Kutatási beszámoló a Pro Progressio Alapítvány számára

Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Mérnök informatika szak

Orvosi készülékekben használható modern fejlesztési technológiák lehetőségeinek vizsgálata

Beágyazott Linux operációs rendszer optimalizálása

Petrás Péter
hallgató

Méréstechnika és Információs Rendszerek Tanszék
Autonóm Intelligens Rendszerek szakirány
Intelligens rendszerek ágazat

Budapest, 2014.01.31.

Tartalomjegyzék:

1. Bevezetés

A kutatásom során egy orvosi készülék kutatási és fejlesztési folyamatában vehettem részt. Feladatom olyan beágyazott Linux operációs rendszer összeállítása volt, amely képes megfelelni mind a rendszer folyamatait szabályozó szoftverrendszernek, mind a grafikus megjelenítési és konfigurálási feladatokért felelős, felhasználói felületet futtató szoftvereknek is. Mindemellett az operációs rendszernek meg kellett felelnie a méret és sebességbeli követelményeknek, illetve egy robosztus és biztonságos rendszertől elvárt követelményeknek.

Elsődlegesen egy olyan rendszer előállítása volt a cél, amely elég kisméretű ahhoz, hogy egy pendrive-on elférjen, képes legyen arról bootolni, illetve, hogy a tesztelési célokra szánt szoftvereket képes legyen indítás után automatikusan futtatni. Plusz feladatnak merült fel, hogy a bootolható pendrive Windows környezet alól is generálható legyen.

A rendszer generálását úgy kellett megtervezni, hogy a későbbiekben ne csak a tesztelési célokra szánt rendszert, hanem azt kiegészítve egy bővebb szoftverrendszert, vagy azt redukálva egy telepítési célokra szánt rendszert egyaránt képes legyen futtatni.

2. Beágyazott operációs rendszerek

A beágyazott rendszer olyan alkalmazás-orientált számítógép, amelyet egy konkrét feladat ellátására terveztek. Ezek a rendszerek informatikai eszközökre épülő integrált alkalmazások, amelyek közös jellemzője:

- a nagyfokú autonomitás
- a fizikai környezettel való intenzív információs kapcsolat

Az általános célú számítógépekkel szemben csupán néhány előre meghatározott feladatot lát el, és sokszor tartalmazhat olyan feladat-specifikus mechanikus és elektronikus alkatrészeket, melyek nem találhatók egy általános célú számítógépben. Mivel a rendszer feladatai a tervezés idején is jól ismertek, a tervezők a feladatnak megfelelően tudják optimalizálni a rendszert, csökkenteni a költségét és méretét, növelni megbízhatóságát.

A beágyazott információs rendszerek tervezése és kivitelezése, olyan komplex alkalmazói

rendszerek létrehozását célozza meg, ahol az alkalmazás-orientált célberendezések a fizikai környezettel integráltak.

3. Konkrét feladat

Az operációs rendszerek optimalizálásával egy konkrét feladaton keresztül foglalkoztam.

Egy olyan célú futtató rendszert kellett készítenem, amely elfér és képes futni egy viszonylag kisméretű pendrive-on, illetve képes futtatni egy célorientált szoftverrendszert úgy, hogy az alkalmas legyen az orvosi készülék felhasználói interfészeinek (monitor, hangok, ledek) tesztelésére.

Szempont volt mindemellett a rendszer generálhatósága, könnyű reprodukálhatósága, illetve indítási sebességének gyorsítása.

4. Rendszergenerálási lehetőségek

Alapvetően két rendszergenerálási stratégia merült fel a tervezésnél.

Az első stratégiaként az, hogy egy már meglévő minimális rendszer feltérképezése mellett minimalizáljuk és optimalizáljuk azt úgy, hogy csak azokat a szükséges funkcionalitásokat hagyjuk meg ebben a rendszerben, amire a futtatás során később szükség lesz.

Egy másik lehetőségként pedig olyan új, célorientált rendszert kívántunk létrehozni, amely egy már meglévő forrás vagy csomaghalmazból generálható. A generálás után így azt már így nem kell redukálni, csupán a rendszer számára szükséges programcsomagokat tartalmazza. Ezután ezt a rendszert kell kiegészíteni az orvosi készülék folyamataihoz szükséges funkcionalitásokkal.

5. Rendszergenerálási lehetőségek tesztelése

A kutatásom során háromféle generálási folyamatot teszteltem.

