Topic V1A

Examples of String Functions

Reading: (Section 2.9)

Strings

V	e	r	o	n	i	c	a	0

X	i	n	m	e	n	g	0

M	a	l	c	o	l	m	0

H	a	o	t	i	a	n	0

S	a	n	a	e	0

T	a	y	l	o	r	0

0x1001 002C	r	0	-	-
0x1001 0028	a	y	l	o
0x1001 0024	a	e	0	T
0x1001 0020	0	S	a	n
0x1001 001C	t	i	a	n
0x1001 0018	0	H	a	o
0x1001 0014	c	o	l	m
0x1001 0010	0	M	a	l
0x1001 000C	m	e	n	g
0x1001 0008	0	X	i	n
0x1001 0004	n	i	c	a
0x1001 0000	V	e	r	o

Quiz

Write RISC-V assembly code for the function `strlen`:

Parameter:

`a0`: address of null-terminated string

Return value:

`a0`: number of characters in string

```
int strlen(char *s){
        char *temp = s;
        while (*s != 0)
                s++;
        return s – temp;
}
```

```
        t0 ← a0
nextC:  t1 ← M[a0]
        if (t1 == 0) goto end
        a0 ← a0 + 1
        goto nextC
end:    a0 ← a0 – t0
```

0x1001 002C	r	0	-	-
0x1001 0028	a	y	l	o
0x1001 0024	a	e	0	T
0x1001 0020	0	S	a	n
0x1001 001C	t	i	a	n
0x1001 0018	0	H	a	o
0x1001 0014	c	o	l	m
0x1001 0010	0	M	a	l
0x1001 000C	m	e	n	g
0x1001 0008	0	X	i	n
0x1001 0004	n	i	c	a
0x1001 0000	V	e	r	o

```
strlen:
        add     t0, zero, a0
nextC:
        lbu     t1, 0(a0)
        beq     t1, zero, end
        addi    a0, a0, 1
        jal     zero, nextC
end:
        sub     a0, a0, t0
        jalr    zero, ra, 0
```

```
        t0 ← a0
nextC:  t1 ← M[a0]
        if (t1 == 0) goto end
        a0 ← a0 + 1
        goto nextC
end:    a0 ← a0 – t0
```

a0	0x1001 0019
t0	0x1001 0019
t1	0x0000 0048

0x1001 002C	r	0	-	-
0x1001 0028	a	y	l	o
0x1001 0024	a	e	0	T
0x1001 0020	0	S	a	n
0x1001 001C	t	i	a	n
0x1001 0018	0	H	a	o
0x1001 0014	c	o	l	m
0x1001 0010	0	M	a	l
0x1001 000C	m	e	n	g
0x1001 0008	0	X	i	n
0x1001 0004	n	i	c	a
0x1001 0000	V	e	r	o

