SP HW2

1. Modify the program in Figure 4.3 of the textbook to use stat() instead of lstat(). What changes if one of the command-line arguments is a symbolic link?

If one of the command-line arg is a symbolic link, `stat` will return information of the file referenced by the symbolic link, but not the information about the symbolic link.

2. Assume you change the file mode creation mask to 777 (octal), then create a regular file. What happens?

The file's permission will be `0000` (octal), which means neither the owner, group nor others has the permission of the file.

3. We mentioned in the class that, after a vfork(), the child process runs in the address space of the parent; what happens if the call to vfork() is from a function other than main() and the child does a return from this function after (returning from) vfork()? Write a test program and report what you have observed. (Please provide your test program and the results here).

We might encounter undefined behaviors. In my case it's segmentation fault.

Test program:

```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

void child_function() {
    printf("Before vfork()\n");

    pid_t pid = vfork();
    if (pid < 0) {
        perror("vfork failed");
        exit(EXIT_FAILURE);
    } else if (pid == 0) {
        // In the child process
        printf("In child process (pid = %d)\n", getpid());
        // Return to parent without exec or _exit
        return;
    } else {
        // In the parent process
        printf("In parent process (pid = %d), child pid = %d\n", ge
    }
}

int main() {
    child_function();  // Calling function containing vfork
    printf("Back in main after child_function call\n");
    return 0;
}
```

Output:

```
Before vfork()
In child process (pid = 1310400)
Back in main after child_function call
In parent process (pid = 1310399), child pid = 1310400
[1]    1310399 segmentation fault (core dumped)  ./vfork
```

4. Figure 4.20 from the textbook shows that the unlink() function modifies the file's changed status time (time). Explain why this happens.

Because the file status time is the last-change time of the i-node, and unlinking the file will change the number of the links of the file, affecting the i-node.