CS 361 - Spring 2025 - Version (Practice)

Name: ______________________________ UIN: ____________________

- Here are some test questions for you to look at.
- If you have any questions you can post on Piazza.
- We will go over some (possibly all) of these on the review day.
- I might add more questions and if that happens then I will make an announcement.
- On the real exam I will have points listed next to the questions.

1. Consider the following source file below.

```c
int flag;
double count = 0.0;
static void function2(void);
~~extern int alternate~~;
void function1(void);

int main(void)
{
    static int other = 5;
    function1();
    function2();
    return other;
}

static void function2(void){return;}
```

(a) List the defined symbols that will be generated from compiling this source file into a relocatable object file.

flag, count, other.0, function 2, ~~function 1~~, ~~function2.0~~, main

does not defined

(b) Which defined symbols can be referenced by another source file without generating a linker error?

function 1, function 2, alternative

2. Consider the output from `readelf` below that contains the symbol table from a relocatable object.

```
Symbol table '.symtab' contains 10 entries:
   Num:    Value          Size Type    Bind   Vis      Ndx Name
     0: 0000000000000000     0 NOTYPE  LOCAL  DEFAULT  UND
     1: 0000000000000000     0 FILE    LOCAL  DEFAULT  ABS symtab.c
     2: 0000000000000000     0 SECTION LOCAL  DEFAULT    1 .text
     3: 0000000000000000     0 SECTION LOCAL  DEFAULT    4 .bss
     4: 0000000000000000     4 OBJECT  LOCAL  DEFAULT    4 myvar.0
     5: 0000000000000000     4 OBJECT  GLOBAL DEFAULT    3 val1
     6: 0000000000000000    11 FUNC    GLOBAL DEFAULT    1 func1
     7: 000000000000000b    43 FUNC    GLOBAL DEFAULT    1 main
     8: 0000000000000000     0 NOTYPE  GLOBAL DEFAULT  UND func3
     9: 0000000000000000     0 NOTYPE  GLOBAL DEFAULT  UND func2
```

(a) What are the names of the defined functions in the object?

(b) What is a possible type for the object `val1`?

(c) What section is `main` in?

3. Consider the pair of source code files below.

srca.c

```c
char capitalize(char c);

int main(void)
{
    char c = 'a';
    c = capitalize(c);
    return 0;
}
```

srcb.c

```c
int count = 0;
int validate(char x);

char capitalize(char x)
{
    if (validate(x)) {
        count += 1;
        return x - 0x20;
    }
    return 0x00;
}
```

no return (no code)

Running the command below will attempt to build the project. Will it succeed? Explain how you know. If the command will not succeed explain which step in the build process will fail and how you would modify the source files above so that it succeeds.

```
$ gcc srca.c srcb.c
```

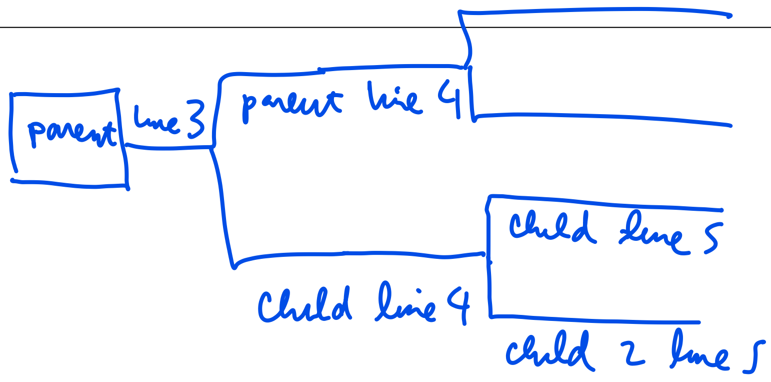
int validate (char x) {

}

4. Write a function that accepts as an argument a pointer to the beginning of the payload, for a block allocated by `malloc`. It should return 1 if that block is allocated otherwise 0. You can assume that there is always a block immediately following the block your function is checking. You can assume that all blocks are 32 bytes in size.

