

Compiler Design Lab-1

- Write a simple C program to find the Endianness (Little or Big) of your machine/architecture, compile your C code to assembly and comment your assembly code line by line, to express the flow of operations that takes place.

```
#include<stdio.h>

int main() {
    unsigned int var = 0x00000001;
    char* cPtr = (char*) &var;
    printf((*cPtr)?"Little %x\n":"Big %x\n", *cPtr);
    return 0;
}
```

- call the assembler 'as' to compile the assembly code to object code.

gcc -S endian.c

- Ask GCC to link your object code to other system wide objects and build the executable.

gcc -v -save-temps -o endian endian.c

- Execute and show the output.

./endian

- **Output:** Little

- With the understanding from the above activities, write a program in GAS assembly syntax to continuously display characters as entered via the keyboard until a special key is pressed. You can define your own special key.

```
.text
.global _start
_start:

loop:
    xorq    %rax, %rax
    movq    %rax, %rdi
    mov     $var, %rsi
    movq    $1, %rdx
    syscall

    cmpb    $27, (var)
    jz      exit

    movq    $1, %rax
    movq    %rax, %rdi
    mov     $var, %rsi
    movq    %rax, %rdx
    syscall

    jmp     loop
exit:
    movq    $0x3c, %rax
    xorq    %rdi, %rdi
    syscall

.data
var: .byte 0
```