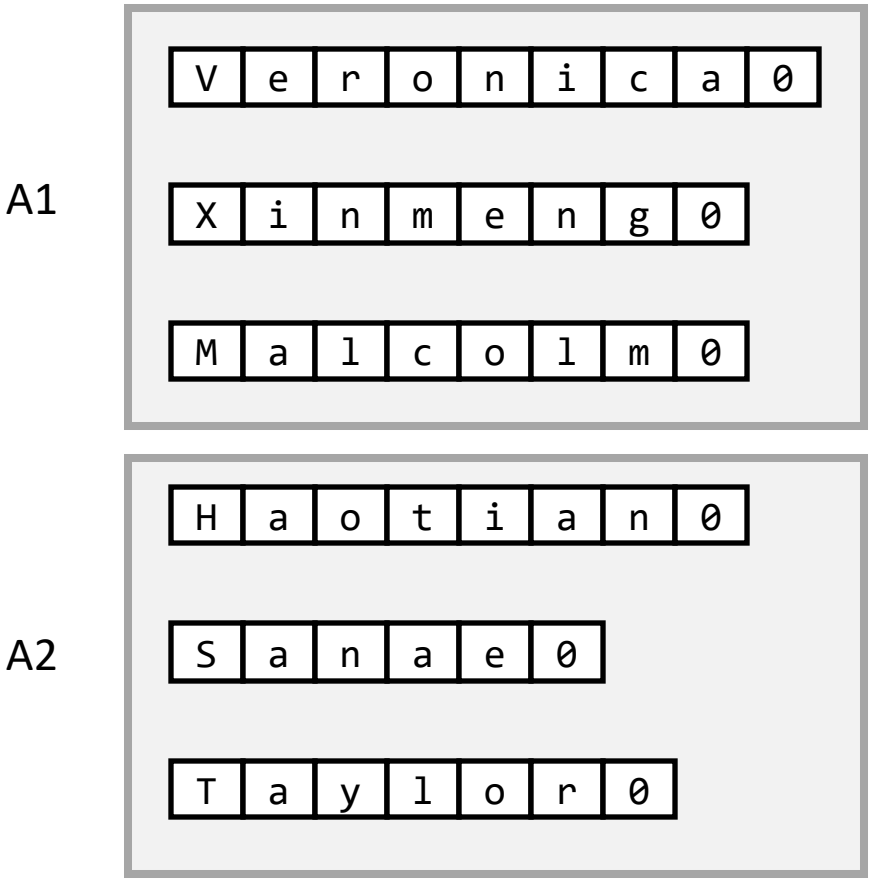
```
strlen:
        add     t0, zero, a0
nextC:
        lbu     t1, 0(a0)
        beq     t1, zero, end
        addi    a0, a0, 1
        jal     zero, nextC
end:
        sub     a0, a0, t0
        jalr    zero, ra, 0
```

```
        t0 ← a0
nextC:  t1 ← M[a0]
        if (t1 == 0) goto end
        a0 ← a0 + 1
        goto nextC
end:    a0 ← a0 – t0
```

a0	0x0000 0007
t0	0x1001 0019
t1	0x0000 0000

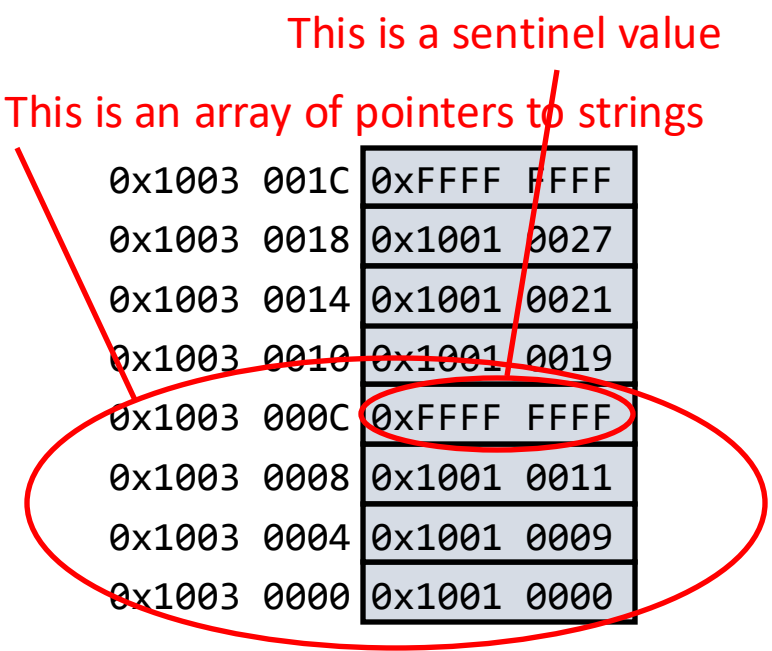
0x1001 002C	r	0	-	-
0x1001 0028	a	y	l	o
0x1001 0024	a	e	0	T
0x1001 0020	0	S	a	n
0x1001 001C	t	i	a	n
0x1001 0018	0	H	a	o
0x1001 0014	c	o	l	m
0x1001 0010	0	M	a	l
0x1001 000C	m	e	n	g
0x1001 0008	0	X	i	n
0x1001 0004	n	i	c	a
0x1001 0000	V	e	r	o

