## 

## University of Dhaka

## Department of Computer Science and Engineering

Operating System Lab- 1

### 

**Submitted by:**

### Sayma Sarwar Ela

Roll: 056

Registration no: 2018-325-351

Sagar Chandra Karmakar Babu

Roll: 030

Registration no : 2018-225-325

**Submitted to:**

1. Dr. Mosaddek Hossain Kamal Tushar

Professor

Department of Computer Science and Engineering

University of Dhaka

2. Dr. Upama Kabir

Professor

Department of Computer Science and Engineering

University of Dhaka

Date: 19 August, 2022

# **Implementing kprintf and kscanf:**

In the assignment the task was to implement kprintf and kscanf functions. The kprintf function is able to print output and kscanf is able to take input. They are custom made input and output functions that work similar to printf and scanf. We used usart2 port of STM32F446re board.

In the arch/dev/usart.c there are some functions that we used for our task. The functions UART\_SendChar, \_USART\_READ, \_USART\_WRITE were called from our functions.

\_UART\_SendChar the data register is configured to load data and the status register is configured to wait for the transfer of data.

In the function \_USART\_WRITE the data send character function is called for all the characters to transfer.

In the \_USART\_READ the function \_UART\_GetChar is called. In the function the status register and the data register are set. The \_USART\_READ function calls GetChar for all the characters in the buffer. When a new line or carriage return is found it stops reading.

In the makefile some modifications were required to allow floating point data type. The following code was added:

*CC = arm-none-eabi-gcc*

*FPU = -mfloat-abi=softfp -mfpu=fpv4-sp-d16*

*CFLAGS = -c -mcpu=$(MACH) -mthumb $(FPU) -std=gnu11 -Wall -O0*

*LFLAGS = -nostdlib -nostartfiles -nodefaultlibs -fno-exceptions -mcpu=$(MACH) -mthumb $(FPU) -T ../kern/asst01/stm32\_ls.ld -Wl,-Map=$(OUTPUT\_FOLDER)/final.map -O0*

*MACH = cortex-m4*

We defined SBC struct. And floating point unit in the CPACR register was activated.

The functions support %c, %x, %d, %x and %f as the value of their first parameter. Where:

\* %c is used for character

\* %s is for string

\* %d for integer

\* %x hexadecimal

\* %f for floating point number

**void kprintf(uint8\_t \*format, uint8\_t\* outvar):**

The kprintf function has two parameters. The first parameter is the address of the format of the variable. And the second is the address of the value to be printed.

For %c we directly used UART sendChar as the output has only one character. For %s uart write is called.

For %d and %x some modification was made. The value was converted from number to string . In order to do that the numbers were divided by there base and and they were appended in a string.

For %f the significand part was converted to string. And the fractional part was converted to string by multiplying by 10.

**void kscanf(uint8\_t \*format,uint8\_t\* invar):**

First parameter is the address of the specification of the format. The second parameter is the destination address where the input will be saved.

For %c er directly called the read function with 1 as parameter.

For %s also the read function is called with 200 as parameter.

For %d and %x conversion is required. The string in input was converted to Int using StringtoInt function. For %x the string StringHextoDec function was used. This function gives an integer as output.

For %f the significand part was extracted by looping and and converted to float. Then the fraction part was calculated by multiplying all the digits by their local value.