Lab2:
C++ under UNIX

EC327

“Introduction to Software Engineering”

Fall 2015

LAB1: SUMMARY

- What do the following commands stand for, do, and what arguments do they take?

```
ls
cd
mv
man
:q
zip
ls -l
```

AGENDA

- **The GNU C++ compiler (`g++`)**
- **UNIX**
  - **Environment Variables**
  - **Piping**
  - **New commands:**
    - **`cat`, `less`, `tail`, `more`**
- **User Input**

The GNU g++ Compiler

hello.cpp:

```
#include <iostream>

using namespace std;

int main()
{
     cout << "Hello World!" << endl;
     return 0;
}
```

The GNU g++ Compiler

Compile:

```
g++ hello.cpp
```

Execute:

```
./a.out
```

Environment Variables

```
echo $PATH
    /opt/local/bin:/opt/local/sbin:/usr/bin:/bin
```

Directories of executables

The GNU g++ Compiler

Phases:

-Preprocessing

- #include
- #define

```
g++ -E hello.cpp

g++ -E hello.cpp | less

g++ -E hello.cpp | tail

g++ -E hello.cpp | head

g++ -E hello.cpp > hello.E
```

“Piping”

-Lexing and Parsing

- Syntax Checking

-Compilation from C to Assembly to Machine Code

-Linking multiple Object Files

The GNU g++ Compiler

Phases:

-Preprocessing
- #include
- #define

-**Lexing and Parsing**
- Syntax Checking

-**Compilation from C to Assembly to Machine Code**.

-**Linking multiple Object Files**

```
g++ -S hello.cpp
g++ -c hello.cpp
cat hello.s
cat hello.o
```

Assembly Code

Object File

The GNU g++ Compiler

Phases:

-Preprocessing

- #include
- #define

-Lexing and Parsing

- Syntax Checking

-Compilation from C to Assembly to Machine Code.

-Linking multiple Object Files

```
g++ -E hello.cpp
g++ -S hello.cpp
g++ -c hello.cpp
```

Preprocessing
Assembly Code
Object File

```
g++ hello.cpp -o hello
./hello
```

Naming the Executable

User Input (1/3)

user_input.cpp

```
#include <iostream>
using namespace std;
int main(){
    //initializing number variable
    int number;
    //reading user input
    cin>>number;
    //output
    cout<<”Your input is ”<<number<<”!”<<endl;
    return 0;
}
```

User Input (3/3)

Compile:

```
g++ user_input.cpp
    -o lab2
```

Execute:

```
./lab2
```

Lab2: Takeaways

Re-compile when changing the source code

Compile all .cpp Files

Name the executable

```
g++ hello.cpp -o hello
```

Only clean code is good code!

```
#include <iostream>
using namespace std;
int main()
{
cout<<"Bad Hello World!";
return 0;
}
```

BAD! ☹

```
#include <iostream>

using namespace std;

int main()
{
        cout << "Hello World!" << endl;

        return 0;
}
```

GOOD! ☺

QUESTIONS?