



Introduction to Computer Systems

Final Exam, Spring 2022

Seoul, June 21, 2022

| | | | | | | | | |
|-------------------|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|
| Student Number: | <input type="text"/> | <input type="text"/> | <input type="text"/> | <input type="text"/> | <input type="text"/> | <input type="text"/> | <input type="text"/> | <input type="text"/> |
| First Name: | <input type="text"/> | | | | | | | |
| Last Name: | <input type="text"/> | | | | | | | |
| Date of Birth: | <input type="text"/> | | | | | | | |
| Program of Study: | <input type="text"/> | | | | | | | |

| Task | 1.1 | 2.1 | 2.2 | 3.1 | 3.2 | 4.1 | Σ |
|----------------|-----|-----|-----|-----|-----|-----|----------|
| Maximal Score | 10 | 4 | 4 | 6 | 6 | 10 | 40 |
| Attained Score | | | | | | | |
| Correction | | | | | | | |

Remarks:

- The duration of the exam is **75 minutes**.
- Please write down your name and student id at the top of **every page**.

Good Luck!

Task 1: Code Injection Attack

1.1 Buffer Overflow

10 points

We have the following function that reads typed string using function **gets**.

```
/* Echo Line */
void echo(){
    char buf[4]; /* Way too small! */
    gets(buf);
    puts(buf);
}

void call_echo(){
    echo();
}
```

1. How could we change the return address of stack frame call_echo()? 4 points
2. To prevent this, there are three general ways: avoiding overflow vulnerabilities, system-level protection, and stack canaries. Describe how each of them protects the code from the code injection attack. 6 points

Task 2: Code Optimization

2.1 Simple Code

4 points

Please optimize the code below and explain why they improve the performance.

```
void count_chars ( int *a, char *b ){

    long i, j;
    for ( i = 0; i < strlen(b); i++){
        a[i]=0;
        for ( j = i; j < strlen(b); j++){
            if ( b[i]==b[j] )
                a[i]+=1;
        }
    }
}
```

2.2 Loop Unrolling

4 points

There are two ways to implement loop unrolling (1 and 2). Which way does get the better performance and why?

```
float x1 = 1, x2=1;
for (i= 0; i< limit; i+=2) {
    1. x1 = x1 * (d[i] * d[i+1]);
    2. x1 = x1 * d[i]; x2 = x2 * d[i+1];
}
return x1 * x2;
```

Task 3: Memory Hierarchy

3.1 Disk Access

6 points

Disk access comprises of three steps: seek, rotational latency and read.

1. Explain each step of disk access. 3 points
2. Suppose the rotational rate of a disk is 3600 RPM, the average seek time is 6 ms, and the average number of sectors per track is 200. Compute the disk access time. 3 points

3.2 Locality

6 points

1. Specify the definition of temporal and spatial locality.

3 points

2. Reorder **for** statements to maximize the locality and explain why.

3 points

```
int sum_array_3d(int a[M][N][N]){
    int i , j , k , sum = 0;
    for ( i = 0; i < N; i ++ )
        for (j = 0; j < N; j ++ )
            for ( k = 0; k < M; k ++ )
                sum +=a[k][i][j];
    return sum;
}
```

Task 4: General Understanding

4.1 Small Questions

10 points

1. What is the working set?

3 points

2. Explain the performance changes regarding the number of the working set and main memory size.

3 points

3. Compared to dynamic linking, explain disadvantages of static linking.

4 points