Tutorial 1

LEI WANG
lei.wang2@ucalgary.ca

TuTh 09:00-9:50AM MS156
MoWe 17:00-17:50PM MS176

Office hour: ICT 720, Monday 2:00-4:00PM

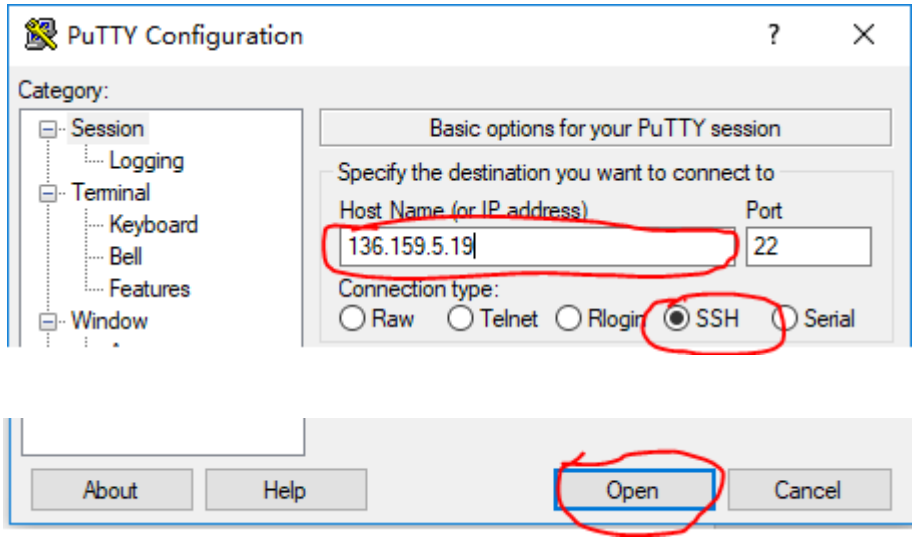
UNIVERSITY OF CALGARY

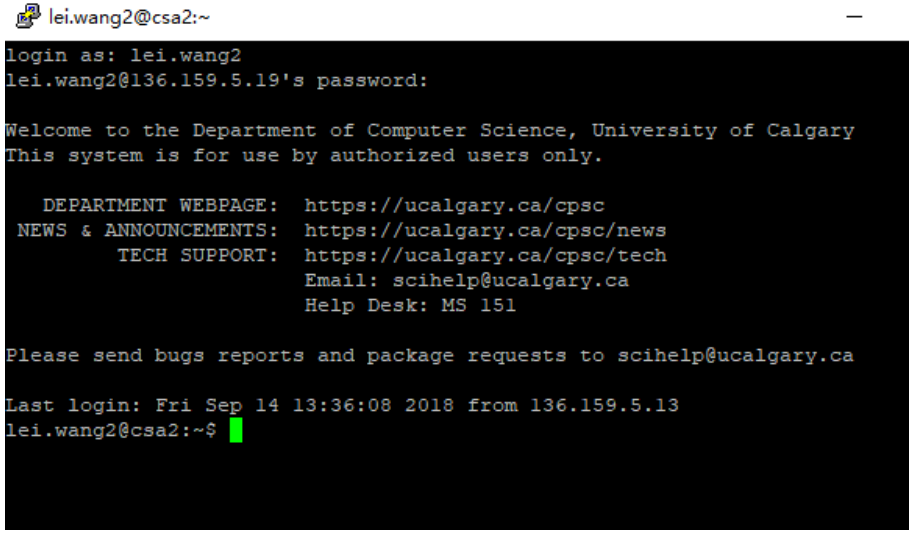
Remote editing

- **Connect to remote server**
  - Use *ssh* (linux/mac, **teminal**) or an *ssh* client program(windows, **putty**)
  - Connect to ucalgary cspc arm server: arm.cpsc.ucalgary.caip address: 136.159.5.19
    more information: http://www.ucalgary.ca/cpsc/tech/services/computing_resources
  - Use your CPSC account to login into the server

Teminal: ssh username@arm.cpsc.ucalgary.ca

Putty:





UNIVERSITY OF CALGARY

Editor

- **vim, vi**
  - Create a c file: *vim <filename.c>* (or any file)
  - **Command mode**
    - :wq write file to disk and quit the editor
    - :q! quit(no warning)
    - :q quit(a warning is printed if a modified file has not been saved)

  - **Intert mode**
    - i insert text before the current cursor position
    - Esc insert mode to command mode

UNIVERSITY OF CALGARY

Hello world

- **Basic C programming**
  - Use gcc compiler

    use command : ***gcc <filename> –o <compiled filename>***

    run program using command: ***./<compiled filename>***
  - Get familiar with printf() and scanf()

