ECE220: Computer Systems & Programming **Fall 2020 ZJUI**
Lab 3 **Tuesday 29 September**

Decoding a Secret Message

In today's lab, your group must write a subroutine to decode a secret message by traversing a sequence of data structures. The goal of this exercise is to help you formulate code that finds the end of a string, makes use of bit vectors, and makes use of data structures to solve a problem. MP2 requires all techniques.

Begin by updating your repository to obtain the **`lab3`** subdirectory. The directory contains a copy of this document and a starting point for your solution in the **`lab3.asm`** file.

The subroutine that you must write is called **`FIND_SECRET`**. An empty version with only a **`RET`** instruction appears in the file given to you, and some code to call the subroutine is also provided. Including comments and blank lines, completing the subroutine requires fewer than 50 lines of code.

The Task

The label **`SECRET`** (provided to you below your subroutine) marks the beginning of a sequence of entries. Each entry consists of a non-empty, NUL-terminated ASCII string followed by bit vector that occupies a single memory location (after the NUL in the string). What is a bit vector? Simply a set of 16 bits. The sequence ends with an empty string (a single NUL character by itself).

The table below illustrates how a few of the entries might appear in memory. The first entry starts at address x5000. The string is "As," and the bit vector (at address x5003) is x1234. The second entry starts in the next memory location, address x5004, with string "m" and, at address x5006, bit vector x9DCB. Finally, at address x5007, we find an entry with string "!*" and bit vector x0030 (at address x500A).

address	contents	Meaning
x5000	x0041	'A'
x5001	x0073	's'
x5002	x0000	NUL
x5003	x1234	bit vector for string "As"
x5004	x006D	'm'
x5005	x0000	NUL
x5006	x9DCB	bit vector for string "m"
x5007	x0021	'!'
x5008	x002A	'*'
x5009	x0000	NUL
x500A	x0030	bit vector for string "!*"

The **`FIND_SECRET`** subroutine accepts as input a bit number (ranging from 0 to 15) in R3. You should first translate this bit number into a bit vector with a single 1 bit. For example, when R3=3, your subroutine should calculate x0008, in which bit 3 is a 1, and all other bits are 0. Similarly, when R3=10, your subroutine should calculate x0400, in which bit 10 is a 1, and all other bits are 0.

Your subroutine must then go through the sequence of entries starting at **`SECRET`** and print the string for any entry in which the bit named by R3 is a 1 in the entry's bit vector. Use an **`AND`** instruction to check, and a **`PUTS`** trap to print the string when appropriate.

Once your subroutine works, exactly one key value (from 0 to 15, passed in R3) will produce a meaningful string in English. Can your team determine the value of this secret key? Will you be the first?