Creating arrays of student names



0x1001 002C	r	0	-	-
0x1001 0028	a	y	l	o
0x1001 0024	a	e	0	T
0x1001 0020	0	S	a	n
0x1001 001C	t	i	a	n
0x1001 0018	0	H	a	o
0x1001 0014	c	o	l	m
0x1001 0010	0	M	a	l
0x1001 000C	m	e	n	g
0x1001 0008	0	X	i	n
0x1001 0004	n	i	c	a
0x1001 0000	V	e	r	o

Arrays of Pointers to Strings



0x1001 002C	r	0	-	-
0x1001 0028	a	y	l	o
0x1001 0024	a	e	0	T
0x1001 0020	0	S	a	n
0x1001 001C	t	i	a	n
0x1001 0018	0	H	a	o
0x1001 0014	c	o	l	m
0x1001 0010	0	M	a	l
0x1001 000C	m	e	n	g
0x1001 0008	0	X	i	n
0x1001 0004	n	i	c	a
0x1001 0000	V	e	r	o

Quiz

Write RISC-V assembly code for the function `maxlen`:

Parameter:

`a0`: address of the first pointer to a name in a class

Return value:

`a0`: number of characters in longest name

```
int maxlen(char **p){
  int max = 0;
  while (*p != -1){
    t = strlen(*p);
    if (t > max)
              max = t;
    p++;
   }
  return max;
}
```

0x1003 001C	0xFFFF FFFF
0x1003 0018	0x1001 0027
0x1003 0014	0x1001 0021
0x1003 0010	0x1001 0019
0x1003 000C	0xFFFF FFFF
0x1003 0008	0x1001 0011
0x1003 0004	0x1001 0009
0x1003 0000	0x1001 0000

0x1001 0020	0	S	a	n
0x1001 001C	t	i	a	n
0x1001 0018	0	H	a	o
0x1001 0014	c	o	l	m
0x1001 0010	0	M	a	l
0x1001 000C	m	e	n	g
0x1001 0008	0	X	i	n
0x1001 0004	n	i	c	a
0x1001 0000	V	e	r	o

```
maxlen:
            add     s0, zero, a0        # s0 ← p
            add     s1, zero, zero      # max ← 0
            li      s2, -1
nextS:      lw      a0, 0(s0)           # a0 ← *p
            beq     a0, s2, end         # if (*p == -1)
            jal     ra, strlen          # t ← strlen(*p)
            addi    s0, s0, 4           # p++
            ble     a0, s1, nextS       # if (t ≤ max)
            add     s1, zero, a0        # max ← t
            jal     zero, nextS
end:        add     a0, zero, s1        # return max
            jalr    zero, ra, 0
```

```
int maxlen(char **p){
  int max = 0;
  while (*p != -1){
    t = strlen(*p);
    if (t > max)
            max = t;
    p++;
   }
  return max;
}
```

0x1003 001C	0xFFFF FFFF
0x1003 0018	0x1001 0027
0x1003 0014	0x1001 0021
0x1003 0010	0x1001 0019
0x1003 000C	0xFFFF FFFF
0x1003 0008	0x1001 0011
0x1003 0004	0x1001 0009
0x1003 0000	0x1001 0000

Code to save/restore s registers and ra missing

0x1001 0020	0	S	a	n
0x1001 001C	t	i	a	n
0x1001 0018	0	H	a	o
0x1001 0014	c	o	l	m
0x1001 0010	0	M	a	l
0x1001 000C	m	e	n	g
0x1001 0008	0	X	i	n
0x1001 0004	n	i	c	a
0x1001 0000	V	e	r	o

```
maxlen:
        add     s0, zero, a0        # s0 ← p
        add     s1, zero, zero      # max ← 0
        li      s2, -1
nextS:  lw      a0, 0(s0)           # a0 ← *p
        beq     a0, s2, end         # if (*p == -1)
        jal     ra, strlen          # t ← strlen(*p)
        addi    s0, s0, 4           # p++
        ble     a0, s1, nextS       # if (t ≤ max)
        add     s1, zero, a0        # max ← t
        jal     zero, nextS
end:    add     a0, zero, s1        # return max
        jalr    zero, ra, 0
```

