

## فصل ۱

# کدهای رده و حالات داخلی

زمانی که نویسه‌ها خوانده می‌شوند، تک به هر کدام کدهای رده اختصاص می‌دهد. این مکانیزم خواندن سه حالت داخلی دارد و تغییر از یک حالت به حالت دیگر تحت تأثیر کدهای رده نویسه‌ها در ورودی قرار دارد.

`\endlinechar` کد نویسه، نویسه پایان سطر به سطرهای ورودی اضافه می‌شود. در `InitEX` کد نویسه پیش‌فرض این نویسه ۱۳ است.

`\par` یک پاراگراف تمام می‌شود و تک وارد حالت عمودی می‌شود. این دستور همچنین با وجود سطرهای خالی، تولید می‌شود.

`\ignorespaces` تا زمانی که چیزی که یک نشانه فاصله نیست، مشاهده شود، این دستور می‌خواند و گسترش می‌دهد.

`\catcode` کدهای رده را پرس و جو می‌کند یا آن‌ها را اختصاص می‌دهد.

`\ifcat` آزمایش می‌کند که آیا دو نویسه، کدهای رده یکسانی دارند یا خیر.

`\spacefactor` دستور فاصله. همان مقداری فاصله وارد می‌کند که یک نشانه فاصله زمانی که `= \spacefactor` ۱۰۰۰ است، وارد می‌کند.

`\obeylines` ماکروی فرمت `Plain TEX` که پایان سطرها را مهم می‌کند.

\obeyspaces ماکروی فرمت Plain TeX که (بیشتر) فاصله‌ها را مهم می‌کند.

## ۱.۱ مقدمه

پردازشگر ورودی تک، سطرهای ورودی را، از یک فایل یا ترمینال، اسکن می‌کند و نشانه‌ها را از نویسه‌ها می‌سازد. پردازشگر ورودی، می‌تواند به عنوان یک ماشین حالات متناهی ساده با سه حالت داخلی در نظر گرفته شود. بسته به حالت داخلی ماشین، نحوه اسکن کردن ماشین ممکن است فرق داشته باشد.

## ۲.۱ پردازش اولیه

ورودی از یک فایل (یا از ترمینال) سطر به سطر اسکن می‌شود. تک روش خودش را برای پایان یک سطر ورودی دارد:

۱. یک سطر ورودی از یک فایل ورودی (منهای پایان‌رساننده سطر، در صورت وجود) خوانده می‌شود.

۲. فاصله‌های پشت سر هم، حذف می‌شوند.

۳. نویسه سرخط (با کد نویسه ۱۳)، بصورت پیش‌فرض <Enter> یا <return> (بسته به سیستم عامل) اضافه می‌شود. اگر مقدار \endlinechar منفی یا بیشتر از ۲۵۵ باشد، هیچ نویسه‌ای اضافه نمی‌شود. دقیقاً مانند اینکه پایان سطر، نویسه توضیح، بصورت پیش‌فرض %، وجود داشته باشد.

تک نویسه‌هایی را که از فایل ورودی می‌خواند، به کدهای نویسه خودش تبدیل می‌دهد. تک بعداً تنها از این کدها استفاده می‌کند و به همین خاطر تک روی تمام سیستم عامل‌ها نتیجه یکسانی را تولید می‌کند.

## ۳.۱ کدهای رده

هر کدام از ۲۵۶ کدهای نویسه (۰-۲۵۵) یک کد رده دارند، اگر چه لزوماً همیشه کد رده یکسان نیست. ۱۶ رده با کدهای ۰-۱۵ وجود دارد. در زمان اسکن ورودی، تک جفت‌های کد-نویسه-کد-رده را شکل می‌دهد. پردازشگر ورودی تنها، این جفت‌ها را می‌بیند. نشانه‌های نویسه‌ای، نشانه‌های دستوری، و

نشانه‌های پارامتری، از این جفت‌ها، ساخته می‌شوند. این نشانه‌ها سپس در اختیار پردازشگرهای گسترش و اجرایی قرار می‌گیرند.

یک نشانه نویسه‌ای از یک جفت کد-نویسه-کد-رده تشکیل شده است که دست‌نخورده به مرحله بعدی پردازش سپرده می‌شود. یک نشانه دستوری، از یک نویسه گریز و یک یا چندین نویسه تشکیل شده است.

کدهای رده عبارتند از:

۰. نویسه گریز: این نویسه علامت شروع یک دستور است. در `IniTeX` نویسه `\` (با کد نویسه ۹۲)، نویسه گریز است.

