

O texto de David Parnas fala sobre como dividir um sistema em módulos e por que isso é tão importante. Modularizar significa quebrar um sistema em partes menores e independentes, chamadas de módulos. Isso traz três grandes vantagens: ajuda no gerenciamento do trabalho, deixa o sistema mais fácil de entender e facilita fazer mudanças sem bagunçar tudo.

No exemplo do artigo, ele usa um sistema chamado KWIC, que gera índices de palavras em contexto. Parnas mostra duas formas de modularizar esse sistema. A primeira é a mais comum: dividir pelas etapas do processo, como entrada de dados, ordenação e saída. Essa abordagem é simples, mas tem um problema sério: se mudar algo no formato dos dados, vários módulos precisam ser

alterados, aumentando o esforço e o risco de erros.

A segunda forma, que Parnas defende, é baseada no princípio de information hiding (ocultamento de informações). Aqui, cada módulo esconde uma decisão de projeto que pode mudar no futuro. Assim, se o jeito de armazenar os dados mudar, só o módulo responsável por isso precisa ser alterado. Os outros continuam funcionando normalmente. Isso deixa o sistema mais flexível, robusto e fácil de manter.

No fim, Parnas conclui que não devemos modularizar só seguindo o fluxo de execução (como em um fluxograma), mas sim pensando nas decisões que podem mudar. Cada módulo deve proteger os outros desses detalhes. Esse conceito,

que parece simples, acabou se tornando uma das bases da engenharia de software moderna, influenciando desde a programação orientada a objetos até boas práticas de arquitetura de sistemas.