

به نام خدا  
دانشگاه اصفهان دانشکده مهندسی کامپیوتر



# تمرین اول

افزونه یدکی

معماری نرم افزار	نام درس
دکتر محمدرضا شهرباف	استاد درس
محمد خورسندی	طراح

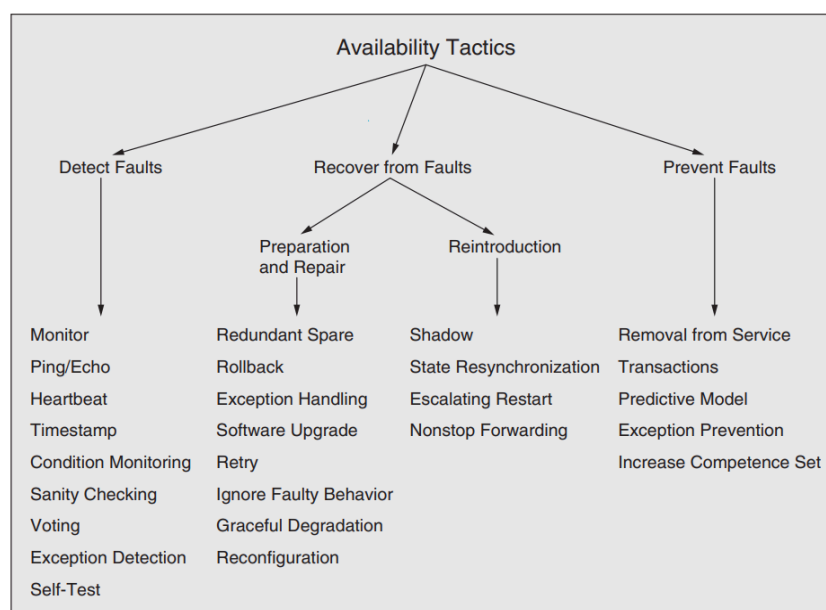
## فهرست مطالب

2.....	فهرست مطالب
3.....	مقدمه
4.....	۱- ساختار اولیه پروژه
4.....	۲- یدک داغ
5.....	۳- یدک گرم
5.....	۴- یدک سرد
5.....	۵- نکات مهم برای پیاده سازی
5.....	۶- آنچه باید تحویل داده شود

## مقدمه

دسترس‌پذیری<sup>1</sup> یکی از ویژگی‌های کلیدی در معماری نرم‌افزار است که بیان می‌کند یک سیستم تا چه حد در زمان نیاز، در دسترس و قابل استفاده است. در بسیاری از سیستم‌های حیاتی مانند خدمات بانکی، فروشگاه‌های آنلاین، سامانه‌های اورژانس و ...، حتی چند ثانیه قطعی می‌تواند خسارت‌های مالی یا اعتباری جدی به همراه داشته باشد. برای افزایش دسترس‌پذیری، سیستم باید بتواند در برابر خرابی‌ها مقاومت کند یا به سرعت بازیابی شود.

به طور کلی سه دسته تاکتیک برای افزایش دسترس‌پذیری نرم‌افزار وجود دارد: جلوگیری از خطا، شناسایی خطا و بازیابی از خطا. در شکل زیر می‌توانید این سه دسته را مشاهده کنید:



یکی از تاکتیک‌هایی که برای بازیابی از خطا استفاده می‌شود، **افزونه یدکی**<sup>2</sup> است که در صورت خرابی مؤلفه‌ی اصلی، جایگزین آن می‌شود و عملکرد سیستم را حفظ کنند. سه الگو معماری زیر بر این تاکتیک بنا شده‌اند:

- **یدک داغ**<sup>3</sup> مؤلفه‌ی پشتیبان به صورت همزمان با مؤلفه‌ی اصلی در حال اجرا و به روز است و در صورت خرابی، بلافاصله می‌تواند جایگزین شود.
- **یدک گرم**<sup>4</sup> مؤلفه‌ی پشتیبان فعال است اما داده‌های آن ممکن است کمی قدیمی باشد؛ برای جایگزینی نیاز به همگام‌سازی سریع دارد.

<sup>1</sup> Availability

<sup>2</sup> Redundant Spare

<sup>3</sup> Hot Spare

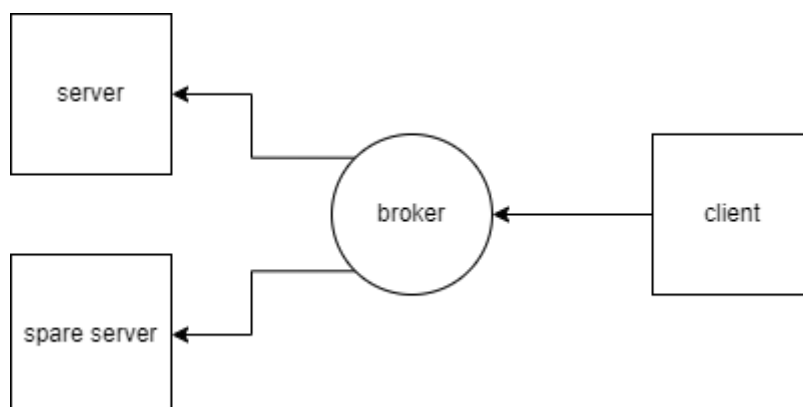
<sup>4</sup> Warm Spare

- **یدک سرد<sup>5</sup>** مؤلفه‌ی پشتیبان غیرفعال است و تنها در زمان خرابی مؤلفه‌ی اصلی راه‌اندازی می‌شود و ممکن است زمان‌بر باشد.

