

# Computer Science & Engineering 150A

## Problem Solving Using Computers – Laboratory

### Lecture 03 - Unix Basics,Pico,Gcc

Shuai Xie

(Adapted from Derrick Stolee, Lin Liu & Shuai Xie )

Spring 2010

- `pwd` – print current working directory
- `mkdir <foldername>` – create a new directory
- `cd <foldername>` – enter a specific directory
- `cd ..` – go to the parent directory
- `cd` – go to the home directory
- `touch <filename>` – create an empty file
- `ls` – show all the files
- `ls -l` – show detail information
- `ls <foldername>` – show files in a specific directory

- `pico <filename>` – "Notepad" in Unix
  - `Ctrl+O` – Save file
  - `Ctrl+X & y` – Save file and exit
  - `Ctrl+X & n` – Exit without saving
  - `Ctrl+C` – Show cursor position

- Copy, Move and Rename
  - `cp <oldfile> <newfile>` – Copy a file to a new file
  - `mv <filename> <directory>` – move a file to a directory
  - `mv <oldname> <newname>` – Rename a file
- Delete Files
  - `rm <filename>` – Be careful!

- Get Info
  - `man <commandname>` – Display the manual for command.
  - `less <filename>` – Display the contents of a file.
  - Press "q" to exit.

- gcc — c compiler in unix system

- Compile:

```
gcc filename.c
```

- Run:

```
a.out
```

# Printf: football game

CSCE150A

Unix Basics

Gcc

Printf

Input

Data Type

```
#include <stdio.h>
```

```
/* this is a multi-line  
comment! */
```

```
// this is a single line comment.
```

```
int main(void)  
{  
    printf("The football game is so cool!\n");  
    return 0;  
}
```

# Printf: gcc -o

CSCE150A

Unix Basics

Gcc

Printf

Input

Data Type

- Compile:

```
gcc -o program file.c
```

- Run:

```
./program
```

- Example:

```
gcc -o sayhello hello.c → ./sayhello
```

- Comparison:

```
gcc hello.c → a.out
```



- `scanf("format string", &variables...);`
  - Use the same as `printf`, except for `&` before variables.

```
scanf("%d %lf", &myInt, &myDouble);
```

- Be sure to define variables ahead of time!
- Remember to keep the same format string when input.

Placeholder	Variable Type
<code>%d</code>	<code>int</code>
<code>%lf</code>	<code>double</code>
<code>%c</code>	<code>char</code>

# Input: Example Program

CSCE150A

Unix Basics

Gcc

Printf

Input

Data Type

```
#include <stdio.h>
int main(void)
{
    int myID;
    printf("Please input your student ID: ");
    scanf("%d", &myID);
    printf("Your ID is %d, welcome to CSCE150A.\n", myID);
    return 0;
}
```

- `int variablename = (integral)number;`
- Examples:
  - `int PricelPhone = 199;`
  - `int NumCar = 3000;`
- Takes 4 bytes.
- ANSI C range: -32767 to 32767

# Double: real numbers

CSCE150A

Unix Basics

Gcc

Printf

Input

Data Type

- `double variablename = number(.decimals);`
- Examples:
  - `double PriceCasoline = 2.39;`
  - `double MinTime = 0.12321237;`
- Takes 8 bytes.

- `char variablename = 'c'`(a letter or symbol);
- Examples:
  - `char grade = 'A';`
  - `char SymbolMoney = '$';`
- Takes 1 byte.
- ASCII code

- `printf("format string", variable list );`
- Outputs "string" with a few variable replacements.

Placeholder	Variable Type
%d	int
%f	double
%c	char
:	:

- `printf("Michael Jackson leaves us in the year of %d.\n",2009);`
- `printf("My puppy weighs %f pounds.\n", 23.75 );`
- `printf("Hope you get %c in this semester.\n", 'A' );`

# Printf: Another example

CSCE150A

Unix Basics

Gcc

Printf

Input

Data Type

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
    int NumCourse=5;
```

```
    double Average=95.50;
```

```
    char Grade='A';
```

```
    printf("This semester I select %d courses.
```

```
    My average score is %.2f. I get all %c.\n",
```

```
    NumCourse, Average, Grade);
```

```
    return 0;
```

```
}
```

Output:

This semester I select 5 courses. My average score is 95.50. I get all A.