Az első esetben egy hivatalos forrástól ingyenesen letölthető minimális operációs rendszert

elemeztem. A telepített szolgáltatásokat, csomagokat és programokat, azok függőségeit és telepített fájljait vizsgáltam meg, majd az orvosi készülék igényeihez mérten redukáltam ezek számát. Így egy kisméretű, azonban a követelményekhez képest még mindig túl nagy rendszert sikerült generálni. Problémaként merült fel továbbá, hogy a rendszer reprodukálhatósága bonyolult és nehézkes.

A második esetben egy meglévő forráscsomagból indult a folyamat. Ezekből a forrásokból fordítással juthattunk egy minimális rendszerhez. Ez a folyamat már elfogadható méretű rendszert generált, azonban ezzel kapcsolatban is problémák merültek fel. A rendszer karbantartása, a lefordult csomagok verziójának nyilvántartása és frissítése lényegesen nehezebb, mint más csomagkezelési technikákra épülő folyamatok esetében. Plusz problémaként merült fel, hogy egy forrás verziójának változtatása során szükséges lehet nem csak ennek a programnak az újrafordítása, hanem a teljes rendszer újrafordítása, ez pedig jelentősen több időt vesz igénybe.

A harmadik esetben egy meglévő csomaghalmazra épülő telepítési folyamatot vizsgáltam. A csomaghalmaz ez esetben kezelhető általános csomagkezelési és csomagtelepítési célokra használható szoftverekkel. A rendszer tulajdonképpen a szükséges könyvtárstruktúra létrehozásából, a szükséges minimális rendszerhez szükséges csomagok telepítéséből, a kernel és kernelmodulok telepítéséből, a bootoláshoz szükséges fájlok generálásából, illetve a tesztelési célokra szánt szoftvernek szükséges csomagok telepítéséből épül fel. Ez a rendszer méretben és sebességben a fordítással keletkezett rendszerhez hasonló eredményt produkált.

A tapasztalatok után a csomaghalmazból történő generálás mellett döntöttünk, ugyanis ebben az esetben lényegesen kisebb méretet kapunk, mint egy már meglévő rendszer redukálási folyamata során és elkerüljük a fordítási folyamatok során felmerülő verziókezelési nehézségeket is.

6. Telepítés és bootolás pendrive-ról

Miután elkészült a rendszerünk, már csak arra kellett azt rávenni, hogy képes legyen bebootolni és automatikusan futtatni a tesztelési célokra szánt szoftvereket egy pendrive-ról úgy, hogy az orvosi készüléknek az adattárolási erőforrásait ne használja, és a pendrive-ra írni ne tudjunk.

Alapvetően kétféle megoldás született. Mindkét esetben a memóriába töltődnek be a pendrive-on tárolt adatok, azonban az egyik esetben egy Linux környezetben használt EXT3 fájlrendszerként

települ rá a generált rendszer, a másik esetben pedig a Live CD-khez hasonlóan egy tömörített squash filesystem-ként tároljuk a generált rendszert. Kutatásom során a squash filesystem-es megoldással foglalkozhattam.

Ennek hátránya a másik megoldással szemben, hogy ezt a tömörített fájlrendszert külön kell generálni. Ha módosítani akarunk rajta, ezt a rendszert ki kell tömöríteni (vagy újra kell generálni), és ezután tudjuk elvégezni a módosításokat. A módosítások után pedig újra kell tömöríteni. Így fejlesztési fázisban ez a folyamat sok felesleges időt elvisz.

Előnye viszont, hogy a rendszer kevesebb helyet foglal, így bootolás során a memóriába töltése kevesebb időt vesz igénybe, amivel a folyamat gyorsul. További előnye, hogy ezt a pendrive-ot Windows környezetben is könnyen tudjuk generálni, ami pedig követelmény volt. Ebben az esetben Windows alól egy Linuxra szánt EXT3 fájlrendszer generálása helyett egy FAT32-es fájlrendszert használó pendrive-ot tudunk formázni (ami Windows alatt is triviális), majd a bootolhatóságért felelős syslinux telepítése után csak fájlmásolási műveleteket (generált rendszer, bootoláshoz szükséges fájlok) igényel a folyamat.

7. Értékelés

Az így keletkezett rendszer az elvárásoknak megfelelő méretet és indulási sebességet produkálta, és a tesztelési célokra szánt szoftverrendszert is megfelelő sebességgel képes futtatni.

A generálási folyamat úgy lett kialakítva, hogy könnyen frissíthető legyen, illetve, hogy képes legyen ne csak a tesztelési célokra szánt rendszer generálására és futtatására, hanem akár egy bővített szoftverrendszer futtatására szolgáló rendszer vagy egy telepítési célokra irányuló rendszer generálására is.