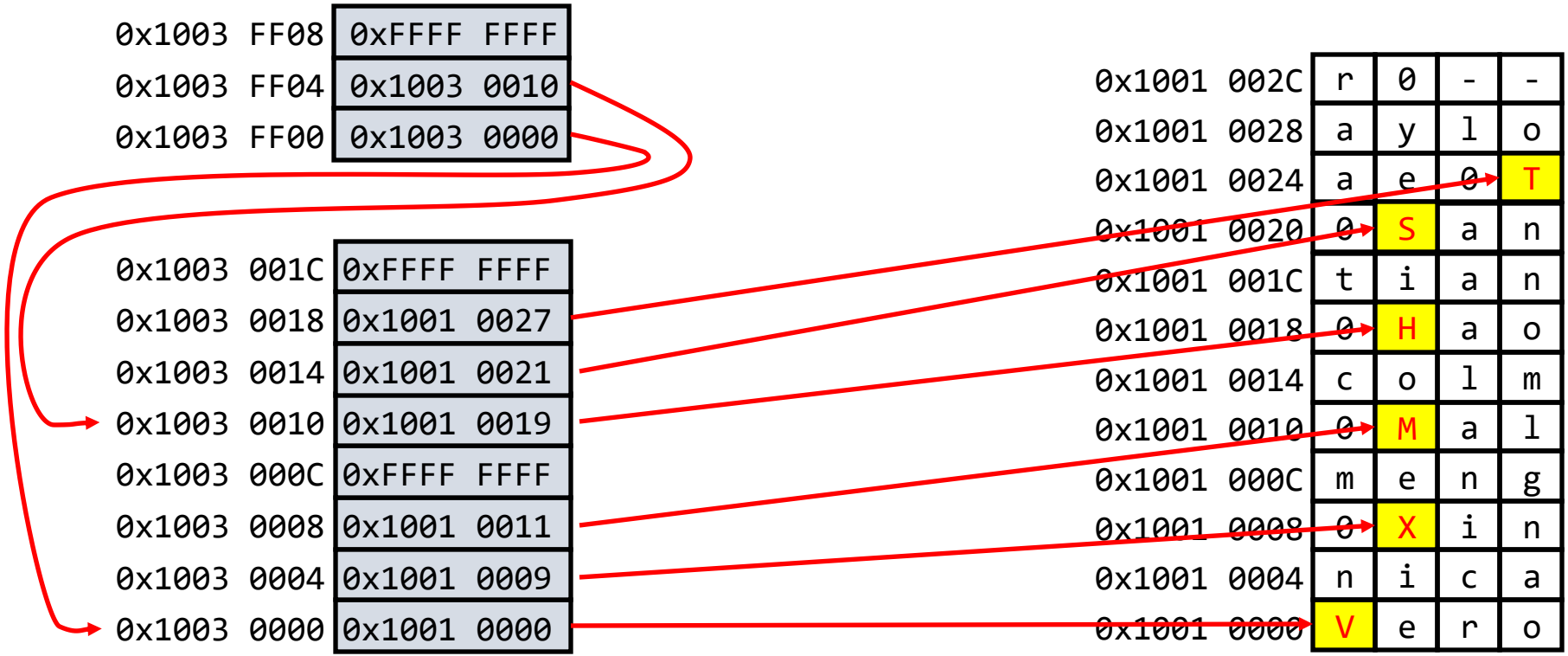
a0	0x0000 0000
s0	0x1003 0004
s1	0x0000 0000

0x1003 001C	0xFFFF FFFF
0x1003 0018	0x1001 0027
0x1003 0014	0x1001 0021
0x1003 0010	0x1001 0019
0x1003 000C	0xFFFF FFFF
0x1003 0008	0x1001 0011
0x1003 0004	0x1001 0009
0x1003 0000	0x1001 0000

