Biztonságkritikus beágyazott rendszerek szisztematikus tesztelése

Répás Gergő V. Inf., gergo.repas@gmail.com

Konzulens: Dr. Majzik István, MIT, majzik@mit.bme.hu
Pintér Gergely, MIT, pinterg@mit.bme.hu
Piroska László, Robert Bosch Kft., laszlo.piroska@hu.bosch.com

A biztonságkritikus beágyazott rendszereken futó szoftverekkel szemben kivételesen magasak a minőségi elvárások. Ezek vonatkoznak a tesztelési folyamatra is: a fejlesztési szabványok (pl. IEC 61508, EN 50128) által is rögzített alapvető elvárás, hogy a teljes funkcionalitás le legyen tesztelve, valamint a kód minden utasítását végrehajtsák a tesztelés során. A megoldásunkkal ezen elvárások alapján a szoftver tesztelési folyamathoz járulunk hozzá két módszer és az azokat támogató eszközök kidolgozásával. Az első módszer és eszköz a tesztesetek automatikus, formális modell alapú generálását támogatja, a másik pedig az így generált tesztesetek végrehajtását hivatott ellenőrizni kétféle forráskód fedettségi mérték meghatározásával. Ezek a kódfedettségi mértékek azt adják meg, hogy a kód mekkora hányada került végrehajtásra a tesztek futtatása során.

A tesztesetek automatikus generálása kiküszöböli a kézi teszt tervezés hibalehetőségeit és költséghatékonyabb is. A költséghatékonyság fokozottan igaz modell alapú fejlesztés esetén, mikor a formális modell már az implementáció előtt rendelkezésre áll. A teszteset generáló eszköz a szoftver véges-automata modelljéből (mint formális specifikációból) indul ki, mely a rendszer funkcionális viselkedését írja le, megadva annak állapotait és a külső események hatására bekövetkező állapotátmeneteit. Az eszköz ebből a modellből olyan teszteseteket állít elő, melyek végrehajtása során a rendszer összes állapotában minden lehetséges esemény bekövetkezésének hatása szisztematikusan tesztelhető. Ez biztosítja, hogy a szoftver teljes funkcionalitása tesztelésre kerüljön. A modell alapján generált teszteseteket a konkrét tesztelő eszköz formátumára képezzük le, így ezek a megvalósított rendszerben futtathatók. A teszt generátor eszköz futási ideje jól skálázható a modell méretével, a generált tesztesetek összesített hossza pedig minimális. A szoftver teljes funkcionalitása gyakran több lépésben kerül tesztelésre, ezért eszközünk képes a funkcionalitás egy részére is teszt eseteket készíteni, mikor egyes események bekövetkezése nem elvárt, ezáltal a generált teszt esetek hossza is csökken.

A beágyazott rendszerek tesztelése során a kódfedettség mérése a kevés rendelkezésre álló erőforrás, a nehéz hozzáférhetőség (pl. fájlrendszer hiánya) miatt jelent kihívást. Az elterjedt kódfedettség mérő eszközök (pl. ggcov) nincsenek felkészítve ezen korlátok figyelembevételére, ezért készítettük el az alacsony memória és CPU igényű, beágyazott kódfedettség mérő megoldásunkat. A kódfedettséget mérő eszköz C nyelven írt szoftverekhez készült, és a fordító-láncba épül be a C előfeldolgozó és fordító közé. Az eszköz a forráskód felműszerezésével utasítás- illetve döntési ág fedettségi mértékek mérésére készíti fel a programot. Az egyes utasításblokkok illetve ágak végrehajtását jól konfigurálhatóan a lokális memóriába naplózza, amely a tesztelés után off-line módon kiolvasható és elemezhető. A tárfoglalás mértéke jól skálázható a tesztelés többlépéses végrehajtásával. A felműszerezés a program funkcionalitását változatlanul hagyja, valamint a kódméret kis mértékű növekedése árán az is garantálható, hogy a tesztelt és a működő program időzítési viszonyai azonosak lesznek. A felműszerező eszköz mellett a kód fedettséget a kódsorok színezésével megjelenítő alkalmazás is elkészült.