ChatGPT

Стил програмирања и стандардизација

Развој **C** кода за уграђене (embedded) системе мора да следи дисциплинован стил и јасно дефинисане стандарде због високе поузданости и дугог животног циклуса који се од ових система очекују. Посебно у критичним доменима (аутомобилска индустрија, индустријска аутоматика, медицински уређаји), софтверски инжењери примењују строге смернице програмирања како би смањили могућност грешака и неодређеног понашања програма. Ове смернице обухватају како општи стил кодирања (конзистентно форматирање, именовање и организацију кода), тако и формалне стандарде безбедног програмирања усмерене на спречавање грешака на нивоу језика.

Индустријски стандарди кодирања за безбедност и поузданост

Најзначајнији скуп правила за стил и безбедност кода у индустрији уграђених система је **MISRA C** стандард (*Motor Industry Software Reliability Association*). MISRA C дефинише строга правила којих се програмери требају придржавати како би избегли неодређено или потенцијално опасно понашање програма. Ова правила, између осталог, укључују забрану коришћења динамичке алокације меморије (нпр. функција *malloc*), неконтролисаних конверзија типова (*cast* операција) и употребе *goto* наредби [1]. Придржавање оваквих стандарда омогућава примену формалне верификације и аутоматизоване статичке анализе кода (помоћу алата као што су *PC-lint* или *Coverity*), чиме се значајно повећава поузданост и безбедност резултујућег софтвера [2]. У домену аутомотиве, поштовање MISRA смерница је де-факто обавезно за испуњавање захтева функционалне безбедности (нпр. у оквиру стандарда **ISO 26262** за аутомобилске системе).

Поред MISRA-е, постоје и други сетови смерница усмерени на побољшање квалитета и сигурности кода. Један од њих је **CERT C** стандард, који представља смернице за сигурно програмирање на **C** језику. Док је MISRA првенствено фокусиран на безбедност система и избегавање кварова (safety) у уграђеним уређајима, **CERT C** нагласак ставља на обезбеђивање софтвера од рањивости и напада (security), пружајући препоруке за спречавање уобичајених софтверских пропуста као што су прекорачење бафера, неконтролисано руковање меморијом и слично [3] [4]. Оба стандарда се широко примењују – MISRA пре свега у аутомобилској и другим безбедносно критичним индустријама, а CERT C у областима где је кључна заштита од сајбер-напада. Важно је нагласити да се MISRA и CERT C не искључују међусобно; напротив, могу се користити комплементарно. Применом MISRA смерница се поставља темељ поузданог и структурно исправног кода, након чега CERT C препоруке додају додатни ниво заштите од злонамерних сценарија, чинећи софтвер и безбедним и сигурним [5]. Поред тога, у пракси се могу срести и други доменски стандарди и препоруке – на пример, **ISO 26262** захтева да произвођачи у аутомобилској индустрији користе одговарајуће стандарде кодирања као део процеса обезбеђивања функционалне безбедности, док **CERT C** допуњује ту причу аспектима сајбер безбедности. У неким организацијама примену налазе и интерни стилски водичи или алтернативни стандарди (попут *Barr-C* смерница за уграђено програмирање), којима се додатно прецизирају правила кодирања у складу са специфичностима пројекта.

Конзистентност стила и одрживост кода

Осим придржавања формалних стандарда, одржавање конзистентног стила кодирања у целом пројекту има велики утицај на читљивост и одрживост софтвера. Под **стилом програмирања** подразумева се читав скуп правила и навика које код чине једноставним за праћење: конзистентно форматирање (увлачење линија, постављање заграда и размакница), смислено именовање променљивих, константи и функција, структурисање кода по логичким целинама, као и писање јасних коментара где год је потребно. Уједначен стил олакшава тимски рад – различити програмери ће брже разумети туђи код ако сви прате исте конвенције. Стилска усклађеност такође поједностављује **code review** поступак (мануелну проверу кода од стране колега) и доприноси смањењу броја грешака у касним фазама развоја.

Стандардизација стила и придржавање договорених смерница данас су саставни део процеса развоја софтвера за микроконтролере. Коришћењем индустријских стандарда као што су MISRA C и CERT C, потпомогнутим алатима за статичку анализу који аутоматски откривају одступања од правила, успоставља се висок ниво квалитета кода [2]. Доследан и добро документован код је не само мање склон грешкама већ је и лакше преносив на нове платформе и одржив током времена. На тај начин, **стил програмирања** и **стандардизација** представљају два повезана аспекта квалитета софтвера – први осигурава читљивост и једнообразност, а други уводи проверљива правила која подижу поузданост и безбедност система у целини. Поштовањем ових принципа, развојни тим гради основу за софтвер који ће бити отпоран на грешке, предвидив у понашању и усклађен са строгим захтевима уграђених критичних апликација. [6]

---

[1] [2] [6] Од изворног C кода до извршне бинарне слике - компилација и распоред у меморији микроконтролера.pdf
file://file-EEqJpprSDsabiR4PCPn5TR

[3] [4] [5] CERT C vs. MISRA C: Decoding the Differences for Secure and Safe Software
https://www.linkedin.com/pulse/cert-c-vs-misra-decoding-differences-secure-safe-mudduluru-jb8we