**Mô tả cài đặt các hàm:**

.

**11. Cài đặt SC\_ReadFloat**

**-** Các bước thực hiện:

1. Xác định số lượng kí tự tối đa mà có thể đọc được từ bàn phím, trong trường hợp này, số lượng kí tự tối đa được đặt là 255.

2. Tạo một buffer có kích thước 255 để lưu trữ dữ liệu đọc từ console, sau đó sử dụng phương thức “Read” của đối tượng “gSynchConsole” để đọc dữ liệu từ bàn phím và lưu trữ vào buffer.

3. Khởi tạo một biến check bằng true. Nếu số ký tự ‘.’ trong buffer nhiều hơn 1 hoặc các ký tự không phải là không phải là số thì biến biến check bằng false.

4.1 Nếu check vẫn bằng true thì chuyển từ buffer sang biến float “ f ” bằng hàm sscanf của c. Sau đó dùng kỹ thuật type punning để chuyển float “ f “ thành int “ f\_int ” nhờ con trỏ mà không làm mất dữ liệu trên bộ nhớ. Sau đó lưu “ f\_int ” vào register số 2 để syscall trả về.

4.2 Nếu check bằng false thì sẽ in ra màn hình console dữ liệu vừa nhập không phải là số thực và lưu 0 vào register số 2 để syscall trả về.

5. Cuối cùng tăng giá trị progam counter.

**12. Cài đặt SC\_PrintFloat**

- Các bước thực hiện:

1. Đọc số thực từ thanh ghi 4 của máy vào biến “ f\_int ”.

2. Sau khi đọc kí tự từ thanh ghi, ta lại dùng kỹ thuật type punning để chuyển “f\_int” từ kiểu int sang biến “ f ” kiểu float.

3. Sử dụng hàm sprintf của c để lưu của kí tự của số thực cần in vào mảng char tên “buffer”.

4. Sau đó dùng phương thức “Write” của đối tượng “gSynchConsole” để ghi kí tự đó ra màn hình. Kí tự được ghi bằng cách truyền địa chỉ của “c” và số lượng kí tự cần ghi, trong trường hợp này là 1.

5. Cuối cùng, tăng giá trị program counter.

**Define trong “syscall.h”**

* Trong file “syscall.h”, các syscall được define như sau: SC\_ReadChar và SC\_PrintChar là 10 và 11, SC\_ReadInt và SC\_PrintInt là 15 và 16, SC\_ReadString và SC\_PrintString là 17 và 18, SC\_ReadFloat và SC\_PrintFloat là 19 và 20.

**Khai báo biến toàn cục trong “.code/threads” 2 file “system.h” và “system.cc”**

* Trong file “system.h”, khai báo thư viện “synchcons.h” và một biến con trỏ gSynchConsole để dùng cho việc read và write ở console
* Trong file “system.cc”, cũng khai bào con trỏ gSynchConsole từ SynchConsole, cài đặt tạo mới và giải phóng bộ nhớ.

**Thay đổi trong .code/filesys/filesys.h**

* Trong lớp FileSystem,
  + khai báo “openfile” là một mảng con trỏ đến các đối tượng “OpenFile”, được sử dụng để theo dõi các file đang mở trong file system. Mỗi đối tượng “OpenFile” đại diện cho một file đang mở và cung cấp các phương thức để thực hiện thao tác đọc ghi, di chuyển trong file.
  + Khai báo biến “index” để theo dõi số lượng các file đang mở trong file system. Nó chỉ ra vị trí kế tiếp trong mảng “openfile” mà một file mới sẽ được thêm vào.
* Ở các hàm Open, tăng giá trị index mỗi khi kết thúc hàm để chỉ ra vị trí kế tiếp trong mảng “openfile”.
* Khai báo và cài đặt hàm open file với type chỉ định.

**Cài đặt chương trình sắp xếp mảng các số nguyên dùng thuật toán Merge Sort**

* Viết chương trình cài đặt trong file “mergesort.c” trong thư mục .code/test.
* Chương trình yêu cầu người dùng nhập số lượng phần tử của mảng.
* Nhập các phần tử của mảng tuần tự.
* Sau khi nhập mảng thì gọi hàm “mergeSort” để sắp xếp các phần tử trong mảng.
* Một file tên “mergesort.txt” được mở để ghi kết quả vào.
* Nếu không mở được thì thông báo lỗi và kết thúc.
* Các phần tử trong mảng được chuyển thành chuỗi kí tự bằng hàm “intToString” và sau đó ghi vào file “mergesort.txt”, cách nhau bởi dấu cách.
* Sau khi ghi xong, file “mergesort.txt” được đóng.
* Chương trình in ra màn hình một thông báo cho người dùng biết rằng kết quả đã được ghi vào file “mergesort.txt”.
* Cuối cùng, kết thúc chương trình bằng Halt().