Multiprocessing: processes; fork & wait; exec

COSC 208, Introduction to Computer Systems, 2022-04-21

Waiting for processes

Q1: *What are all possible outputs of this program?*

```c
int main() {
    int pid = fork();
    if (pid == 0) {
        printf("Child\n");
    } else {
        wait(NULL);
        printf("Parent\n");
    }
    return 0;
}
```

Q2: *What are all possible outputs of this program (assuming the new process has PID 13346)?*

```c
int main() {
    int pid = fork();
    printf("A %d\n", pid);
    if (pid == 0) {
        printf("B\n");
    } else {
        wait(NULL);
        printf("C\n");
    }
}
```

Running a different program

Example program

```c
int main(int argc, char **argv) {
    printf("Begin\n");
    int pid = fork();
    if (pid == 0) {
        printf("Child\n");
        char *cmd[] = { "/usr/bin/date", NULL };
        execv(cmd[0], cmd);
    } else {
        printf("Parent\n");
    }
    printf("End\n");
    return 0;
}
```

Q3: *What is the output produced by running* `./progA`*, assuming no errors occur?* **progA:**

```c
int main() {
    pid_t a = fork();
    if (a == 0) {
        execv("./progB", NULL);
        printf("A 2nd gen\n");
        return 0;
    } else {
        wait(NULL);
        printf("A 1st gen\n");
        return 0;
    }
}
```

progB:

```c
int main() {
    pid_t b = fork();
    if (b == 0) {
        printf("B 2nd gen\n");
        return 0;
    } else {
        wait(NULL);
        printf("B 1st gen\n");
        return 0;
    }
}
```

Practice

QA: *What does the following code output?*

```c
int main(int argc, char **argv) {
    int value = 100;
    int pid = fork();
    if (pid == 0) {
        value -= 50;
    } else {
        value += 50;
    }
    printf("My value is %d\n", value);
    return 0;
}
```


QB: *What does the following code output?*

```c
int main(int argc, char **argv) {
    printf("Begin\n");
    int pid = fork();
    if (pid == 0) {
        printf("Child\n");
        return 0;
    } else {
        printf("Parent\n");
    }
    printf("End\n");
}
```

QC: *How would you modify the above program such that* `Child` *always prints before* `Parent`*?*