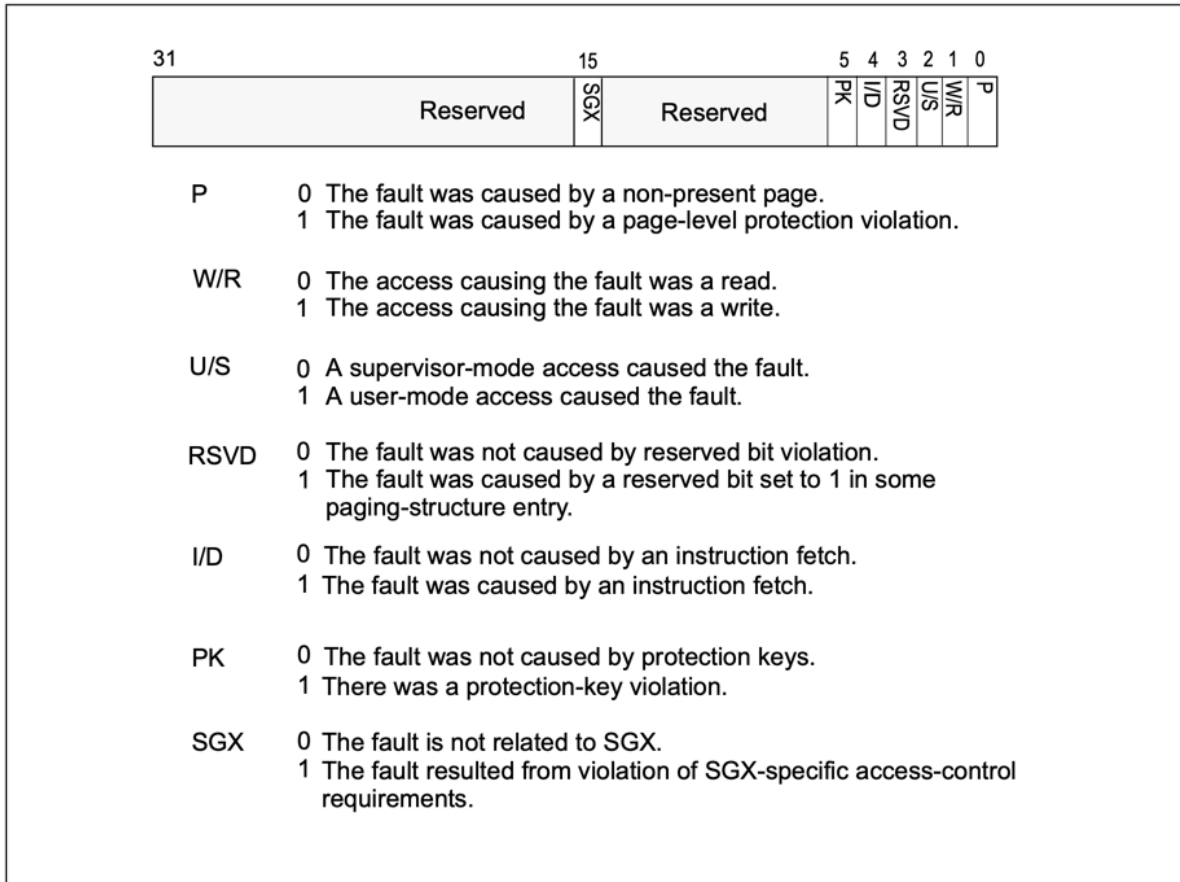


**Figure 6-4. Stack Usage on Transfers to Interrupt and Exception-Handling Routines**



**Figure 4-12. Page-Fault Error Code**

```

1  CS5600, Handout week11.a
2
3  /* file: mmap.c */
4
5  #include <fcntl.h>
6  #include <stdio.h>
7  #include <stdlib.h>
8  #include <sys/mman.h>
9  #include <sys/stat.h>
10 #include <sys/types.h>
11 #include <unistd.h>
12
13 void mmapwrite(int fd, int size);
14 void normalwrite(int fd, int size);
15
16 int main(int argc, char **argv) {
17     struct stat stat;
18     int fd;
19
20     if (argc != 2) { // Check for required cmd line arg
21         printf("usage: %s <filename>\n", argv[0]);
22         exit(0);
23     }
24
25     /* Copy input file to stdout */
26     if ((fd = open(argv[1], O_RDONLY, 0)) < 0)
27         perror("open");
28
29     fstat(fd, &stat);
30
31     // option 1
32     mmapwrite(fd, stat.st_size);
33
34     /* // option 2
35      * normalwrite(fd, stat.st_size);
36      */
37
38     close(fd);
39
40     return 0;
41 }
42
43 void mmapwrite(int fd, int size) {
44
45     /* Ptr to memory mapped area */
46     char *bufp;
47
48     bufp = mmap(NULL, size, PROT_READ, MAP_PRIVATE, fd, 0);
49
50     write(STDOUT_FILENO, bufp, size);
51
52     return;
53 }
54
55
56 void normalwrite(int fd, int size) {
57
58     char *buf = malloc(size);
59
60     read(fd, buf, size);
61
62     write(STDOUT_FILENO, buf, size);
63
64     return;
65 }

```

## Question:

Which runs faster, option 1 or option 2? by how much?

## Exercise:

Try to run both options by yourself:

```

$ cat /dev/urandom | head -c 1000000000 > 1G.file
$ make mmap
$ time ./mmap 1G.file > /dev/null

```

```

$ vim mmap.c
// switch to option 2
$ make mmap
$ time ./mmap 1G.file > /dev/null

```