6. časť wekových weekendových zamyslení

Na počiatku bol Počítač. A rozumel len Strojovému Jazyku a jeho Assembleru. A Pamäť bola jednoliata a bez zvláštnych vlastností, bez tvaru. A Programátor videl Pamäť a usporiadal ju podľa svojej vôle a predstáv, a označil ju návestiami. Programátor tak vedel použiť Pamäť vo svojom Počítači, a výpisy jeho programov odzrkadľovali mená jednotlivých miest v Pamäti. A to bolo dobre.



Ale Programátora unavilo písanie Programov ako rad príkazov. A On videl, že niektoré časti Programu sa opakujú a dajú sa vydeliť do osobitnej časti kódu. A tak Programátor stvoril Podprogramy. A Programátor si uvedomil, že Podprogramy môžu volať Podprogramy. A pre poriadok vo volaní Podprogramov, On stvoril Zásobník, do ktorého ukladal adresy Návratu, aby sa Podprogramy mohli navrátit na svoje miesto. A keďže On sa stále zaoberal assemblerom, vyčlenil časť Pamäte na odovzdávanie Parametrov Podprogramu, a vyčlenil ďalšiu Pamäť pre potreby Podprogramu a pre hodnoty, na ktorých má pracovať. A to bolo dobre.

Lenže Pamäť zvolala, rieknuc:"Ty si nekonečný, avšak ja som obmedzená.". A Programátora unavilo vyčleňovanie častí pamäte pre odovzdávanie Parametrov každému Podprogramu. A programátora unavilo vyčleňovanie dočasných oblastí Pamäte pre každý Podprogram.

Vtedy Programátor myslel na opätovné použitie Pamäte, ktorá bola potrebná len dočasne v Podprograme. A On videl, že Zásobník je ideálnym miestom pre takú dočasnú pamäť a pre Parametre. A tak Programátor umiestnil Parametre a dočasné premenné na Zásobník. A On nazval dočasné Premenné Lokálnymi, a trvalé Premenné Globálnymi. A oddelil Lokálne Premenné od Globálnych Premenných. A Výsledky Podprogramu boli umiestnené do Globálnych Premenných. A to bolo dobre.

A tak vidiac, že odovzdanie Parametrov Podprogramu je dobré, Programátor sa rozhodol, že Podprogram by mohol vracať Výsledky rovnakým spôsoom. A tak Podprogramy mohli vracať Výsledky v Globálnych Premenných aj na Zásobníku. A Programátor bol preto spokojný.

A všetky programy boli napísané v Assembleri. Ale Programátor bol čoraz viac unavený z písania Assembleru, keď sa Programy stali dlhými a zložitými. A vtedy Programátor stvoril Jazyky. A keďže Jazyky Mu dovolili písať abstraktné Programy, a keďže sa Počítače a ich Pamäte stále zväčšovali, Programátor sa prestal starať o to, ako sú Premenné uložené v Pamäti.

Ale zanedlho sa Programátorove Programy stali dlhými a komplikovanými. A programátor sa začal starať len o Návratové hodnoty z Podprogramov, a zabúdal na to, ktoré Globálne Premenné sú Podprogramom zmenené. A takáto zmena Globálnej Premennej sa nazývala Vedľajší Efekt. A zmeny Globálnych Premenných sa stali rozsiahlymi a nevysledovateľnými. A Vedľajších Efektov pribúdalo na všetkých miestach v programe. A Vedľajšie Efekty spôsobovali mnoho chýb. A bolo z toho mnoho nárekov a škrípania zubov a trhania šiat a vlasov. A Programátorov Manažér ho zbil, volajúc: "Ty naničhodník!"

A tak sa stalo, že Programátor sa rozhodol, že Globálna Premenná a jej Vedľajší Efekt sú Zlo. A On riekol, že nesmú byť Globálne Premenné a že Podprogramy môžu vrátiť výsledky len cez Návratové Parametre.

A tak On vykázal Globálne Premenné aj s ich Vedľajšími Efektami do Pekla, aby o nich už nikdy viac nepočul.

Cenou tohto bolo, že Podprogram mohol zmeniť Premennú len ak mu bola odovzdaná ako Parameter, alebo ju vrátil ako Návratový Parameter. Ale keďže Počítačový Priemysel bol plodný a úrodný, veľkosť Pamäte sa zniekoľkonásobila a použitie Zásobníka na Parametre a Dočasné Premenné bolo vyhlásené za posvätné.

A o spotrebu Pamäte sa Programátor vôbec nestaral, keďže mal nadbytok Pamäte a Jazyky ju rozdeľovali za neho.

Lenže vtom sa utvorila nová podoba Počítača zvaná Mikrokontrolér. A hľa, bol rýchly a lacný a malý, takže Hardverový Inžinier si mohol zvoliť Lacné, Rýchle aj Dobré naraz. Ale Mikrokontroléry boli prekliate a nemohli mať väčšiu než malú pamäť. Ale Hardverový Inžinier sa o to nestaral, lebo znal, že má všetky drôty a vodiče pohromade, a všetko ostatné bol Softverový Problém.

A vyhlásil víťazstvo a bol prehlásený za Hrdinu.

A tak Programátor zahájil Projekt. A Programátorovi došla Pamäť. A Projekt zhavaroval. A Programátorov Manažér prišiel k Programátorovi, hovoriac: "Ty si zmeškal termíny a prekročil rozpočet. Tvoj kód je nanič. Naprav svoje chyby, alebo si hľadaj iný džob!" A Programátor veľmi zosmutnel, pretože bez džobu bol odsúdený na život v špine, oblečenie v handrách a bývanie v kartónovej krabici.

A Globálna Premenná zvolala na Programátora, rečiac: "Ty si hlúpy. Ak ma použiješ s Rozvahou, umožním Ti ušetriť Čas aj Pamäť. Ale daj si pozor, aby sa so mnou nevrátili aj Vedľajšie Efekty." A Programátor videl, že ak Globálna Premenná je menená len v jednom Podprograme a v ostatných je použitá Len Na Čítanie, tak má moc a vládu nad Vedľajšími Efektami. A keďže len skrze Globálnych Premenných sa dalo komunikovať s Obsluhou Prerušenia, ktorá nemá Parametre ani Návratovú Hodnotu, Globálne Premenné mali odrazu veľkú cenu.

A tak Programátor vypustil Globálne Premenné z Pekla, aby im vládli Pravidlá Obmedzeného Prístupu. A Projekt bol zachránený, a Programátorov Manažér zahundral: "No už bol najvyšší čas."

A tak sa stalo, že Programátor spoznal, že Pravidlá sú dôležité. Avšak je tiež dôležité vedieť, Kedy treba Pravidlá porušiť.

A to je naše ponaučenie.

Autor: Stu-san (preklad s povolením autora)
Zdroj

<- 5. časť

7. časť -->