

Review

CSC311 Systems Software

03 December 2015

1. What is a kernel?
2. How do programs interact with the kernel?
3. Through what mechanism can an operating system give every process secure access to every address in the computer's memory space?
4. How big must the array of characters in a C program that holds the word "word" be?
5. What does the second statement in code do?

```
int year = 1968;  
int *yp = &year;
```

6. What is wrong with this code?

```
int * f() {  
    int year = 1968;  
    int *yp = &year;  
    return yp;  
} // f()
```

7. What is wrong with this code?

```
struct process {  
    double serviceTime;  
    double interarrivalTime;  
    double arrivalTime;  
    double serviceStartTime;  
    double serviceCompleteTime;  
} // process
```

```
typedef struct process Process, *ProcessPointer;
```

```
ProcessPointer pp = (ProcessPointer) malloc(sizeof(ProcessPointer));
```

8. What is wrong with this code?

```
void spaceRace( int year ) {
    switch( year ) {
        case 1957:
            printf( "The first satellite was launched in %4d.\n",
                year );
        case 1961:
            printf( "The first person orbited the earth in %4d.\n",
                year );
        case 1965:
            printf( "The first person walked in space in %4d.\n",
                year );
        case 1968:
            printf( "The first people orbited the moon in %4d.\n",
                year );
        case 1969:
            printf( "The first people walked on the moon in %4d.\n",
                year );
    } // switch
} // spaceRace( int )
```

9. What is wrong with this code?

```
#define THRESHOLD = 100;
```

10. What does this program do? Use the data that it generates to plot a histogram. Use a spreadsheet program.

```
#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

#define MAX 10.0
#define BINS 100
#define TRIALS 1000000

int main( int argc, char** argv ) {
    time_t t;
    srand( (unsigned) time(&t) );

    int histogram[BINS];

    int i;
    for( i = 0; i < BINS; i++ ) {
        histogram[i] = 0;
    }
}
```

```

} // for

double mean = 1.0;
for( i = 0; i < TRIALS; i++ ) {
    double x = -mean * log(((double) rand())/RANDMAX );
    //printf( "%8.4f\n", x );
    if( x < 10.0 ) {
        histogram[(int) floor(x/MAX * BINS)]++;
    } // if
} // for

for( i = 0; i < BINS; i++ ) {
    printf( "Number_of_values_between_%8.4f_and_%8.4f_is_%4d\n",
           (i * MAX/BINS), ((i + 1) * MAX/BINS), histogram[i] );
} // for
} // main( int, char** )

```