

تاب‌آوری خطا

تحمل‌پذیری اشکال یا عیب‌تابی^[۱] (به انگلیسی: Fault tolerance) ویژگی ای است که سامانه را قادر می‌سازد تا با به وجود آمدن اشکال (یک یا چندین اشکال) در اجزایش، به صورت صحیح به کار خود ادامه دهد. در این دسته از سامانه‌ها در صورت بروز اشکال، در صورتی که کیفیت عملکرد کلی سامانه کاهش یابد، میزان این کاهش متناسب با میزان بزرگی شدت اشکال است، در حالی که در طراحی‌های ساده‌تر، این اشکال‌ها می‌توانند به‌طور کلی موجب از کار افتادن سامانه شوند. توجه به موضوع تحمل‌پذیری اشکال بعد از پیدایش سامانه‌های همیشه روشن و ایمن بحران اهمیت ویژه‌ای یافت. از ویژگی ادامه کارکرد سامانه‌ها با وجود کارکرد ناصحیح بعضی از بخش‌های آن‌ها، تحت عنوان شکست امن یاد می‌شود.^[۲]

دلایل

دلایل نیاز به تحمل خطا^[۳]

1. اتکا پذیری: کارکرد بلادرنگ سیستم بدون هیچ وقفه‌ای
2. دسترسی: دسترسی به اطلاعات در هر زمان
3. امنیت: جلوگیری از دسترسی‌های غیرمجاز

دلایل به وجود آمدن خطا

1. خطاهای نرم‌افزاری یا سخت‌افزاری
2. دسترسی غیرمجاز

طراحی مقاوم در برابر خطا

طراحی مقاوم در برابر خطا به طراحی ای گفته می‌شود که به سامانه این امکان را می‌دهد در مواجهه با خطا در بعضی از بخش‌هایش، کارکرد خود را در همان سطح یا در سطحی پایین‌تر ادامه دهد و از کار افتادن کلی سامانه جلوگیری کند.^[۴] این طراحی در بحث‌های مربوط به معماری رایانه به‌طور معمول به این برمیگردد که در صورت کاهش عملکرد رایانه، با کاهش توان عملیاتی یا افزایش زمان پاسخ‌دهی به کار خود ادامه دهد. این طراحی در هر دو حوزه سخت‌افزار و نرم‌افزار وجود دارد.

روش‌های تحمل‌پذیری اشکال

به‌طور عمومی تحمل اشکال می‌تواند با دو روش عمومی حاصل شود:

- خود تثبیت

- نسخه برداری

خود تثبیت

در این روش طراحان شرایطی را که ممکن است سامانه دچار خطا شود را تشخیص می‌دهند و سامانه را به گونه‌ای طراحی می‌کنند که از پس این شرایط بتواند برآید. در نهایت هدف اصلی طراحان در این روش رسیدن به یک سامانه خود تثبیت است تا هیچوقت دچار خطا نشود.

معایب: مشکلات اصلی این روش وقتی است که سلسله‌ای از خطاهای حیاتی در سامانه رخ دهد یا هزینه سامانه با ضریب اطمینان بالا خیلی زیاد شود.

نسخه برداری

در این روش سامانه به صورتی طراحی می‌شود که در صورت بروز هر گونه خطای حیاتی بتواند به حالت ایمن قبلی بازگردد. راهبرد این روش به این گونه است که از حالت قبلی خود نسخه دخیره دارد و در صورت لزوم می‌تواند آن را بازیابی کند. این روش معایب کمتری نسبت به روش خودتثبیت دارد.

روش‌های تحمل‌پذیری اشکال در نرم‌افزار



درخت رویکرد نرم‌افزار در رابطه با تحمل‌پذیری اشکال.

روش‌های تحمل اشکال در حوزه نرم‌افزار به سه دسته کلی تقسیم می‌شود: [۵]

- جلوگیری یا اجتناب از خطا
- حذف خطا
- پیش‌بینی خطا
- تحمل اشکال

جلوگیری یا اجتناب از خطا

این روش رویکردی شبه خودتشبیه دارد و نرم‌افزاری بدون خطا طراحی می‌شود.

حذف خطا

این رویکرد به این صورت است که بعد از طراحی نرم‌افزار، آزمون‌های زیادی بر روی آن انجام می‌شود تا نرم‌افزار نهایی بدون هرگونه خطا شود.

پیش‌بینی خطا

پیش‌بینی خطاهای احتمالی در نرم‌افزار و شرایط رخدادشان و خطرهای احتمالی ناشی از آنها.