Eg. First C program “hello world”

```
#include <stdio.h>
int main()
{
    printf(“hello, world\n”);
    return 0;
}
```

```
lei.wang2@csa2:~/tutorial1$ gcc hello.c -o hello.out
lei.wang2@csa2:~/tutorial1$ ./hello.out
hello, world
lei.wang2@csa2:~/tutorial1$
```


UNIVERSITY OF CALGARY

scanf()

- scanf()

  Eg. Enter your name

```
#include <stdio.h>
int main(){
    char str1[20];
    printf("Enter name: ")
    scanf("%s", str1);
    printf("Entered Name: %s\n", str1);
    return 0;
}
```

```
lei.wang2@csa2:~/tutorial1$ gcc enterName.c -o enterName.out
lei.wang2@csa2:~/tutorial1$ ./enterName.out
Enter name: leiwang
Entered Name: leiwang
lei.wang2@csa2:~/tutorial1$
```

Exercise 1

- Experiment to find out what happens when printf's argument string contains \c, where c is some character not listed above.
- Write a program to print the corresponding Celsius to Fahrenheit table.
- Write a program that calculates the volume of a sphere.
- Write a program to calculate the sum of three numbers with getting input in one line separated by a comma.

References: The C programming Language by Brian W.Kernighan and Dennis Ritchie

C-style string

- **What is C-style string**

C-style string is simply an array of characters that uses a null terminator.

Eg.

char myString[] = "string"; ⟶ *"string\0"*

Although "string" only has 6 letters, C automatically adds a null terminator to the end of the string for us

Evidence.

use *sizeof()* function to get the length of a string

Eg.

int length = sizeof(myString);

getchar() and putchar()

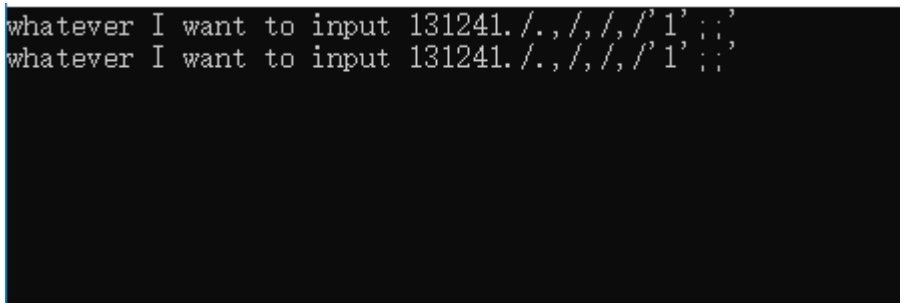
- getchar() read the next available character from the screen
- putchar() puts the passed character on the scree
- Eg.Enter your name

```
#include<stdio.h>
int main(){
    int c;
    printf("Enter name : ");
    c = getchar();
    printf("\n Entered name: ");
    putchar(c);
}
```

getchar() and putchar()

- Let's solve the problem of scanf() and printf()

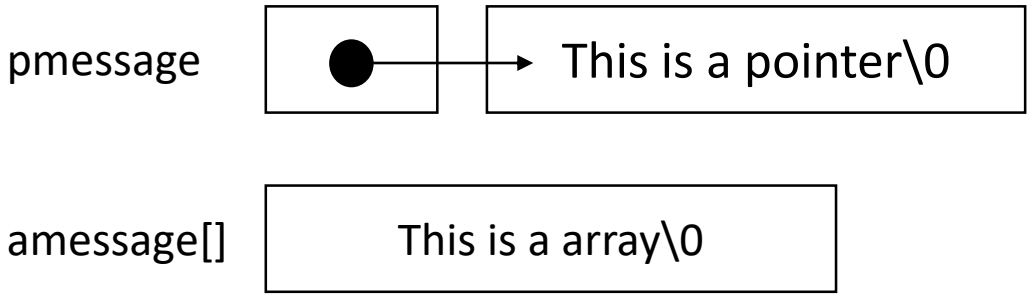
```
#include <stdio.h>
#include <string.h>
int main()
{
        int c;
        c = getchar();
         while(c != EOF){
                        putchar(c);
                        c = getchar();
    }
}
```



pointers, addresses and arrays

- Difference between pointer and array
  - *char amessage[] = "this is a array"; /* an array */*
  - *char *pmessage = "this is a pointer"; /* a pointer */*

- **pointer* is to dereference the pointer to get the value it points
- *&variable* is to get the address of the variable
- *pointer++* moves the pointer to the next character



