispatcher	Make processor busy, by using FIFS
File system	Finish z502 file system
Interrupt handler	Capture event and put it from corresponding queue to ready queue
PCB composed of queue	Since each pcb only occur in one of queues, so I use pcb to combine as a queue.

B. High level design



C. Design justification

There are several advantages for this design

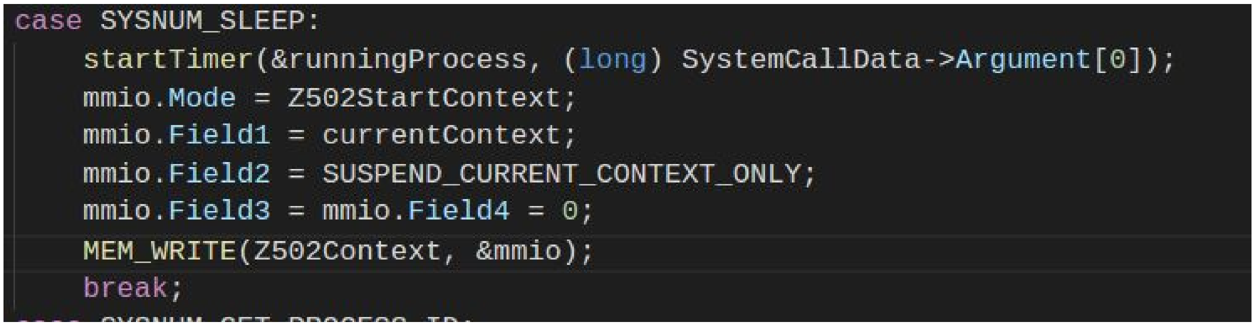
- Concise and easily to manipulate. Basically the each queue is combined by pcb, by using doubly linked list, and there is a empty header for each queue, so it makes dequeue and enqueue easy to manipulate. And Also there is an array to save all the pcb so terminate a process is pretty simple, since pcb could only occur in one of the queue, so just simply remove it from the queue. And mark its state as terminated.
- High cohesion and low coupling, basically in each function block, there is a function to offer the system call function. And in base c, there is hardly some underlying details about the how each system call implemented. That is high cohesion. And also there is no more connections between different system call function, which is low coupling.

D. bug

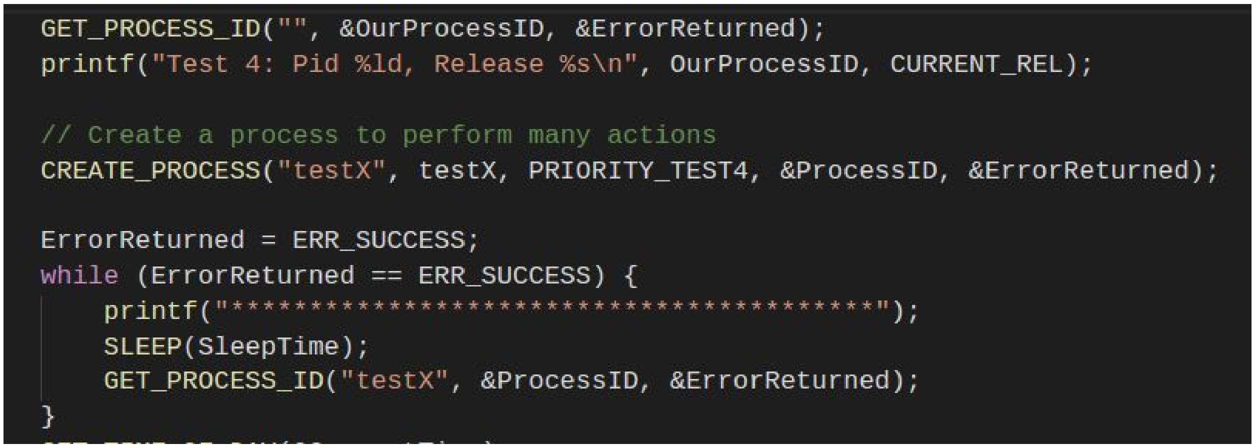
I tried to develop multiprocessor version, however, I developed in linux system, and it seems that start context suspend doesn't work, since after I suspend it, the simulation continue to do things for its test.

Here is screenshot:

I call suspend current context only for test4 in SLEEP system call. However the while loop continue work, and it seems that didn't suspend this current context.



```
case SYSNUM_SLEEP:
    startTimer(&runningProcess, (long) SystemCallData->Argument[0]);
    mmio.Mode = Z502StartContext;
    mmio.Field1 = currentContext;
    mmio.Field2 = SUSPEND_CURRENT_CONTEXT_ONLY;
    mmio.Field3 = mmio.Field4 = 0;
    MEM_WRITE(Z502Context, &mmio);
    break;
```



```
  GET_PROCESS_ID("", &OurProcessID, &ErrorReturned);
  printf("Test 4: Pid %ld, Release %s\n", OurProcessID, CURRENT_REL);

  // Create a process to perform many actions
  CREATE_PROCESS("testX", testX, PRIORITY_TEST4, &ProcessID, &ErrorReturned);

  ErrorReturned = ERR_SUCCESS;
  while (ErrorReturned == ERR_SUCCESS) {
      printf("*****************************************");
      SLEEP(SleepTime);
      GET_PROCESS_ID("testX", &ProcessID, &ErrorReturned);
  }
```