CSCI442 Homework 3

Ally Smith (Section A)

February 23, 2022

Question 1.

Consider the following code:

```
char buf[100];
int fd0 = open(“foo.txt”, O_RDONLY);
int fd1 = open(“bar.txt”, O_WRONLY | O_CREAT | O_APPEND,
                0666)
printf(“Hola!\n”);
dup2(fd1, STDOUT_FILENO);
printf(“Como estas?\n”)
dup2(fd0, STDIN_FILENO);
for (;;) {
    ssize_t bytes_read = read(STDIN_FILENO,
                              buf, sizeof(buf));
    if (bytes_read == 0)
        return 0;
    ssize_t bytes_to_write = bytes_read;
    while (bytes_to_write > 0)
        bytes_to_write -= write(STDOUT_FILENO,
            buf + bytes_read - bytes_to_write,
            bytes_to_write);
}
```

The program is invoked like so:

```
$ cat foo.txt
Muy bien!
$ cat bar.txt
Buenos dias!
$ ./a.out
```

(a) What was printed on standard output?

```
Hola!
Buenos dias!
Como estas?
Muy bien!
```

(b) What does foo.txt contain?

```
Muy bien!
```

(c) What does bar.txt contain?

```
Buenos dias!
```

Question 2.

Consider the following code:

```
int pipefd[2];
pipe(pipefd);
```

Assume that the operating system assigns the file descriptor `3` for the read end of the pipe, and the file descriptor `4` for the write end of the pipe.

(a) What is `pipefd[0]` after this operation is performed?

`3`

(b) What is `pipefd[1]` after this operation is performed?

`4`

Question 3.

The following state transition table is a simplified model of process management, with the labels representing transitions between states of `READY`, `RUN`, `BLOCKED`, and `NONRESIDENT`.

$\frac{to}{from}$	**READY**	**RUN**	**BLOCKED**	**NONRESIDENT**
READY	–	1	–	5
RUN	2	–	3	–
BLOCKED	4	–	–	6

3.1) `READY` ⇒ `RUN`

A process could go through this transition if it is the next process in line to be executed by the operating system.

3.2) `RUN` ⇒ `READY`

This transition could occur if a process times out.

3.3) `RUN` ⇒ `BLOCKED`

This transition could occur if a process has to wait for an I/O operation.

3.4) `BLOCKED` ⇒ `READY`

A process could go through this transition if an I/O event occurs that a process was waiting on.

3.5) `READY` ⇒ `NONRESIDENT`

A process could go through this transition if it is suspended into secondary memory.

3.6) `BLOCKED` ⇒ `NONRESIDENT`

A process could go through this transition if it is suspended into secondary memory.

Question 4.

Assume that at time 5 no system resources are being used except for the processor and memory. Now consider the following events:

T5: P1 executes a command to read from disk unit 3.
T15: P3's time slice expires.
T18: P4 executes a command to write to disk unit 3.
T20: P2 executes a command to read from disk unit 2.
T24: P3 executes a command to write to disk unit 3.
T28: P3 is swapped out.
T33: An interrupt occurs from disk unit 2: P2's read is complete.
T36: An interrupt occurs from disk unit 3: P1's read is complete.
T38: P5 terminates.
T40: An interrupt occurs from disk unit 3: P3's write is complete.
T44: P3 is swapped back in.
T48: An interrupt occurs from disk unit 3: P4's write is complete.

For each time 25, 35, and 45, identify which state (or possible states) each process is in. If a process is blocked, further identify the event on which it is blocked.

	Time 25	Time 35	Time 45
P1	`BLOCKED`, DU3	`BLOCKED`, DU3	`READY`
P2	`BLOCKED`, DU2	`READY`	`READY`
P3	`BLOCKED`, DU3	`NONRESIDENT`	`NONRESIDENT`
P4	`BLOCKED`, DU3	`BLOCKED`, DU3	`BLOCKED`, DU3
P5	`RUN`	`RUN`	`EXIT`