

CIS 2107

Assignment 1

Written Part

1 Tools

1. We've said that when we run gcc on a .c file, there are several stages: preprocessing, compiling, assembling, and linking.

- (a) What happens during the preprocessor stage? What is input? What is output?

It interprets macros/comments (things before or in-between an #)

Input: Header and C files. Output: Preprocessed Header and C files.

- (b) What happens during the compilation stage? What is input? What is output?

It translates the preprocessor's output and turns it into Assembly.

Input: Preprocessed Header and C files. Output: Assembly instructions.

- (c) What happens during the assembly stage? What is input? What is output?

It translates Assembly instructions to processor instruction.

Input: Assembly instructions. Output: Processor instructions.

- (d) What happens during the linking stage? What is input? What is output?

It rearranges the processor instructions to create an executable file.

Input: Processor instructions. Output: Executable file.

2. What's the command to "go up" a level in the directory tree?

2. `cd ..`

3. What's the command to "go to" your home directory?

3. `cd ~`

4. What's the command to "go to" the root of the directory tree?
4. cd /
5. What's the command to delete a file?
5. rm
6. What's the command to delete a directory?
6. rm -d
7. What's the command to create a directory?
7. mkdir
8. What's the command to make a copy of a file?
cp filename filenamecopy
9. What's the command to rename a file?
mv filename newfilename
10. Suppose that there's a unix shell command called **petronam**. What's the command that I'd type in the shell in order to see the help page for the proper usage of **petronam**?
man petronam
11. What's the unix shell command to view the contents of a file in hexadecimal?
xxd filename
12. What's the unix shell command to display a file one screen at a time?
less filename
13. How would I run the command **keurig** taking its input from a file called **beans**?
beans < keurig
14. How can I take the output of a **bake** command and use it as input for a **slice** command?
bake | slice
15. What's the command to print a long directory listing (names, permissions, owner, length, *etc*)?
ls -l
16. What's the command to alter the permissions of a file to allow the owner to read, write, or execute the file?
chmod

2 Arguments to Functions

17. What's printed by each of the following?

```
(a) public class WhatsPrinted01 {  
2     public static void func(int x) {  
3         x++;  
4     }  
5  
6     public static void main(String args[]) {  
7         int x=5;  
8         func(x);  
9         System.out.println(x);  
10    }  
11 }  
12
```

(a) 5

```
(b) public class WhatsPrinted02 {  
2     public static void func(int x, int y, int z) {  
3         System.out.println("x=" + x +  
4                             ", y=" + y +  
5                             ", z=" + z);  
6     }  
7  
8     public static void main(String args[]) {  
9         int x=5, y=10, z=15;  
10        func(y, z, x);  
11    }  
12 }  
13
```

(b) x=10, y=15, z=5

```
(c) public class WhatsPrinted03 {  
2     public static void func(int x, int y, int z) {  
3         x++;  
4         y+=x;  
5         z+=y;  
6     }  
7  
8     public static void main(String args[]) {  
9         int x=5, y=10, z=15;  
10        func(y, z, x);  
11        System.out.println("x=" + x +  
12                             ", y=" + y +
```

```

13         ", z=" + z);
14     }
15 }

```

(c) x=5, y=10, z=15

```

(d) public class WhatsPrinted04 {
2     public static void func(int A[]) {
3         A[0]++;
4     }
5
6     public static void main(String args[]) {
7         int A[] = {10,20,30};
8         func(A);
9         System.out.println(A[0]);
10    }
11 }

```

(d) 11

```

(e) public class WhatsPrinted05 {
2     public static void func(int A[]) {
3         int B[] = new int[A.length];
4         for (int i=0; i<A.length; i++)
5             B[i]=A[i]+1;
6         A=B;
7     }
8
9     public static void main(String args[]) {
10        int A[] = {10,20,30};
11        func(A);
12        System.out.println(A[0]);
13    }
14 }

```

(e) 10

```

(f) class Derp {
2     public int x;
3
4     public Derp(int x) {
5         this.x=x;
6     }
7
8     public String toString() {
9         return new String("x=" + x);
10    }
11 }
12
13 public class WhatsPrinted06 {
14     public static void func(Derp d) {
15         d.x++;
16         System.out.println(d);
17     }
18
19     public static void main(String args[]) {
20         Derp d1 = new Derp(10);
21         func(d1);
22     }
23 }

```

(f) x=11

```

(g) class Derp {
2     public int x;
3
4     public Derp(int x) {
5         this.x=x;
6     }
7
8     public String toString() {
9         return new String("x=" + x);
10    }
11 }
12
13 public class WhatsPrinted07 {
14     public static void func(Derp d) {
15         d.x++;
16     }
17
18     public static void main(String args[]) {
19         Derp d1 = new Derp(10);
20         func(d1);

```

```

21         System.out.println(d1);
22     }
23 }

```

(g) x=11

```

(h) class Derp {
2     public int x;
3
4     public Derp(int x) {
5         this.x=x;
6     }
7
8     public String toString() {
9         return new String("x=" + x);
10    }
11 }
12
13 public class WhatsPrinted08 {
14     public static void func(Derp d) {
15         d = new Derp(222);
16     }
17
18     public static void main(String args[]) {
19         Derp d1 = new Derp(10);
20         func(d1);
21         System.out.println(d1);
22     }
23 }

```

(h) x=10

3 Thinking about the memory hierarchy

In class, we discussed the computer storage hierarchy, and the fact the types of storage high in the “pyramid” given in Chapter 1 (*e.g.* registers, L1 cache, ...) are orders of magnitude faster than the types at its base (*e.g.* hard drives, optical storage, ...).

Some access times are given for current hardware¹. (Recall that the abbreviation ns is for nanoseconds and ms is for milliseconds.)

So, looking at the table, we see for example if we’d like to read something stored in a L1 cache, it should take about a half of a nanosecond to fetch it, *etc.*

location	access time
L1 cache	0.5 ns
L2 cache	7 ns
RAM	100 ns
hard drive	10 ms
DVD	140 ms

In our every day lives, we’re not used to dealing with units of time this small. To give us a sense of how much slower devices at the bottom of the pyramid are from the top, let’s pretend instead that access times are in units to which we’re more accustomed.

Using the same ratios as we had in the table above, fill in the following *fake* access times. For example, in the table of real access times, we see that it takes 0.5 ns on average to read from a L1 cache, and 7 ns to read from a L2 cache. L2 caches are on average 14 times slower. If we pretend that it takes 1 second to read from an L1 cache, it would take 14 seconds to read from a L2 cache. Using this same procedure, fill in the rest of the table.

location	<i>fake access time</i>
L1 cache	1 sec
L2 cache	14 sec
RAM	200 sec
hard drive	20,000,000 sec
DVD	280,000,000 sec

¹Most of these come from a talk given at a recent conference on systems <http://www.cs.cornell.edu/projects/ladis2009/talks/dean-keynote-ladis2009.pdf>.

Now, let's do the same thing, but instead of time, let's consider distance. Pretend that we're baking a cake. We're doing all of our work on the kitchen counter, and different ingredients that we need for the cake are found in different storage places. If the L1 cache is the cabinet two feet above the counter, the L2 cache would be 14 times farther away, so it's 28 feet away (maybe in our basement?). Using the same ratios as before, fill in the rest of the chart:

location	<i>fake distance</i>
L1 cache	2 feet
L2 cache	28 feet
RAM	400 feet
hard drive	40,000,000 feet
DVD	560,000,000 feet