

روشهای افزایش سرعت اجرای کد متلب

ویرایش اول، آذرماه ۱۳۹۶

در این راهنما، به تعدادی از راهکارهای نرم‌افزاری به منظور افزایش سرعت اجرای کد متلب (MATLAB) اشاره می‌شود. اعمال این موارد، در کنار راهکارهای سخت‌افزاری نظیر افزایش مقدار RAM، افزایش فرکانس کلاک CPU، پردازش موازی با استفاده از پردازنده‌های چند هسته‌ای، همواره مد نظر کاربران می‌باشد. این راهنما متمرکز بر راهکارهای کد نویسی است. شایان ذکر است که قرار دادن دستور tic در اول کد (یا بخشی از کد) و دستور toc در آخر آن، زمان اجرای برنامه مذکور در رایانه مربوطه را مشخص می‌کند (بطور نمونه در برنامه ۱). به علاوه استفاده از ابزارک Analyze Code و Profiler جهت بررسی عملکرد و گلوگاه‌های زمانی برنامه توصیه می‌شود. راهکارهای نرم‌افزاری ذیل برای افزایش سرعت اجرای کد متلب از تارنمای www.mathworks.com گردآوری و تنظیم شده است که در این راهنما برای هر یک نیز برنامه‌ای نوشته شده است. بدیهی است برای کسب اطلاعات تکمیلی با ذکر جزئیات می‌توان از تارنمای مذکور و help متلب استفاده نمود.

- **پیش‌تعریف ابعاد ماتریس:** در ابتدای برنامه، ابعاد هر بردار و ماتریسی که قرار است در برنامه استفاده شود، با دستور zeros یا ones تعریف شود. بطور نمونه در برنامه ۱ اگر پیش‌تعریف ابعاد بردار Y و k انجام نشود (با حذف دو دستور zeros)، مدت اجرا در رایانه نگارنده، حدود ۷/۸ ثانیه و چنانچه پیش‌تعریف انجام شود، مدت اجرا حدود ۱ ثانیه است (با تعریف اولیه ابعاد متغیر، حدود ۷/۸ برابر سرعت اجرا افزایش یافت).

برنامه ۱

```
clear all
tic
h=10^-3;
y(1)=0;
x=1;
y=zeros(1,(10^7));
k=zeros(1,(10^7));
for i=2:10^7
    y(i)=y(i-1)+h*(x-y(i-1));
    k(1,i-1)=y(i)^2+sin(rand^2);
end
toc
```

- **حداقل استفاده از حلقه:** معمولاً زمان‌برترین قسمت برنامه‌های متلب، حلقه‌ها هستند. استفاده حداقل از حلقه‌ها یا استفاده از دستورات جایگزین بجای آن‌ها، باعث افزایش سرعت اجرا می‌شود. به عنوان نمونه در برنامه ۳ بجای استفاده از حلقه for که در برنامه ۲ استفاده شده است، می‌توان از دستور sum یا mean استفاده نمود.

برنامه ۲

```
clear all
x=rand(1,10^7);
y=rand(1,10^7);
X=0;
Y=0;
for i=1:10^7
    X=x(i)+X;
    Y=y(i)+Y;
end
z=(X/10^7)^2+(Y/10^7)^2;
```

- **پردازش موازی:** توجه شود که در حالت عادی نرم‌افزار متلب فقط از یک هسته رایانه استفاده می‌نماید. با استفاده از دستور matlabpool (در نسخ ۲۰۱۷-۲۰۱۵ متلب، دستور parpool) می‌توان از همه هسته‌های رایانه، به منظور انجام پردازش موازی و نهایتاً افزایش سرعت اجرای برنامه استفاده کرد. در این حالت، بجای دستور for از دستور parfor استفاده می‌شود. به عنوان نمونه، مدت زمان اجرای برنامه ۴ بدون تعریف پردازش موازی، حدود ۱۱ ثانیه و مدت اجرای همین برنامه با استفاده از چهار هسته (برنامه ۵) حدود ۴/۸ ثانیه است.

برنامه ۳

```
clear all
x=rand(1,10^7);
y=rand(1,10^7);
z=mean(x)^2+mean(y)^2;
```

- **ارتباط با سایر نرم‌افزارها:** با استفاده از دستور MEX در متلب، می‌توان کدهای نوشته شده به زبان C یا Fortran را فراخوانی نمود. در اینصورت سرعت برنامه، به این دلیل که کد مستقیماً در زبان کامپایلر نوشته می‌شود، افزایش می‌یابد.

برنامه ۴

```
clear all
S=zeros(8,1000);
for k = 1:8
    S(k,:) = eig(rand(1000));
end
```

- **گذشتن semicolon در انتهای هر خط:** عدم نمایش محاسبات برنامه در حین اجرا، سبب افزایش سرعت می‌شود. همچنین دستورات نمایش گرافیکی نظیر plot، mesh و surf در داخل حلقه به شدت سبب کاهش سرعت می‌شود.

برنامه ۵

```
clear all
S=zeros(8,1000);
matlabpool open local 4
% parpool
parfor k = 1:8
    S(k,:) = eig(rand(1000));
end
matlabpool close
```

- **دستور شرطی:** استفاده از دستور if-elseif-end بجای نوشتن چند دستور متوالی if-end به افزایش سرعت اجرا کمک می‌نماید. بدیهی است در صورتی که تعداد شروط در برنامه زیاد باشد، افزایش سرعت با این تکنیک، محسوس خواهد بود. در صورتی که تبحر کافی در این خصوص وجود نداشته باشد، برای اجتناب از اشتباه توصیه می‌شود، از همان if-end های متوالی استفاده شود.

- **بالا بردن اولویت برنامه متلب:** در بخش Task Manager ویندوز از گزینه با اهمیت (High) برای اولویت (Priority) دادن به نرم‌افزار متلب استفاده شود. به علاوه، استفاده از ویندوز ۶۴ بیتی نیز سبب افزایش سرعت خواهد شد.

- **عدم ذخیره اطلاعات غیر ضروری:** از ذخیره کردن اطلاعات اضافی و غیر ضروری (مخصوصاً در حلقه‌ها و بخش‌های تکراری) اجتناب شود.

گردآوری و تنظیم: وحید بَهلوری