



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

دانشکده مهندسی کامپیوتر و فناوری اطلاعات
آزمایشگاه OS
گزارش مسئله فیلسوف‌های غذاخور

نگارش
آرش حاجی صفی - 9631019

آذر 1398

یک آرایه از `threading.Lock()` با اندازه‌ی 5 گرفته ام به اسم `ph`. هر فیلسوف یک قفل متناظر با خودش در این آرایه دارد؛ قفل مربوط به فیلسوف `i` ام `ph[i-1]` می‌باشد (چون ایندکس‌ها از 0 شروع می‌شوند).

یک آرایه از وضعیت‌ها گرفته ام به اسم `state` که وضعیت فیلسوف `i` ام `state[i-1]` می‌باشد. مقدار اولیه وضعیت همه‌ی فیلسوف‌ها `thinking` است و وضعیت همیشه بین (`"thinking"`, `"hungry"`, `"eating"`) می‌باشد.

فیلسوف `(i-1)%5` سمت چپ فیلسوف `i` ام و فیلسوف `(i+1)%5` فیلسوف راستی فیلسوف `i` ام هستند.

هر فیلسوفی که گرسنه شد، تلاش می‌کند که چنگال سمت چپ و راست خود را بردارد (با متد `takeForks`)، این در صورتی ممکن است که فیلسوف سمت چپ و فیلسوف سمت راست مشغول خوردن نباشند، این وضعیت با متد `test` سنجیده می‌شود. وقتی این وضعیت برقرار بود و وضعیت فیلسوف به `eating` تغییر کرد، چنگال‌ها را به هر ترتیبی که بردارد امکان پذیر است.

وقتی خوردن فیلسوف تمام شد، چنگال‌ها را روی میز می‌گذارد و به فیلسوف سمت چپ و راست خود اطلاع می‌دهد. یعنی متد `test` را برای فیلسوف سمت چپ و راست فرا می‌خواند و در صورتی که شرایط خوردن آنها برقرار باشد و گرسنه باشند، چنگال‌ها را بر می‌دارند و مشغول خوردن می‌شوند.

روال‌های چک کردن وضعیت‌ها و ست کردن وضعیت‌ها به عنوان نواحی بحرانی با قفل با نام `lock` در کد محافظت شده اند.

این روش حل این مسئله هیچ مشکل و `deadlock` و بن بست را ایجاد نمی‌کند و حل مشکل `starvation` را تضمین می‌کند.