۱. شروع گروه: این نویسه باعث می‌شود تا تک داخل یک سطح جدید گروه‌بندی شود. در فرمت `Plain TeX`، نویسه `{`، نویسه شروع گروه است.

۲. پایان گروه: این نویسه باعث می‌شود تا تک خارج از سطح گروه‌بندی فعلی شود. در فرمت `Plain TeX`، نویسه `}`، نویسه پایان گروه است.

۳. مبدله ریاضی: این نویسه حائل شروع و پایان فرمول‌های ریاضی است. در فرمت `Plain TeX`، نویسه `$`، مبدله ریاضی است.

۴. تب (جهش) ترازبندی: این نویسه جداکننده ستون‌های (ردیف‌های) جدول‌هایی است که توسط دستورات بدوی `\halign` (`\valign`) ساخته شده‌اند. در فرمت `Plain TeX`، نویسه `&`، تب ترازبندی است.

۵. پایان سطر: نویسه‌ای که تک به عنوان علامت پایان یک سطر ورودی در نظر می‌گیرد. در `IniTeX`، نویسه `\return` (با کد نویسه ۱۳)، نویسه پایان سطر است. اتفاقی نیست که `IniTeX`، کد نویسه ۱۳ را همچنین به پارامتر `\endlinechar` اختصاص می‌دهد.

۶. نویسه پارامتر: این نویسه نشان‌دهنده پارامترهای ماکروهاست. در فرمت `Plain TeX`، نویسه `#`، نویسه پارامتر است.

۷. بالانویس: از این نویسه برای حروف چینی توان (نویسه بالانویس قبل از خود توان قرار می‌گیرد)

در محیط ریاضی استفاده می‌شود. همچنین برای وارد کردن کدهای نویسه که نمی‌توانند در فایل ورودی، وارد شوند، استفاده می‌شود. در فرمت Plain TeX، نویسه  $\wedge$ ، نویسه بالانویس است.

۸. زیرنویس: از این نویسه برای حروف چینی زیرنویس (نویسه زیرنویس قبل از خود زیرنویس قرار می‌گیرد) در محیط ریاضی استفاده می‌شود. در فرمت Plain TeX، نویسه  $_$ ، نویسه زیرنویس است.

۹. نادیده پنداشته شده: نویسه‌هایی که به این رده تعلق دارند، از ورودی حذف می‌شوند و بنابراین تأثیری بیشتری روی پردازش تک ندارند. در فرمت Plain TeX، نویسه  $\langle null \rangle$  (با کد نویسه ۰)، نویسه نادیده پنداشته شده است.

۱۰. فاصله: تک توجه خاصی به نویسه‌های فاصله می‌کند. در IniT<sub>E</sub>X، نویسه اسکی  $\langle space \rangle$  (با کد نویسه ۳۲)، یک نویسه فاصله است.

۱۱. حرف: در IniT<sub>E</sub>X، تنها نویسه‌های  $a..z$  و  $A..Z$  به این رده تعلق دارند. معمولاً بسته‌ها، تعدادی از نویسه‌های «پنهان» (به عنوان نمونه @) را تبدیل به یک حرف می‌کنند.

۱۲. غیر: IniT<sub>E</sub>X نویسه‌هایی را که به رده‌های دیگر تعلق ندارند، در این رده قرار می‌دهد. بنابراین، به عنوان نمونه، اعداد و نقطه‌گذاری‌ها در این رده هستند.

۱۳. فعال: نویسه‌های فعال مانند یک دستور تک عمل می‌کنند با این تفاوت که با نویسه گریز شروع نمی‌شوند. در فرمت Plain TeX، تنها نویسه  $\sim$ ، نویسه فعال است، که برای تولید فاصله نشکن، تعریف شده است.

۱۴. نویسه توضیح: زمانی که یک نویسه توضیح در یک سطر ورودی قرار می‌گیرد، هر چیزی که در آن سطر ورودی بعد از نویسه توضیح قرار بگیرد، از نظر تک، توضیح به شمار رفته و هیچ تأثیری در پردازش تک ندارد (تک آن را نادیده می‌گیرد). در IniT<sub>E</sub>X، نویسه  $\%$ ، نویسه توضیح است.

۱۵. نویسه باطل: این رده برای نویسه‌هایی است که نباید در ورودی ظاهر شوند. در IniT<sub>E</sub>X، نویسه اسکی  $\langle delete \rangle$  (با کد نویسه ۱۲۷)، نویسه باطل است.

کدهای رده، کدهای نویسه با دستور زیر قابل تغییر است:

`\catcode` (عدد) (مساوی) (عدد)

عدد اول، معمولاً به شکل زیر داده می‌شود:

(نویسه) \ یا (نویسه) ^

که هر دو، کد نویسه، نویسه را مشخص می‌کنند.

