Architectural Document

WHAT is included in the design

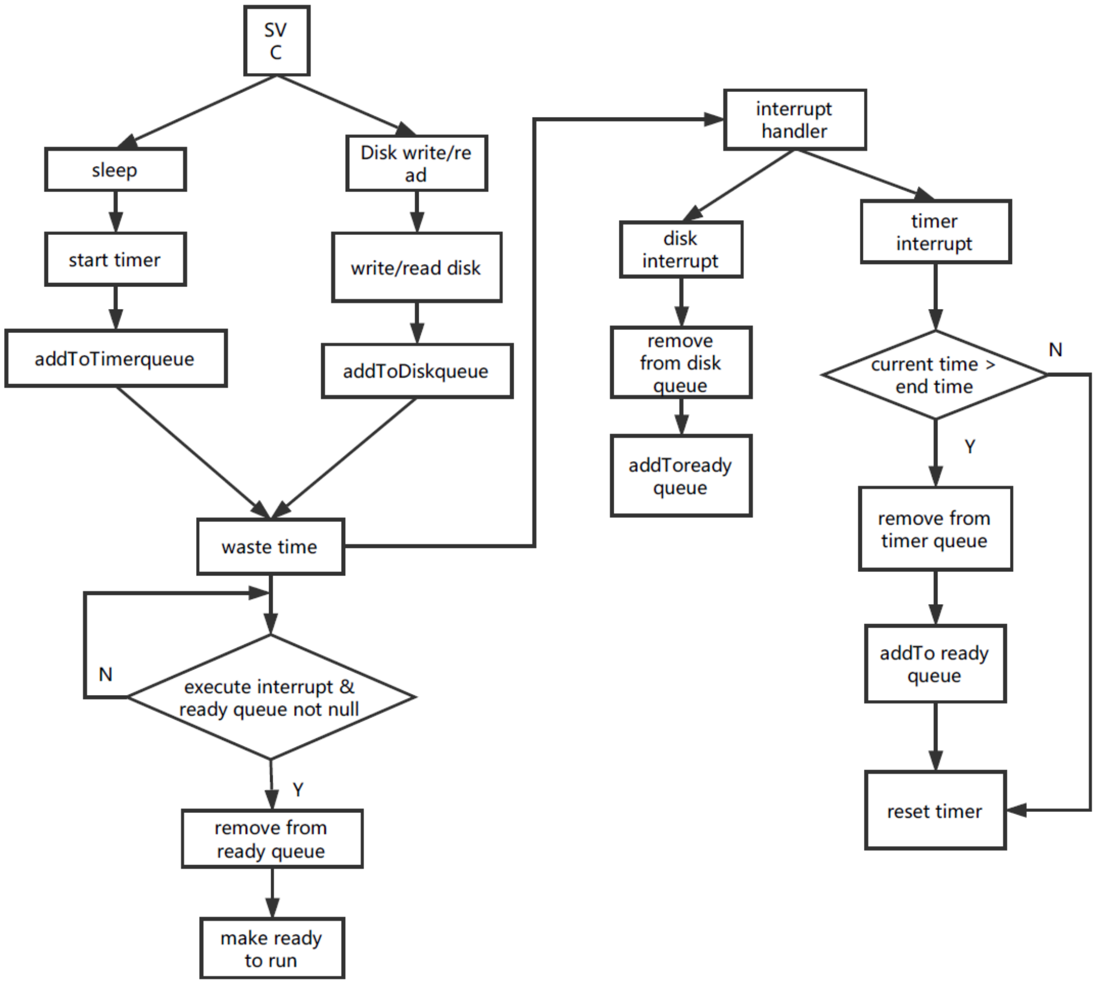
base.c:

Expanded svc to handle the system call. In the interrupthandler to deal with timer interrupt and disk interrupt. Using functions from other source file.