0x1001 002C	r	0	-	-
0x1001 0028	a	y	l	o
0x1001 0024	a	e	0	T
0x1001 0020	0	S	a	n
0x1001 001C	t	i	a	n
0x1001 0018	0	H	a	o
0x1001 0014	c	o	l	m
0x1001 0010	0	M	a	l
0x1001 000C	m	e	n	g
0x1001 0008	0	X	i	n
0x1001 0004	n	i	c	a
0x1001 0000	V	e	r	o

Creating an Array of Sessions

An Array of Pointers to Pointers to Strings



sumax:

Parameter:

a0: address of a list of classes

Return value:

a0: sum of longest names in each session

0x1003 FF08	0xFFFF FFFF
0x1003 FF04	0x1003 0010
0x1003 FF00	0x1003 0000

0x1003 001C	0xFFFF FFFF
0x1003 0018	0x1001 0027
0x1003 0014	0x1001 0021
0x1003 0010	0x1001 0019
0x1003 000C	0xFFFF FFFF
0x1003 0008	0x1001 0011
0x1003 0004	0x1001 0009
0x1003 0000	0x1001 0000

```
int sumax(char ***s){
        int sum = 0;
        while (*s != -1){
                t = maxlen(*s);
                sum = sum + t;
                s++;
        }
        return sum;
}
```

0x1001 0024	a	e	0	T
0x1001 0020	0	S	a	n
0x1001 001C	t	i	a	n
0x1001 0018	0	H	a	o
0x1001 0014	c	o	l	m
0x1001 0010	0	M	a	l
0x1001 000C	m	e	n	g
0x1001 0008	0	X	i	n
0x1001 0004	n	i	c	a
0x1001 0000	V	e	r	o

sumax:

Parameter:

a0: address of a list of classes

Return value:

a0: sum of longest names in each session

0x1003 FF08	0xFFFF FFFF
0x1003 FF04	0x1003 0010
0x1003 FF00	0x1003 0000

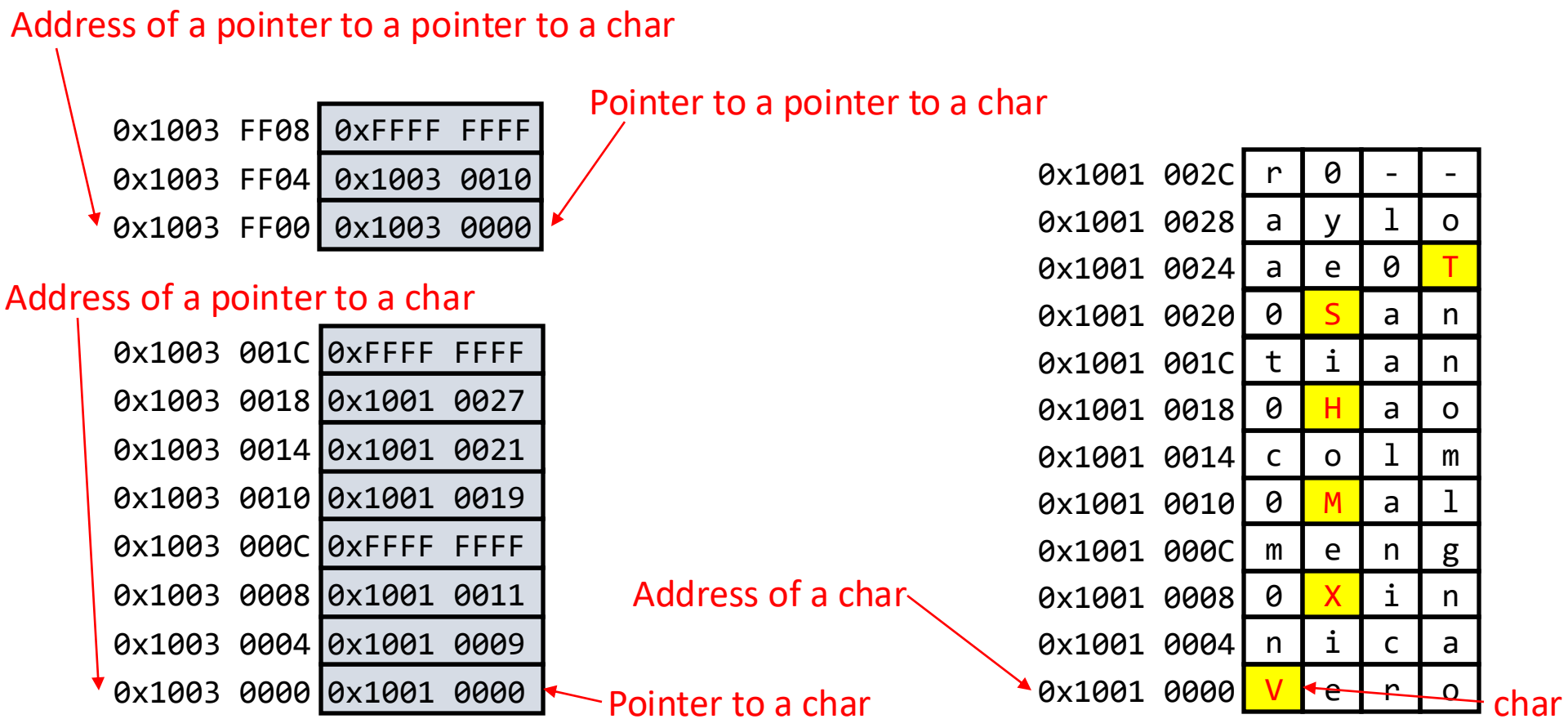
0x1003 001C	0xFFFF FFFF
0x1003 0018	0x1001 0027
0x1003 0014	0x1001 0021
0x1003 0010	0x1001 0019
0x1003 000C	0xFFFF FFFF
0x1003 0008	0x1001 0011
0x1003 0004	0x1001 0009
0x1003 0000	0x1001 0000

```
int sumax(char ***s){
        int sum = 0;
        while (*s != -1){
                t = maxlen(*s);
                sum = sum + t;
                s++;
        }
        return sum;
}
```

