

1. FEJEZET

Bevezetés a Symbian operációs rendszerbe

Napjainkban a mobilkommunikáció szerepe és piaca átalakulóban van. A pusztán távközlésre kialakított eszközökből a technológiai fejlődés, a felhasználói igények és a piaci verseny nyomására egy új eszközcsalád, az okos telefonok (smartphone-ok) családja fejlődött ki. Az ezektől a készülékektől elvárt szolgáltatások között található a multimédiás anyagok (hang és videó) összeállítása, elküldése, megtekintése, az adatkommunikációt igénybe vevő üzleti alkalmazások telepíthetősége, szórakoztató, már-már a kézi játékkonzolok minőségét megközelítő játékok futtatása, az új technológiák magától értetődő alkalmazása.

A készülégyártók nagy csoportja felismerte, hogy ezekre a kihívásokra egy naprakész, közösen fejlesztett operációs rendszer jelentheti a leggazdaságosabb választ. Természetesen minden gyártó szeretné, ha ez a közös platform testre szabható lenne a saját készülékeik képességeire, valamint megjelenésében tükrözné a cég saját elképzeléseit, egyéni arculatát. A könyvben bemutatott Symbian operációs rendszer és a rá épülő Series60 platform célkitűzése, hogy mindezekre a gyakorlati kérdésekre sikeres választ adjon.

1.1. