Forks, pipes and shared memory

Edward Zhang

SOFTENG 370 T2

Fork

Fork creates a new process, which becomes a child of the caller. Both processes then contiune execution from the line where fork was called.

1. Fork returns negative when forking failed
2. Fork returns zero within the new child process
3. Fork returns a positive value with the process id within the parent. This is a `pid_t`

Fork example

```
int main(void) {
  printf("Hello from pid: %d\n", getpid());
  int fork1 = fork(); // actually a pid_t
  printf("Forked! I’m pid %d and fork returned %d\n", getpid
      (), fork1);
  int fork2 = fork(); // actually a pid_t
  printf("Forked again! I’m pid %d and fork returned %d\n",
      getpid(), fork2);
  return 0;
}
```

How many times are each line printed, and when is the return zero and non-zero?

Forked data

When forked, all pages allocated for a process are copied. This includes pages that store the stack, or memory on the heap (i.e. from malloc).
Copy-on-Write (CoW) is used so unless the child process modifies the data, a needless copy is not made.

A note on addresses

Consider this code. What's returned?

```c
int main(void) {
    int *data = malloc(sizeof(int));
    *data = 9001;
    int fork1 = fork(); // actually a pid_t
    if (fork1 == 0) {
        *data = 9000;
    }
    printf("Forked! I'm pid %d and the data is at %p is %d\n
        ", getpid(), data, *data);
    return 0;
}
```

A note on addresses

Consider this code. What's returned?

```c
int main(void) {
    int *data = malloc(sizeof(int));
    *data = 9001;
    int fork1 = fork(); // actually a pid_t
    if (fork1 == 0) {
        *data = 9000;
    }
    printf("Forked! I'm pid %d and the data is at %p is %d\n
        ", getpid(), data, *data);
    return 0;
}
```

How do we have different data at the same memory address?

A note on addresses

Consider this code. What’s returned?

```c
int main(void) {
    int *data = malloc(sizeof(int));
    *data = 9001;
    int fork1 = fork(); // actually a pid_t
    if (fork1 == 0) {
        *data = 9000;
    }
    printf("Forked! I’m pid %d and the data is at %p is %d\n
        ", getpid(), data, *data);
    return 0;
}
```

How do we have different data at the same memory address?
A: Virtual memory space is unchanged, even if a copy is made in physical memory.