در فرمت Plain TeX، دستور `\active` بصورت زیر تعریف شده است:

```
\chardef\active=13
```

بنابراین، برای تغییر کد رده نویسه { به یک نویسه فعال، می‌توان نوشت:

```
\catcode`\{=\active
```

فرمت LaTeX، دستورات `\makeatletter` و `\makeatother` را به شکل زیر تعریف می‌کند:

```
\def\makeatletter{\catcode\@=11 }
```

```
\def\makeatother{\catcode\@=12 }
```

بنابراین دستور `\makeatletter`، نویسه «پنهان» @ را به یک حرف تبدیل می‌کند، و دستور

`\makeatother` دوباره آن را به نویسه غیر تبدیل می‌کند.

دستور `\catcode` همچنین برای پرس و جو کدهای رده نیز کاربرد دارد. مثال زیر:

```
\count255=\catcode`\{
```

یک عدد تولید می‌کند که واگذاری پذیر است.

برای آزمایش کدهای رده، می‌توان از دستور زیر استفاده کرد:

(نشانه<sub>۲</sub>) (نشانه<sub>۱</sub>) `\ifcat`

تک هر چیزی را که بعد از `\ifcat` باشد، گسترش می‌دهد تا دو نشانه گسترش نیافتنی پیدا شوند.

سپس کدهای رده این دو نشانه با هم مقایسه می‌شوند. نشانه‌های دستوری کد رده ۱۶ دارند. بنابراین تمام

نشانه‌های دستوری با یکدیگر برابر و با نشانه‌های نویسه‌ای نابرابر هستند.

## ۴.۱ از نویسه‌ها تا نشانه‌ها

پردازشگر ورودی تک، سطرهای ورودی را از یک فایل یا از ترمینال کاربر، اسکن می‌کند و نویسه‌ها را در ورودی، به نشانه‌ها تبدیل می‌کند. سه نوع نشانه وجود دارد:

- نشانه‌های نویسه‌ای: هر نویسه‌ای که خودش به تنهایی با یک کد رده مناسب، به مراحل دیگر پردازش تک سپرده می‌شود.
- نشانه‌های دستوری که به دو دسته تقسیم می‌شوند:

– دستور حرفی: این دستور از یک نویسه گریز (یک نویسه با کد رده °) و یک رشته از «حروف» تشکیل شده است. دستور حرفی یک نشانه واحد است.

– دستور نمادی: این دستور از یک نویسه گریز و تنها یک نویسه که حرف نیست (کد رده اش ۱۱ نیست) تشکیل شده است.

دستور نمادی که از یک نویسه گریز و یک نویسه فاصله (\) تشکیل شده است، یک دستور فاصله خوانده می‌شود.

اگر تشخیص بین دستور حرفی و دستور نمادی، غیرضروری باشد، هر دو فقط دستور خوانده می‌شوند.

- نشانه‌های پارامتری: یک نویسه پارامتر (یک نویسه با کد رده ۶، بصورت پیش فرض #) به اضافه یک عدد بین ۱.۰۹، با یک نشانه پارامتری جایگزین می‌شود. نشانه‌های پارامتری، تنها در ماکروها، می‌توانند باشند. یک نویسه پارامتر ماکرو و نویسه پارامتر ماکرو دیگری (که حتماً لازم نیست کد رده یکسانی داشته باشند) که بعد از هم قرار می‌گیرند، با تنها یک نشانه نویسه‌ای جایگزین می‌شوند. این نشانه، کد رده ۶ (پارامتر ماکرو) و کد نویسه، نویسه پارامتر دومی را دارد. شایع‌ترین نمونه جایگزینی ## با ۶# است. در اینجا، زیرنویس نشانگر کد رده نویسه است.

## ۵.۱ پردازشگر ورودی به عنوان یک ماشین حالات متناهی

پردازشگر ورودی تک، می‌تواند یک ماشین حالات متناهی با سه حالت داخلی در نظر گرفته شود. این ماشین در هر لحظه از زمان، در یکی از این سه حالت است و بعد از تغییر از یک حالت به حالت دیگر،

هیچ حافظه‌ای از حالات قبلی ندارد.

### ۱.۵.۱ حالت ج: سطر جدید

تک در شروع هر سطر جدید ورودی، وارد حالت ج می‌شود و این تنها زمانی است که تک در این حالت است. در حالت ج، تمام نشانه‌های فاصله (نویسه‌های با کد رده ۱۰) نادیده گرفته می‌شوند و نویسه پایان سطر، به یک نشانه \par تبدیل می‌شود. تمام نشانه‌های دیگر باعث می‌شوند تا تک وارد حالت م شود.

