CS333 - INTRODUCTION TO OPERATING SYSTEMS - 19APCS2
PROJECT 1

Sauce:
19125063 - Châu Hữu Phát
19125090 - Trương Quý Hải
19125119 - Trương Lăng Trường Sơn

Ideas:

1. Handle all exceptions which are listed in machine/machine.h:
- Case no exception: return the control to the operating system.
- Case syscall exceptions: will be handled by functions for user system calls.
- Case other exceptions: print error message and halt the system.

2. void ModifyReturnPoint():
- This code is adapted from `../machine/mipssim.cc`.
- Set previous program counter.
- Set program counter to next instruction with all instructions are 4 bits wide.
- Set next program counter for brach execution.

3. int ReadNum():
- The system call ReadNum will read all the characters from the console, which are in one line.
- If there are any errors in the reading process => return 0
- Else return the number
- In the process of reading characters from the console.
  + If there are some special characters found => return 0.
  + If there is any '-' sign in the middle of the sequence => error => return 0.
  + If the number is longer than any int in C/C++ => overflow => return 0.
- After reading and converting numberBuffer(char*) to number(int), compare numberBuffer and number again. If those two are different => some errors occurred in the process of converting => return 0.

4. void PrintNum(int number):
- Use kernel synchConsoleOut->putChar to print the characters sequence to the console.
- if number is 0 => print 0 to console, if number is int32_min => print -2147483648 to console.
- In the while loop, if the number is greater than 0, store the last unit number in numberBuffer for every loop and divide the number by 10.
- After constructing the numberBuffer => print to the console.

5. char ReadChar():
- Use kernel->synchConsoleIn->GetChar().

6. void PrintChar(char character):
- Use kernel->synchConsoleOut->PutChar(character).

7. int RandomNum():
- Use random() function.

8. void ReadString (char[] buffer, int length):
- Create a buffer with length + 1.
- Use for loop and ReadCharSys from function void ReadChar() to append all characters of the string.
- Append '\0' at the end of the string.

9. void PrintString(char* buffer, int length):
- Use a for loop to loop through the string and print each character of the string: kernel->synchConsoleOut->PutChar(buffer[i]).

10. Help program:
- Use the PrintString function to print the introduction of the group, ASCII, and sorting program.

11. ASCII program:
- Use PrintChar from char 32 to char 126.

12. Sorting program:
- First, get inputs from the user: array size, elements, sorting order (using do…while loop and PrintString, ReadNum functions).
- Second, check the input validation. If the inputs are wrong, the user will be asked to input again.
- Then, use bubble sort algorithm to sort the array and display result using PrintString, PrintNum, PrintChar functions.

How to run:

1. In code/build.linux run
   - make depend
   - make
2. In code/test we can run
   - make readnum: this test file includes readNum() and PrintNum()
   - make char_t: this test file includes ReadChar() and PrintChar()
   - make random_num: this test file includes RandomNum() function
   - make string_t: this test file includes ReadString() and PrintString()
   - make help: this program includes a description about our team and two programs: ascii and sort.
   - make ascii_prog: this program will print ascii characters
   - make bubble_sort: this program will receive n integers and sort those integers base on bubble sort.
3. In code/test run: ../build.linux/nachos -x readnum to run.