

والد

ع فایل ← ع فرزند هم دارد

```
int inchild = 0; int status = 0;
pid_t child[E];
int process_id = 0;
for (int i = 0; i < E; i++) {
    sprintf(fout[i], "%d.txt", i+1);
}
```

اسم فایل ها
مسئله کد را

fgets

همه ی سطر درستی
گرفته

```
for (int i = 0; i < E; i++) {
    scanf("%s", fname[i]);
}
```

① اسم فایل ها
از stdlib
می گیریم

```
for (int i = 0; i < E; i++) {
    process_id = i;
    child[i] = fork();
    if (child[i] == 0) {
        inchild = 1;
        break;
    }
}
```

آیدی هر پروسه را بخوان
index ی که داریم
تا بتوانیم در حلقه while
اطلاعاتی را بخوانیم

② در این قسمت
پروسه ها را ایدار
می کنیم و در آرایه
زنجیره ای کنیم

فرزند → while (inchild == 1) {

```
    i = process_id;
    execl("/bin/grep", "grep", "print", fname[i], ">",
    fout[i], NULL);
}
```

→ اجرای execl
در هر پروسه
فرزند

والد → while (inchild == 0) {

```
    for (int i = 0; i < E; i++) {
        wait(&status);
    }
    for (int i = 0; i < E; i++) {
        cat_file(fout[i]);
    }
}
```

→ مسأله کامپلکسیتی
پروسه های فرزند ی را جمع

ب تعداد فرزند ها
wait را موا
می کنیم

→ فایل فرزند ی