ساختار داده‌ی صف با آرایه‌ی دایره‌ای و اندازه‌ی ثابت

□ نسخه‌ی چاپی ۱ هدیه‌ی ۶ صفحه‌ای

مهندس کامران بیات

شماره‌ی ۳ مهندسی نرم‌افزار — صفحه‌ی ۱۲

صف یکی از ساده‌ترین ساختارهای داده است اما در سیستم‌های بلادرنگ همین سادگی است که تفاوت بین کار کردن و از کار افتادن را رقم می‌زند. در این شماره می‌خواهیم ببینیم چرا آرایه‌ی دایره‌ای با اندازه‌ی ثابت در بسیاری از کاربردها بر صف پویا ترجیح داده می‌شود و چه هزینه‌ای برای این ترجیح پرداخته می‌شود.

چرا صف پویا کافی نیست

در صف پویا هر عنصر جدید با تخصیص حافظه‌ی تازه ساخته می‌شود و هر عنصر حذف‌شده به سیستم‌عامل برگردانده می‌شود. این کار روی کاغذ درست است اما در عمل سه مشکل جدی ایجاد می‌کند. تخصیص حافظه زمان می‌برد و زمان آن قابل پیش‌بینی نیست. تکه‌تکه شدن حافظه به مرور زمان باعث می‌شود با وجود آزاد بودن حافظه، تخصیص بزرگ شکست بخورد. و در سیستم‌های بلادرنگ هر دو مورد یعنی تأخیر غیرقابل پیش‌بینی و شکست تخصیص ممکن است به فاجعه منجر شود.

آرایه‌ی دایره‌ای این مشکل‌ها را با یک فرض ساده حل می‌کند: اندازه‌ی صف از ابتدا مشخص است و حافظه‌ی آن یک بار و برای همیشه تخصیص داده می‌شود. اندیس ابتدا و انتها با محاسبه‌ی باقی‌مانده روی اندازه جلو می‌روند و وقتی به انتهای آرایه می‌رسند دوباره از ابتدا شروع می‌کنند. هیچ تخصیص و آزادسازی در مسیر اجرا وجود ندارد.

ساختار داده و عملیات

ساختار داده شامل یک آرایه‌ی ثابت، اندیس ابتدا، اندیس انتها و شمارنده‌ی تعداد است. عملیات اصلی دو تا هستند: افزودن در انتها و برداشتن از ابتدا. هر دو عملیات در زمان ثابت اجرا می‌شوند و هیچ حلقه‌ای در آن‌ها وجود ندارد. این ویژگی برای سیستم‌های بلادرنگ ارزش بالایی دارد زیرا کران زمانی هر عملیات را می‌توان از پیش تضمین کرد.

```
enqueue(x):
    if count == capacity:
        return ERR_FULL
    buf[tail] = x
    tail = (tail + 1) % capacity
    count = count + 1

dequeue():
    if count == 0:
        return ERR_EMPTY
    x = buf[head]
    head = (head + 1) % capacity
    count = count - 1
    return x
```

پر شدن صف و راه‌های برخورد با آن

وقتی صف پر است دو راه وجود دارد. یا فرستنده منتظر می‌ماند تا جا خالی شود، یا داده‌ی تازه رها می‌شود. کدام راه درست است به کار سیستم بستگی دارد. در سیستم صوتی داده‌ی قدیمی بی‌فایده است و رها کردن آن بهتر است از نگه داشتن آن است. در سیستم کنترل موتور هر داده ارزشمند است و رها کردن آن خطرناک است. پس تصمیم مهندسی است نه تصمیم کدنویسی.

اندازه‌گیری و تنظیم

اندازه‌ی صف را نباید حدس زد. باید اندازه‌ی بیشینه‌ی صف در شرایط واقعی اندازه گرفت و سپس با ضریب اطمینان روی آن گذاشت. اگر صف هرگز بیش از ۶۰ درصد پر نشود اندازه‌ی آن بیش از حد است و حافظه هدر می‌رود. اگر صف اغلب پر شود یا داده رها می‌شود اندازه کافی نیست و باید بزرگ‌تر شود. این اندازه‌گیری باید در محیط واقعی انجام شود نه در محیط آزمایشگاه.

اشتباه‌های رایج

اشتباه رایج اول فراموش کردن بررسی پر بودن است. اشتباه دوم استفاده از اندیس خطی است که بعد از چند بار اجرا از آرایه بیرون می‌زند. اشتباه سوم نوشتن داده در خانه‌ی پر است بدون آن که داده‌ی قدیمی رها شود. هر سه اشتباه در کدهای واقعی دیده شده‌اند و هر سه می‌توانند سیستم را از کار بیندازند.

جمع‌بندی

آرایه‌ی دایره‌ای ساده‌ترین ساختار داده‌ی صف است و به همین سادگی کارآمدترین آن در بسیاری از کاربردهای بلادرنگ است. زمان ثابت، حافظه‌ی ثابت و رفتار قابل پیش‌بینی سه ویژگی‌ای است که سیستم‌های بلادرنگ به آن نیاز دارند و آرایه‌ی دایره‌ای هر سه را با هزینه‌ی کم فراهم می‌کند. در شماره‌ی آینده پیاده‌سازی این ساختار را در زبان C می‌بینیم.

پیشینه‌ی صف در سیستم‌های بلادرنگ