### ۲.۵.۱ حالت ق: قلم‌انداختن فاصله‌ها

تک، در هر حالتی که باشد، بعد از یک دستور حرفی یا یک دستور فاصله (ولی نه بعد از دستورهای نمادی دیگر)، یا، زمانی که در حالت م است، بعد از یک فاصله، وارد حالت ق می‌شود. در این حالت، تمام فاصله‌های بعدی یا نویسه‌های پایان سطر در این سطر ورودی، دور انداخته می‌شوند.

### ۳.۵.۱ حالت م: میانه سطر

شایع‌ترین حالت، حالت م، «میانه سطر» است. تک بعد از نویسه‌های با کد رده ۱-۴، ۶-۸، و ۱۱-۱۳، و بعد از دستورهای نمادی غیر از دستور فاصله، وارد این حالت می‌شود. اگر در این حالت یک نویسه پایان سطر مشاهده شود، یک نشانه فاصله ساخته می‌شود.

## ۶.۱ دسترسی به مجموعه کامل نویسه‌ها

اگر بخواهیم خیلی سخت‌گیر باشیم، پردازشگر ورودی تک یک ماشین حالات متناهی نیست. به این خاطر که در زمان اسکن سطر ورودی، تمام سه‌تایی‌هایی که از دو نویسه بالانویس یکسان (با کد رده ۷) و یک نویسه بعدی (با کد نویسه کوچکتر از ۱۲۸) تشکیل شده‌اند، با یک نویسه تنها با کد نویسه بین ۰-۱۲۷ که با کد نویسه اصلی به اندازه ۶۴ تفاوت دارد، جایگزین می‌شوند.

از این مکانیزم، به عنوان نمونه، می‌توان برای دسترسی به مکان نویسه‌هایی در یک قلم، که نمی‌توانند وارد شوند، مثلاً به این خاطر که این نویسه‌ها، نویسه‌های کنترلی اسکی هستند، استفاده کرد. بارزترین نمونه‌ها، نویسه‌های اسکی <return> و <delete> هستند. مکان این نویسه‌ها در یک قلم ۱۳ و ۱۲۷

است که بصورت  $M^{\sim}$  و  $^{\sim}?$  قابل دسترسی هستند. هرچند از آنجایی که کد رده  $^{\sim}?$  ۱۵ است (نویسه باطل)، کد رده  $^{\sim}?$  باید قبل از اینکه نویسه ۱۲۷ قابل دسترسی باشد، تغییر کند.

در نسخه ۳ تک، این مکانیزم مقداری تغییر و گسترش پیدا کرده است تا بتوان به ۲۵۶ نویسه دسترسی پیدا کرد. در مکانیزم جدید، هر چهارتایی  $^{\sim}xy$  که هر دو  $x$  و  $y$  اعداد شانزده‌شانزده‌ی (دستگاه اعداد پایه ۱۶) کوچک ( $9-0$  و  $f-a$ ) هستند، با یک نویسه بین  $0-255$  (نویسه رقمی که در دستگاه اعداد پایه ۱۶ بصورت  $xy$  نمایش داده شده است)، جایگزین می‌شود. مکانیزم جدید، یک محدودیت کوچک برای کاربرد مکانیزم قبلی ایجاد می‌کند. اگر، به عنوان نمونه،  $a^{\sim}$  برای تولید نویسه ۳۳، تایپ شود، تک به درستی متوجه یکی از  $9-0$  و  $f-a$  بعدی نمی‌شود.

درست است که این روند، پردازشگر ورودی تک رو تا حدی قدرتمندتر از یک ماشین حالات متناهی واقعی می‌کند، اما در عین حال با بقیه روند اسکن کردن تک، رابطه‌ای ندارد. بنابراین از نظر مفهومی ساده‌تر است که وانمود کرد که این جایگزینی سه‌تایی یا چهارتایی نویسه‌ها که با  $^{\sim}$  شروع می‌شود، پیشاپیش انجام شده است. در عمل، این موضوع ممکن نیست به این خاطر که یک سطر ورودی ممکن است به نویسه‌هایی غیر از  $^{\sim}$ ، کد رده ۷ بدهد که این موضوع پردازش بیشتر این سطر ورودی را تحت تأثیر قرار می‌دهد.

## ۷.۱ تغییر از یک حالت داخلی به حالت داخلی دیگر



## کتاب نامه



## تاریخچه بازی‌های



## مجوز کتاب