1 Vocab

- The vocabulary term for the given definition "the formal parameters in the function header are copies of the actual parameters passed to them, and changing the formal parameters won't change the actual parameters."
- There is one type of C pointer that is special, in that it can point to any type of variable. What is this?
- The hardware that enforces a restriction when a program tries to access a part of the memory that it does not own.
- Error that occurs when you try to dereference an address that is not divisible by 4.
- A bug where dead storage progressively builds up until the runtime heap is full.
- The range of values for an unsigned byte.
- The range of values for 13 bits and it represents a signed number.
- The approximation of 2^{30}
- $\sqrt{2^{16}} * 2^3$
- What is the representation of -5 in ones complement assume 16 bits.
- Translate this binary number into octal: 10100011101101100011100010110111000 01
- Convert the previous question into hex.
- Create a mask to get the 3rd to 5th bits (inclusive) in a byte.
- Bug that might show up when you move your code to a different machine.
- Draw out the argv for a program that runs with `./a.out alpha beta gamma`
- 2 terms for "area of memory on the RTS for one active function"
- True or false, can we create a pointer to a function?
- How would you declare a function pointer?
- What is the `sizeof(char *)` on a 32 bit machine.
- The internal data structure that has detailed information on each running process.
- Three possible return values of fork are?
- A function that is called with different types and number of parameters.

- The part of memory that program code goes into.
- The part of memory that holds global data.
- A function specifically devoted to initializing a data structure.
- A function specifically devoted to cleaning up a data structure.
- A `typedef` without providing any fields at all
- A list of unused memory of a specific structure waiting for resuse.

2 Unix Commands

- Create a hardlink called A in your home directory, pointing to a file B in the present directory.
- Make a new directory called `backups`
- Change *file1*'s name to *file2*
- Run the executable *myprog* with input coming from *input.1* with output going to *output.1*
- Change the permissions of *file1* to be read, write, and executable by only the owner, no privileges by the group nor to the world. (Your root should be set to this)
- List all files, including "dot" files, in the current directory

3 Diagramming

Given the code below, draw out the contents of the all the variables used at the stated point. If it is unknown, draw a question mark '?'. Please follow the pointer-drawing conventions used in the lecture:

```
int ****foo() {
    int i, j, k;
    char ****grid;

    grid = calloc(sizeof(char ***), 2);

    for (i = 0; i < 2; i++) {
        grid[i] = calloc(sizeof(char **), 1);
        grid[i][0] = calloc(sizeof(char *), 4);
        for (j = 0; j < 4; j++) {
            grid[i][0][j] = malloc(sizeof(char), j);
        }
    }

    // Draw a diagram from this point

    return grid;
}
```

4 C

1. Please list the range of sizes for the types in C. If e.g. if we ran `sizeof` on them.

- char
- short
- int
- int *
- char *
- void *

2. Write a program that returns 1 if the inputted string begins with a letter (upper or lowercase) and the rest of the string is only alphanumeric characters or an underscore ('_'), else it returns 0.

5 Adv C

1. Given the initial dictionary to be letters 0 - 9, Where 0 begins as code 0 and 9 is code 9, compress this sequence.

```
08134723492343
```

3. Below is an output of Compress. Show the original uncompressed file. As a hint, the contents of the file are at least 8 characters, and may aptly describe your reaction to this test. Recall that the ASCII lowercase letters start with code 97 ('a').

```
31184C90 38240000
```

4. What would you do to make this header file (temp.h) to be only included once. Please include your additions below.

```
#define RANDOM_NUMBER 1234567

extern int a;
int b;
```

5. Using at most three nonblank characters, fill in the call of test, so that the printf will happen. (The limit is actually a hint; more complex answers are likely to be wrong.) Explain why the printf happens. Be sure that your call will not generate a compile error; this is an easy mistake to make.

```
#define test(a) { \
   int temp = a; \
   if (temp == (a + 2)) \
   printf("Impossible?\n"); \
}
test(     )
```

6 Systems

1. Create a program that runs Make, redirects its output (both stdout and stderr) to nothing (/dev/null). Then runs the program which is passed in via commandline arguments.

E.g.

```
$> ls
. .. ./a.out Makefile execName.c
$> ./a.out execName
```

2. How can you write to a pipe and then clear what you wrote to the pipe?

3. If the following program is compiled to an executable a.out and then run with command:

```
a.out a.out echo a.out echo a.out
```

How many execvp callls in all will result, and what will the output be?

```
int main(int arc, char **argv) {

   int n;

   for (n = 1; n < argc; n++)

   return 0;

}
```