L.1.7 System Calls

↳ programmatic way in which a computer program requests a service from the kernel of the OS it is executed on.

For ex, we want to access a file in a C++ program, that file will be accessed using a system call.

Categories of system calls

(i) File related - open (), read(), write(), close (), create(),

(ii) Device related - read(), write(), reposition, ioctl (input output control), fcntl (file control)
↓
accessing hardware

(iii) Information related - get Pid, attributes, get system Time, data.
↓
metadata
↓
process related information.
(filesize, extension)

(iv) Process control - Load, execute, abort, *fork, wait, signal, allocate etc.

↑ creating multiprocessing environment.

(v) communication - pipe(), create/ delete connections, shmget().

$\downarrow$

processes communicating with each other

L-1.8 Fork System Call

Fork system call $\rightarrow$ create a child process

Fork() $\rightarrow$ 0 child
$\rightarrow$ +1 parent
$\searrow$ -1 child x (not created)

$$\begin{tikzcd}
 & P \arrow[d] & \\
 & \text{fork} \arrow[dl] \arrow[dr] & \\
C_1 & & P
\end{tikzcd}$$

Now, they would be parallely executed

ex.
```
main () {
    fork();
    printf ("Hello")
}
```

output

```
Hello
Hello
```

ex.
```
main() {
    fork();
    fork();
    print(Hello);
}
```

$$\begin{tikzcd}
 & & P \arrow[d] & & \\
 & & \text{fork} \arrow[dl] \arrow[d] & & \\
 & C_1 \arrow[dl] \arrow[d] & P \ \text{fork} \arrow[d] \arrow[dr] & & \\
C_2 & C_1 & C_3 & P &
\end{tikzcd}$$

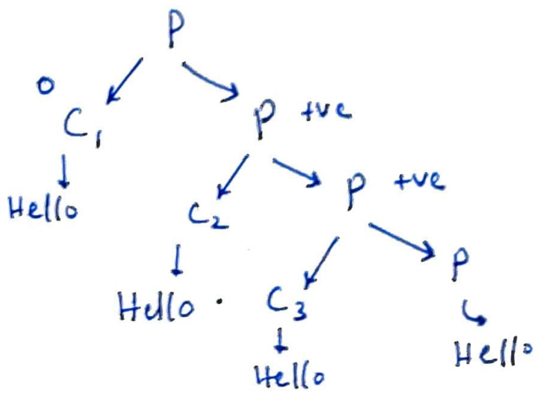
output :

```
Hello
Hello
Hello
Hello
```

L-1.9 Questions on Fork system call

```
int main() {
    if (fork() && fork())
        fork();
    printf("Hello");
}
```

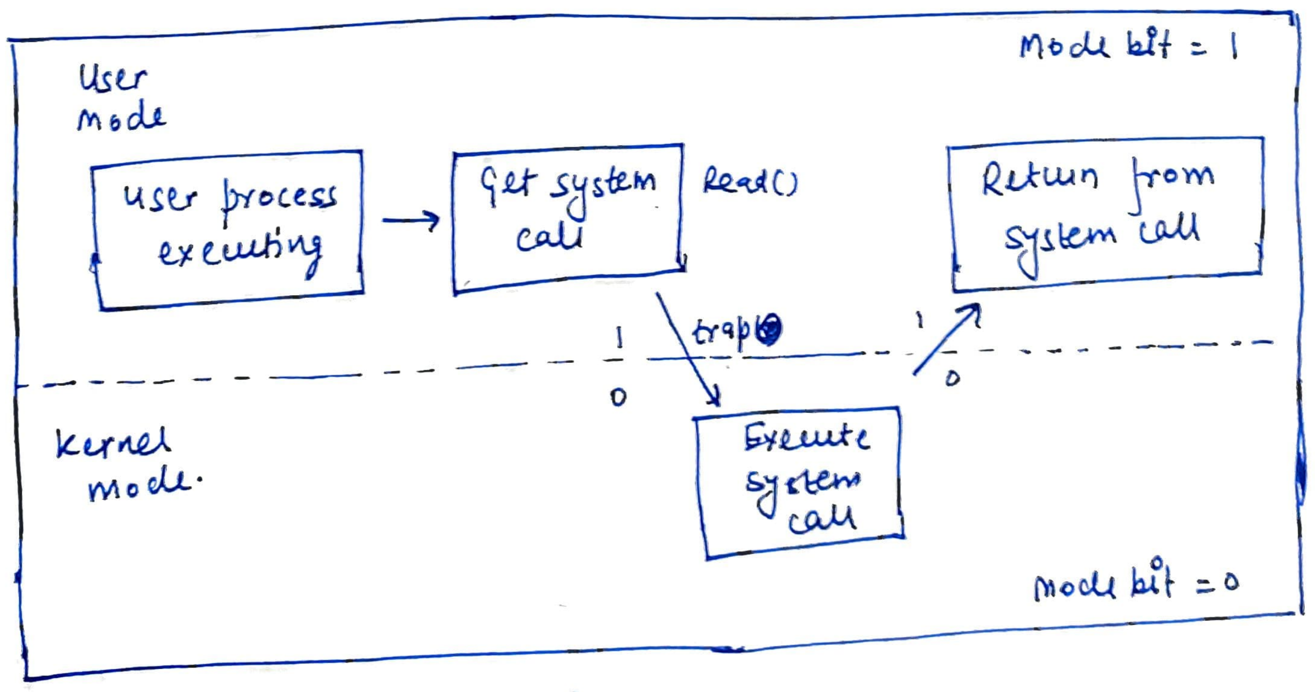
How many times is hello printed.?



ans - 4

L-1.10 User Mode and kernel mode

Suppose there is a program in C++ that wants to read a file in Hard disk.



HDD