**مسئله غذاخوری فیلسوف‌ها**

**مقدمه**

مسئله غذاخوری فیلسوف‌ها یکی از مسائل کلاسیک در زمینه همروندی (Concurrency) است که برای بررسی مسائل بن‌بست، گرسنگی (Starvation) و رقابت منابع مطرح شده است. این مسئله شامل تعدادی فیلسوف است که به دور یک میز نشسته و می‌خواهند به طور متناوب غذا بخورند و فکر کنند. هر فیلسوف نیاز دارد برای غذا خوردن از دو چنگال استفاده کند. با توجه به اینکه تعداد چنگال‌ها محدود و برابر با تعداد فیلسوفان است، چالش اصلی مدیریت دسترسی به این چنگال‌هاست.

**هدف پروژه**

هدف این پروژه، طراحی و پیاده‌سازی یک شبیه‌ساز برای مسئله غذاخوری فیلسوف‌ها با استفاده از ساختمان داده‌هایی نظیر صف، آرایه و متغیرهای همروندی است. این شبیه‌ساز باید:

1. از بن‌بست جلوگیری کند.
2. گرسنگی هیچ فیلسوفی را ایجاد نکند.
3. رفتار سیستمی پایدار ارائه دهد.

**شرح مسئله**

1. **ساختار مسئله:**
   * تعداد فیلسوفان به صورت ورودی تعیین می‌شود.
   * هر فیلسوف می‌تواند در یکی از حالات زیر باشد:
     + **فکر کردن**
     + **منتظر ماندن برای چنگال‌ها**
     + **خوردن غذا**
2. **شرایط:**
   * هر فیلسوف برای غذا خوردن به دو چنگال نیاز دارد: یکی در سمت چپ و دیگری در سمت راست.
   * اگر همه فیلسوفان همزمان بخواهند غذا بخورند، ممکن است سیستم دچار بن‌بست شود.
3. **چالش‌ها:**
   * جلوگیری از بن‌بست با استفاده از الگوریتم مناسب.
   * تضمین عدالت در دسترسی به منابع برای جلوگیری از گرسنگی فیلسوفان.

**ابزارها و ساختمان داده‌ها**

* **آرایه:** برای نگهداری وضعیت فیلسوف‌ها (فکر کردن، منتظر بودن، خوردن)
* **صف:** برای مدیریت درخواست‌های دسترسی به چنگال‌ها.
* **Semaphore یا Mutex**  برای همگام‌سازی دسترسی به چنگال‌ها.

**ویژگی‌های پیاده‌سازی**

1. **ورودی:**
   * تعداد فیلسوفان
   * زمان مورد نیاز هر فیلسوف برای خوردن و فکر کردن
2. **خروجی:**
   * نمایش وضعیت لحظه‌ای فیلسوف‌ها (فکر کردن، منتظر بودن، خوردن)
   * گزارش نهایی شامل تعداد دفعاتی که هر فیلسوف غذا خورده است.
3. **عملکردها:**
   * مدیریت تخصیص چنگال‌ها به فیلسوفان.
   * نمایش وضعیت سیستم به صورت بلادرنگ.
4. **الگوریتم‌های پیشنهادی:**
   * استفاده از الگوریتم‌های همروندی مانند Monitor.
   * استفاده از شماره‌گذاری فیلسوفان برای مدیریت دسترسی (فیلسوف شماره فرد ابتدا چنگال سمت راست و سپس چنگال سمت چپ را برمی‌دارد و برعکس)

**پیاده‌سازی پروژه**

1. **ساختار برنامه:**
   * تعریف یک کلاس برای فیلسوفان.
   * تعریف ساختمان داده‌ای برای مدیریت چنگال‌ها.
   * پیاده‌سازی همروندی با استفاده از thread و mutex در ++C
2. **توابع اصلی:**
   * think(): شبیه‌سازی فکر کردن.
   * eat(): شبیه‌سازی غذا خوردن.
   * request\_forks(): درخواست چنگال‌ها.
   * release\_forks(): آزادسازی چنگال‌ها.
3. **نمایش وضعیت:**
   * هرt واحد زمان، وضعیت فعلی فیلسوفان و چنگال‌ها نمایش داده شود.

**نکات فنی**

* استفاده از mutex برای جلوگیری از دسترسی همزمان چند فیلسوف به چنگال‌های مشترک.
* مدیریت صف درخواست‌ها با استفاده از صف اولویت‌دار برای تضمین عدالت.
* مدیریت حالت‌های فیلسوفان با استفاده از آرایه.

**ویژگی‌های اضافی برای کسب امتیاز بیشتر**

1. امکان شبیه‌سازی زمان واقعی با وارد کردن زمان فکر کردن و غذا خوردن.
2. نمایش گرافیکی ساده از وضعیت فیلسوفان و چنگال‌ها.
3. پشتیبانی از تعداد فیلسوفان دینامیک و قابلیت تغییر در طول اجرا.
4. ارائه گزارش‌های آماری شامل میانگین زمان انتظار هر فیلسوف.

**نتیجه‌گیری**

این پروژه می‌تواند به دانشجویان کمک کند تا مفاهیم همروندی، مدیریت منابع مشترک و الگوریتم‌های جلوگیری از بن‌بست را درک کنند. شبیه‌سازی دقیق مسئله غذاخوری فیلسوف‌ها همچنین یک چالش مناسب برای تقویت مهارت‌های برنامه‌نویسی و حل مسئله است.