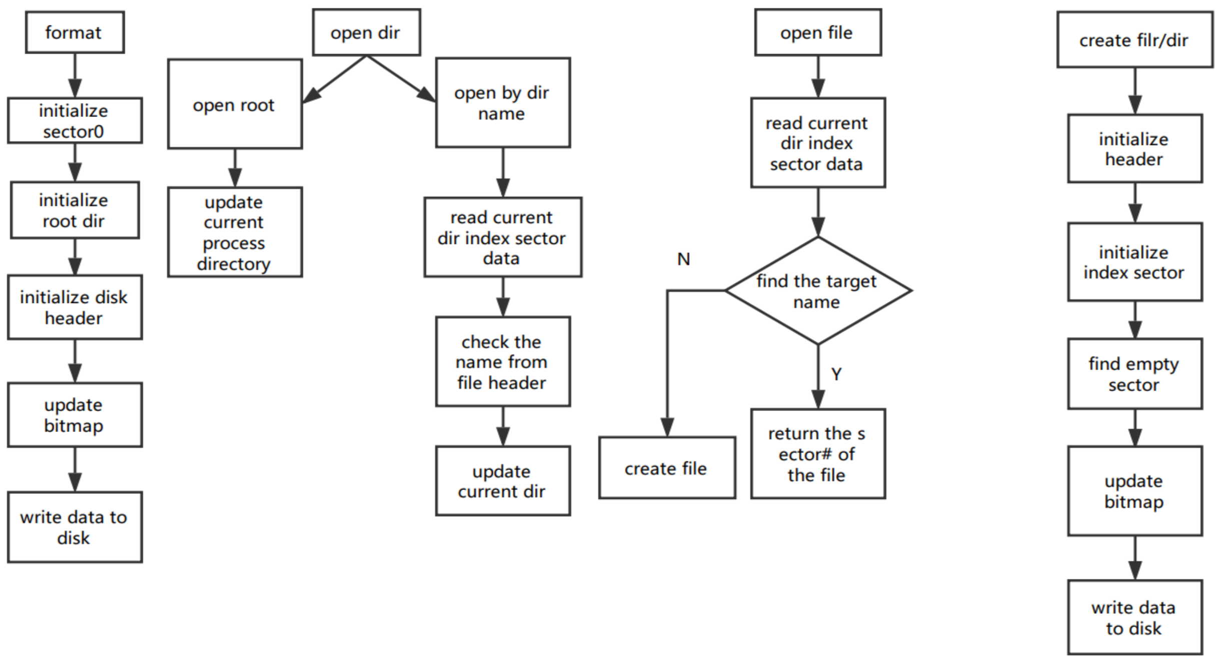
disk.c:
Defined the functions to operate the disk queue. Methods are called in the interrupthandler.
file.c:
Implemented functions related to file system call.
oscreateProcess.c:
Implemented functions related to process queue and timer queue.
printScheduler.c:
Used with the print interface as required. Assigned the value of the SP Data Structure
disk.h:
Defined the structure of disk queue.
file.h:
Defined the structure and union used in file system, including sector0, disk header and index sector.
lock.h:
Defined the variables needed when using lock.
oscreateProdess.h:
Defined the structures of process queue, pcb, and timer queue.
printFullScheduler.h:
Includes the function in printScheduler.c.

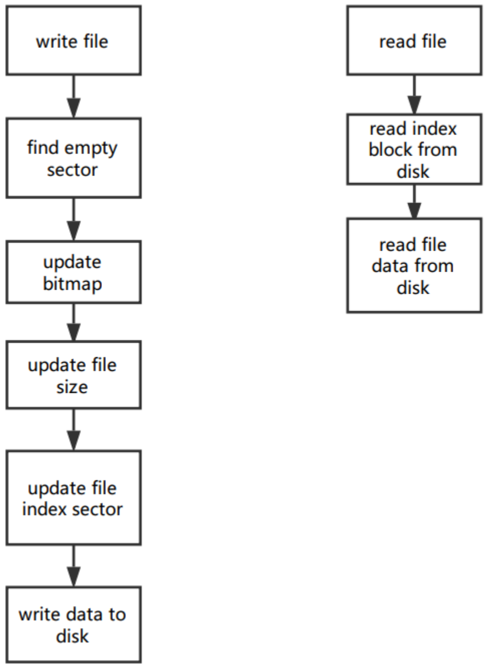
High Level Design: Graph

Interrupt handler


SV
C
sleep
Disk write/re
ad
start timer
write/read disk
addToTimerqueue
addToDiskqueue
waste time
N
execute interrupt &
ready queue not null
Y
remove from
ready queue
make ready
to run
interrupt
handler
disk
interrupt
timer
interrupt
remove
from disk
queue
addToready
queue
current time >
end time
N
Y
remove from
timer queue
addTo ready
queue
reset timer

File system


format
initialize sector0
initialize root dir
initialize disk header
update bitmap
write data to disk
open dir
open root
update current process directory
open by dir name
read current dir index sector data
check the name from file header
update current dir
open file
read current dir index sector data
find the target name
N
Y
create file
return the s ector# of the file
create filr/dir
initialize header
initialize index sector
find empty sector
update bitmap
write data to disk


write file
find empty sector
update bitmap
update file size
update file index sector
write data to disk
read file
read index block from disk
read file data from disk

Justification

a. In the Interrupt handler, the interrupt processing program is defined in different source file to operate the timer queue, disk queue and ready queue. This made the programs more clear and readable.
b. The variables and functions regarding to process is defined in oscreateProcess.c. The header of timer queue, disk queue, ready queue and pcb queue are defined as global variables. The current process is also a global variable. This is reasonable since these variables are frequently used by many routines. The process of dispatch is also defined in oscreateProcess.c, used in svc after sleep, disk write and read.
c. All the functions and variables related to file system are defined in file.c and file.h. Bitmap and header for each disk are defined as global variables. The bitmap will be updated first and write the value of the variable into disk.

Anomalies or bugs

When implemented test 1 I found that the after start timer, the program in the svc after the MEM_WRITE/ MEM_READ function call, and the interrupt handler are executed at the same time.