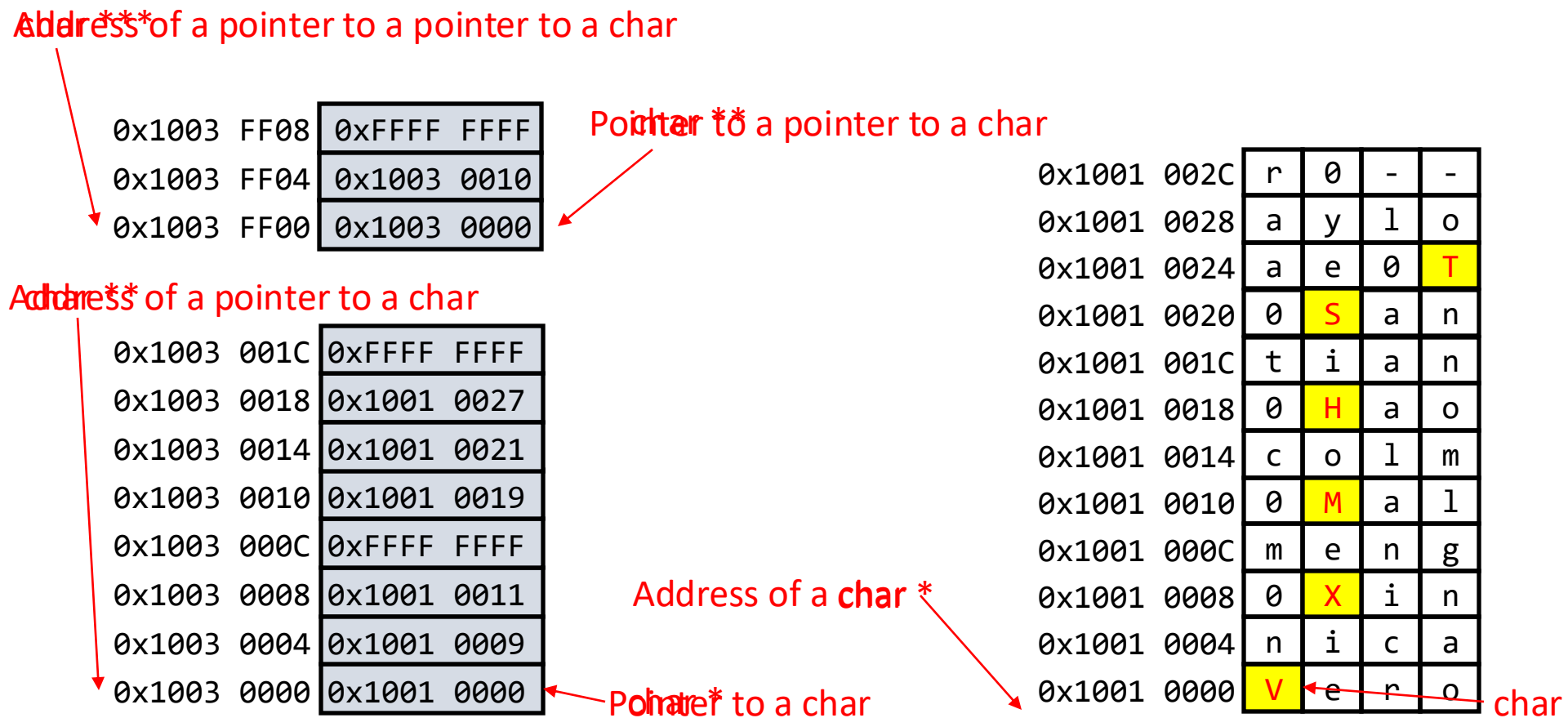
```
sumax:  add          s0, zero, a0          # s0 ← s
        add          s1, zero, zero        # sum ← 0
        addi         s2, zero, -1          # s2 ← -1
        lw           a0, 0(s0)             # a0 ← *s
        beq          a0, s2, end
next:   jal          ra, maxlen            # t ← maxlen(*s)
        add          s1, s1, a0            # sum ← sum + t
        addi         s0, s0, 4             # s++
        lw           a0, 0(s0)             # a0 ← *s
        bne          a0, s2, next
end:    add          a0, zero, s1
        jalr zero, ra, 0                   # return sum
```