5. Consider the following code segment below. What will be output to `STDOUT` from running it?

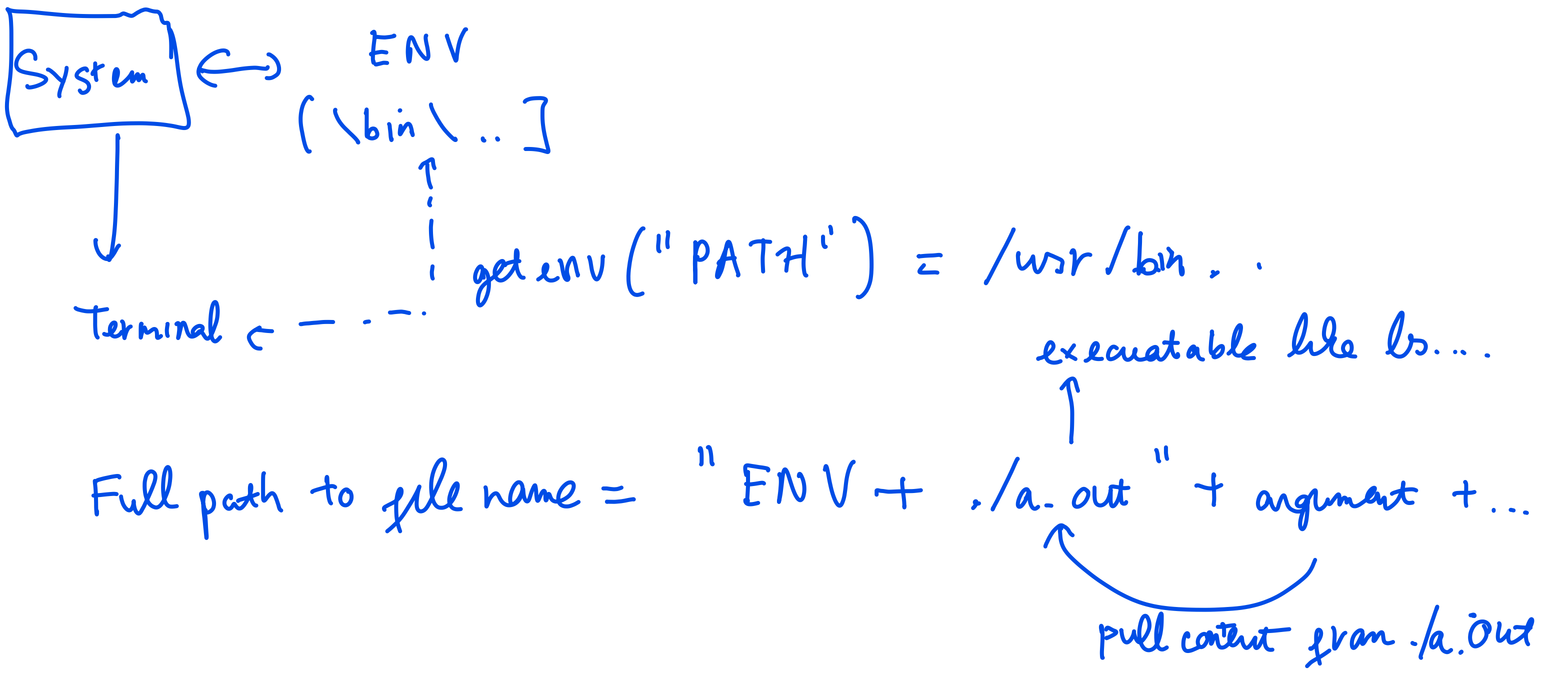
```
1 int main(void)
2 {
3     fork();
4     fork();
5     printf("hello\n");
6     return 0;
7 }
```



6. Shell Commands - The program `file` takes one argument which specifies the path to a file. It determines the file's type and prints that information to STDOUT. See the example below.

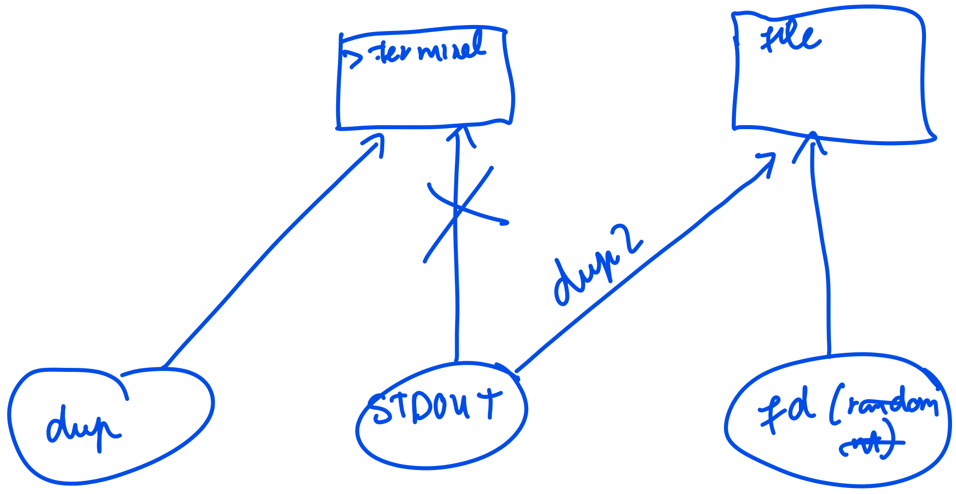
```
$ file main.tex
main.tex: LaTeX 2e document, ASCII text
```

(a) Assume that `file` is in the directory `/usr/bin`. Write a code segment that calls `execv` to run `file` and pass as an argument the filepath `/home/test/temp.txt`.



System
ENV
[\bin\ ..]
getenv("PATH") = /usr/bin . .
Terminal
executable like ls....
Full path to file name = "ENV + ./a.out" + argument + ...
pull content from ./a.out

(b) Write a code segment that calls `execv` to run `file` with the argument `/home/test/other.txt` and writes the result to file `./output.txt` instead of STDOUT.



dup = duplicate (STDOUT) ⇒ dup pt to STDOT pt to

dup2 → redirect pointer

dup2(fd1, STDOUT)

7. Consider the memory system below. Assume all values are hexadecimal unless otherwise specified.

- virtual addresses are 8 bits
- pages are 8 bytes in size
- Below is a table of the relevant parts of the page table for a specific process.

VPN	Valid bit	PPN
00	0	–
01	0	–
02	0	–
03	1	3D
04	0	–
05	1	77
06	1	1F
07	0	–
08	1	E0
09	0	–

VPN	Valid bit	PPN
0A	1	CC
0B	1	93
0C	1	29
0D	0	–
0E	1	78
0F	1	F0
10	1	22
11	0	–
12	1	21
13	0	–

(a) Give two virtual addresses that would result in page hits. They must access different virtual pages. What are their corresponding physical addresses?

(b) Give two virtual addresses that would result in page faults. They must access different virtual pages.

(c) What are the two possibilities for how the kernel would resolve a page fault, that we discussed in class.