در این تمرین، هدف شما پیاده‌سازی و تحلیل این سه الگوهای به‌عنوان راهکارهایی برای افزایش دسترس‌پذیری سیستم است.

## ۱- ساختار اولیه پروژه

پروژه اولیه در [این لینک](#) موجود است. در این پروژه، یک سیستم ساده شامل مؤلفه‌های زیر طراحی شده است:



- دو سرور با نام‌های server و spare که هر دو یکسان و پیاده‌سازی یک سیستم ذخیره و بازیابی کتاب هستند.
- یک بروکر<sup>6</sup> که درخواست‌های کاربر را دریافت می‌کند و به یکی از سرورها می‌فرستد. به علاوه این بروکر قطعی سرورها را تشخیص می‌دهد و در صورت آماده بودن سرور دیگر به آن سوئیچ می‌کند.
- رابط کاربری شامل دو صفحه است.
- صفحه‌ی ایجاد، حذف و مشاهده کتاب‌ها (/)
- با استفاده از این صفحه می‌توان بررسی کرد داده‌ها بعد از تغییر سرور حفظ شده و به درستی بازیابی شوند.
- صفحه‌ی مانیتورینگ سرور برای شبیه‌سازی خطا (/monitor)
- با استفاده از این صفحه می‌توان یک خطا در سیستم شبیه‌سازی کرد به طوری که سرور از دسترس خارج شود.
- بعد از کلیک، یک تایمر شروع می‌شود و وقتی یک سرور جدید بالا آمد، پایان می‌یابد. از این تایمر برای تحلیل دسترس‌پذیری استفاده کنید.

<sup>5</sup> Cold Spare

<sup>6</sup> Broker

شما باید روی همین پروژه، سه الگوی ذکر شده را پیاده‌سازی کنید.

## ۲- یدک داغ

- در این حالت، هر دو سرور هم‌زمان و کاملاً فعال هستند.
- بروکر باید در لحظه‌ای که یکی از سرورها از کار می‌افتد، به سرور دیگر سوییچ کند.
- در این مدل، هر تغییر (مثلاً افزودن/حذف کتاب) باید هم‌زمان روی هر دو سرور اعمال شود.

## ۳- یدک گرم

- در این مدل، تنها سرور اصلی فعال است.
- سرور یدک روشن و آماده است ولی مستقیماً سرویس نمی‌دهد.
- داده‌ها در بازه‌های زمانی مشخص با سرور اصلی همگام‌سازی می‌شوند (ترجیحاً با استفاده از API)
- وقتی سرور اصلی از کار می‌افتد، پراکسی باید به سرور پشتیبان سوییچ کند.

## ۴- یدک سرد

- فقط سرور اصلی فعال است.
- زمانی که سرور اصلی خراب شود، سرور پشتیبان به صورت خودکار روشن می‌شود.
- برای بازیابی داده‌ها می‌توانید از ذخیره لاگ‌ها و یا پایگاه داده سرور اصلی استفاده کنید.
- بروکر باید متوجه شود که سرور اصلی خاموش شده است، و وقتی سرور پشتیبان آماده شد، به آن سوییچ کند.

## ۵- نکات مهم برای پیاده‌سازی

- از ping که در سرورها پیاده‌سازی شده می‌توانید برای بررسی سلامت سرورها استفاده کنید. هر چند مکانیزم بررسی سرورها در بروکر پیاده‌سازی شده، به طوری که در صورت متصل بودن هر یک سرورها بروکر به طور خودکار متوجه شده و به آن متصل می‌شود. در صورت نیاز می‌توانید این مکانیزم را در جهت بهبود کارایی تغییر دهید.
- برای بروکر از پورت ۸۰۰۰، برای سرور اصلی ۸۰۰۱ و برای سرور یدکی از ۸۰۰۲ استفاده کنید.
- نحوه راه اندازی پروژه در فایل README در مخزن داده شده قرار داده شده است.
- توجه کنید که همه درخواست‌ها باید به بروکر ارسال شوند و رابط کاربری نباید مستقیماً به سرورها درخواست ارسال کند.

## ۶- آنچه باید تحویل داده شود

- برای هر کدام از الگوهای بالا یک برنچ در گیت (مجموعاً سه برنچ) بسازید و پیاده سازی مربوط به هر الگو را در برنچ مربوطه انجام دهید.
- یک گزارش مختصر از پیاده سازی هر یک از الگوها ارائه دهید. گزارش بایستی شامل تصمیمات کلیدی و تاثیر گذار بر کارایی مانند نحوه همگام سازی داده ها و جایگزینی سرور باشد. از ذکر جزئیات در گزارش پرهیز کنید.
- با استفاده از صفحه پایش (monitor) موجود در رابط کاربری میانگین زمان بازیابی هر حالت را محاسبه و سپس دسترسی پذیری هر حالت را با استفاده از فرمول زیر محاسبه کنید.

$$\frac{MTBF}{MTBF+MTTR}$$

$MTBF$  زمان میانگین بین دو خرابی و  $MTTR$  زمان میانگین مورد نیاز برای بازیابی سیستم است. فرض کنید سیستم به طور میانگین یک بار در روز خرابی دارد.