Code to save/restore **s** registers and **ra** missing

Types

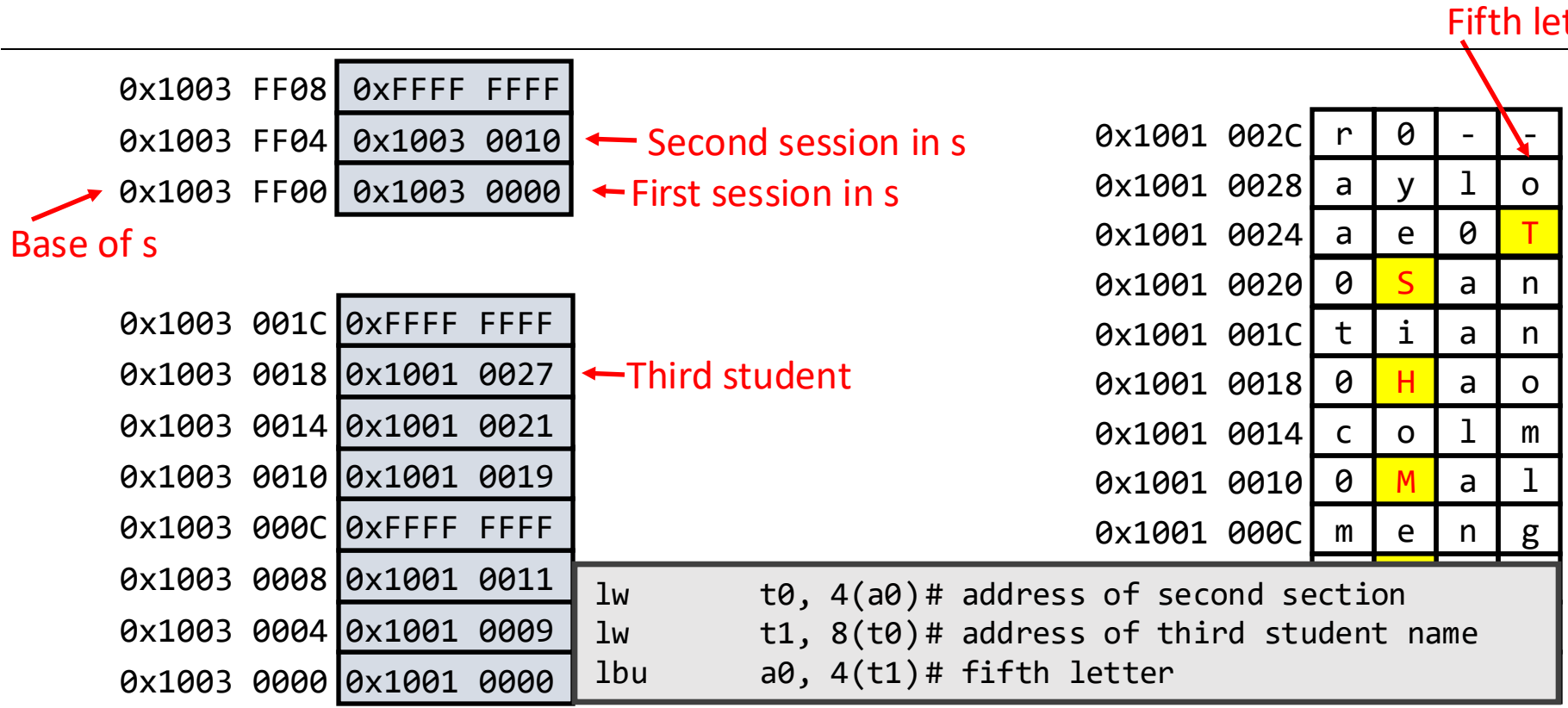

Address of a pointer to a pointer to a char
Pointer to a pointer to a char
0x1003 FF08 0xFFFF FFFF
0x1003 FF04 0x1003 0010
0x1003 FF00 0x1003 0000
Address of a pointer to a char
0x1003 001C 0xFFFF FFFF
0x1003 0018 0x1001 0027
0x1003 0014 0x1001 0021
0x1003 0010 0x1001 0019
0x1003 000C 0xFFFF FFFF
0x1003 0008 0x1001 0011
0x1003 0004 0x1001 0009
0x1003 0000 0x1001 0000
Pointer to a char
Address of a char
0x1001 002C r 0 - -
0x1001 0028 a y l o
0x1001 0024 a e 0 T
0x1001 0020 0 S a n
0x1001 001C t i a n
0x1001 0018 0 H a o
0x1001 0014 c o l m
0x1001 0010 0 M a l
0x1001 000C m e n g
0x1001 0008 0 X i n
0x1001 0004 n i c a
0x1001 0000 V e r o
char

Types


Address of a pointer to a pointer to a char
0x1003 FF08 0xFFFF FFFF
0x1003 FF04 0x1003 0010
0x1003 FF00 0x1003 0000
Pointer to a pointer to a char
Address of a pointer to a char
0x1003 001C 0xFFFF FFFF
0x1003 0018 0x1001 0027
0x1003 0014 0x1001 0021
0x1003 0010 0x1001 0019
0x1003 000C 0xFFFF FFFF
0x1003 0008 0x1001 0011
0x1003 0004 0x1001 0009
0x1003 0000 0x1001 0000
Pointer to a char
Address of a char
0x1001 002C r 0 - -
0x1001 0028 a y l o
0x1001 0024 a e 0 T
0x1001 0020 0 S a n
0x1001 001C t i a n
0x1001 0018 0 H a o
0x1001 0014 c o l m
0x1001 0010 0 M a l
0x1001 000C m e n g
0x1001 0008 0 X i n
0x1001 0004 n i c a
0x1001 0000 V e r o
char

Quiz

Let s be an array of pointers to lists of students in sessions of a class



What have we learned?

An Array of Pointers to Pointers to Strings