# CS 214 Recitation(Sec. 6)

Zuohui Fu
Ph.D. Department of Computer Science
Office hour: Mon 2pm-3pm
Email: zf87 AT cs dot rutgers dot edu
09/21/2017

# Topics

- Basic LINUX & shell commands
- C data types (arrays, unions, enums)

# Basic LINUX & shell commands

- In LINUX, the shell is called BASH
- Need to notice about BASH:
  - Case Sensitivity (commands & filenames) PWD ≠ pwd
  - Directory separator is “/” - forward-slash such as /usr/src/linux
  - Filename not need to be “filename.txt” style filename.xxx.xxx.txt is OK

# Command Sheets

- For Beginners:
  https://diyhacking.com/linux-commands-for-beginners/

- Linux Bash Shell Cheat Sheet - Learn Code the Hard Way:
  https://learncodethehardway.org/unix/bash_cheat_sheet.pdf

# C data types

## Array

- char c[10] = {‘0’,‘1’,‘2’,‘3’,‘4’}; what are the values of the other 5 elements?
- int a[10] = {0}; int a[] = {1,2,3,4,5}; how many elements are in array a?
- char s[5] = {‘h’,‘i’}; char *str = s; Is a pointer to the address of first element of array s?
- use malloc to allocate a consecutive block of memory of the specified number of bytes:
- char *str = (char*)malloc(100*sizeof(char));
- use free to release the specified block of memory back to the system free(str);

# Unions

- A union is a special data type available in C that allows to store different data types in the same memory location.
- union Data {

int i;

float f;

char str[10];

}data;

# Example-what’s the output

- #include<stdio.h>

```
union  {
int i;
char  x[2];
}a;

int main(void)
{
a.x[0]  =  10;
a.x[1]  =  1;
printf("%d\n",a.i);
return  0;
}
```

# Example Answer-266

The int in the Union takes 4 bytes, char x[2] takes 2 bytes. So the Union takes 4 bytes.

The four bytes is byte3, byte2, byte1, byte0. Then a.x[0] = 10 denotes byte0(0xa), a.x[0] = 1 denotes byte1(0x1)

So in hex, it would be 0x0000010a, which is 266.

# enums- enumerated type

- enum Weekday {sun,mon,tue,wed,thu,fri,sat};

  enum Weekday workday, weekend;  //sun=0, mon=1, …, sat=6

  printf("%d",sun); // output is 0

- we can manually specify the value of enum elements:

  enum Weekday {sun=7, mon=1, tue, wed, thu, fri, sat};

- since each element has a integer value, we can use it to compare with other values eg. if (tue==mon) or if(fri>tue)

# Solution to the HW

```
#include<stdio.h>
int main()
{
for (int i = 1; i <10; i = i+1)
        {       for(int j = 0; j <1; j = j + 1)
                {
                        printf('*');
                }
        printf("\n");
        }
return 0;
}
```

# HW2 Assignment

0. What's wrong with this

#define line? #define N 10;

1. Suppose you defined the macro

    #define SIX 2*3

    Then, suppose you used it in another expression:

    int x = 12 / SIX;

What value would x be set to?

Write your own version of atoi

Take a char, inspect its int value and return its corresponding int value

e.g.

```
int test = my_atoi('5');
if( test == 5 )
{
        return 0;
}
else
{
        return -1;
}
```

Next, take a string of any length, scan its chars until you hit the '\0' and return the entire string's int value

e.g.

```
int test = my_atoi("512");
if( test == 512 )
{
        return 0;
}
else
{
        return -1;
}
```