تحمل خطا

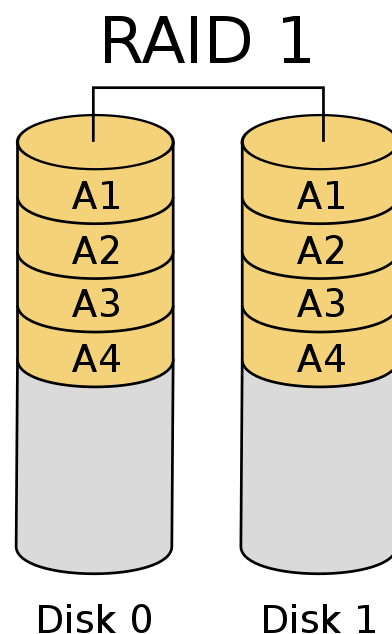
در این روش طراحان با این تفکر که خطا وجود دارد سامانه را به گونه‌ای که بتواند در این شرایط هم عملکرد صحیح داشته باشد، طراحی می‌کنند.

رویکرد تحمل‌پذیری اشکال

در زمان رخداد خطا دو راه حل وجود دارد:^[6]

- درمان خطا: تلاش می‌کند که از بروز خطاها ی بعدی جلوگیری کند و قبل از خرابکاری و رسیدن به شرایط بحرانی، به اتمام برسد.
- پردازش خطا: در این رویکرد سامانه دو راه دارد، یا به دنبال پاک کردن خطا است که **بازیابی خطا** نام دارد یا با استفاده از نسخه برداری این کار را انجام می‌دهد که **جبران خطا** نام دارد.

روش‌های تحمل‌پذیری اشکال در دیسک‌ها



RAID 1 نسخه ی دوم از هر داده را بر روی دیسک کمکی ذخیره می‌کند.

اصلی‌ترین روش در این حوزه به کارگیری **آرایه چندگانه دیسک‌های مستقل** (به انگلیسی: redundant array of independent disks) یا **RAID** است.

از اصلی‌ترین اهداف این فناوری:

1. افزایش توان کارکرد از راه تکنیکی به نام **data striping** (خواندن و نوشتن داده‌ها بر روی چند دیسک به صورت موازی)

2. افزایش توان تحمل در برابر اشکال (**fault-tolerance**) از راه نسخه برداری و **تصحیح گره‌های داده‌ها (Error correcting)**.

منابع

1. «سامانه عیب‌تاب» [رایانه و فناوری اطلاعات] هم‌ارز «**fault-tolerant system**»؛ منبع: گروه واژه‌گزینی (<https://a.dll.ir/%D9%88%D8%A7%DA%98%D9%87%E2%80%8C%DA%AF%D8%B2%DB%8C%D9%86%D/B%8C/>). جواد میرشکاری، ویراستار. دفتر پنجم. فرهنگ واژه‌های مصوب فرهنگستان. تهران: انتشارات فرهنگستان زبان و ادب فارسی. شابک ۹۷۸-۹۶۴-۷۵۳۱-۷۶-۴ (ذیل سرواژه سامانه عیب‌تاب)
2. Oscar González (1997, University of Massachusetts - Amherst). «**Adaptive Fault Tolerance and Graceful Degradation**» (http://scholarworks.umass.edu/cgi/viewcontent.cgi?article=1186&context=cs_faculty_pubs). تاریخ وارد شده در تاریخ = را بررسی کنید (کمک)
3. Sumit jain (jul 2, 2014). «**Fault tolerance in distributed systems**» (<https://www.slideshare.net/sumitjain2013/fault-tolerance-in-distributed-systems>). تاریخ وارد شده در تاریخ = را بررسی کنید (کمک)
4. "Johnson, B. W. "Fault-Tolerant Microprocessor-Based Systems" IEEE Micro, vol. 4, no. 6, pp. 6-21. 1984 (به انگلیسی).
5. Laura L. Pullum "Software Fault Tolerance Techniques and Implementation". Artech House. ۲۰۰۱.
6. «Techniques for Fault Tolerance in Software» (https://web.archive.org/web/20170508174550/http://srel.ee.duke.edu/sw_ft/n). بایگانی‌شده از اصلی (http://srel.ee.duke.edu/sw_ft/node5.html) در ۸ مه ۲۰۱۷. دریافت‌شده در ۲۵-۰۴-۲۰۱۷.

برگرفته از «https://fa.wikipedia.org/w/index.php?title=تاب‌آوری_خطا&oldid=33595590»

آخرين ويرايش ۱۲ ماه پيش توسط HujiBot انجام شده

ويکي پدیا