- **pmessage** stores the address of the first character of the sequence which is char 'T'. so pmessage is a pointer, and *pmessage is the value it points which is called pointee.
- **amessage[]** stores the whole sequence, **amessage** stores the address of the first character of the sequence, which is amessage[0]. **So amessage is also a pointer!**

pointers, addresses and arrays

- How do we prove it?
- What's the difference between pmessage and amessage?

pointers, addresses and arrays

- Eg. Print the address of an array

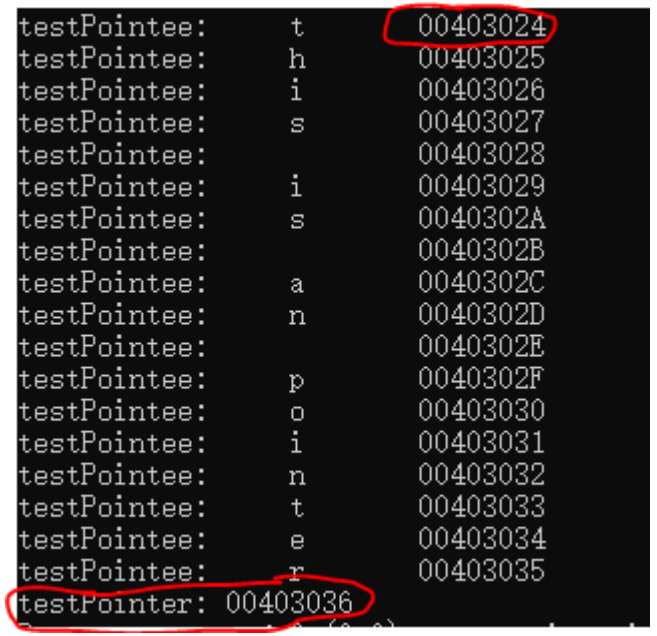
```
testArray[0]    0060FEF7    t
testArray[1]    0060FEF8    h
testArray[2]    0060FEF9    i
testArray[3]    0060FEFA    s
testArray[4]    0060FEFB
testArray[5]    0060FEFC    i
testArray[6]    0060FEFD    s
testArray[7]    0060FEFE
testArray[8]    0060FEFF    a
testArray[9]    0060FF00    n
testArray[10]   0060FF01
testArray[11]   0060FF02    a
testArray[12]   0060FF03    r
testArray[13]   0060FF04    r
testArray[14]   0060FF05    a
testArray[15]   0060FF06    y
testArray:      0060FEF7
```

```
#include <stdio.h>
#include <string.h>
int main()
{
    char testArray[] = "this is an array";
    for(int i=0; testArray[i] != '\0'; i++){
        printf("testArray[%d] \t %p \t %c \n", i, &testArray[i], testArray[i]);
    }
    printf("testArray: \t %p", testArray);
}
```

pointers, addresses and arrays

- Eg. Print the address of a pointer

```
#include <stdio.h>
#include <string.h>
int main()
{
    char testArray[] = "this is an array";
    char * testPointer = "this is an pointer";
    while(*testPointer != '\0')
    {
        printf("testPointee: \t %c \t %p \n", *testPointer, testPointer);
        testPointer++
    }
    printf("testPointer: %p", testPointer);
}
```



strcpy

- Definition
  - array version

```
void strcpy(char *s, char *t)
{
        int i;
        i = 0;
        while((s[i] = t[i]) != '\0')
                i++;
}/* copy t to s */
```

  - pointer version

```
void strcpy(char *s, char *t)
{
        while((*s = *t) != '\0'){
                s++;
                t++;
        }
}/* copy t to s */
```

strcpy

- Example

```
#include<stdio.h>
#include<string.h>
int main(){
        char src[40];
        char dest[100];
        memset(dest, '\0', sizeof(dest));
/* memset() reset all the content in dest[] to '\0' */
        strcpy(src, "this is an example");
        strcpy(dest, src);
        printf("copied string: %s\n", dest);
        return(0)
}
```

```
lei.wang2@csa2:~/tutorial1$ vim strcpy.c
lei.wang2@csa2:~/tutorial1$ gcc strcpy.c -o strcpy.out
lei.wang2@csa2:~/tutorial1$ ./strcpy.out
copied string: this is an example
lei.wang2@csa2:~/tutorial1$
```

Exercise 2

- Write the function strend(s, t), which returns 1 if the string t occurs at the end of the strings, and zero otherwise.
- Learn about **Array of pointers** and **pointer to an array.**
- Compare two string without using string library functions.
- Write a program to sort a string array in ascending order.

Slides and other materials: D2L->Course Content->Tutorial L01, L06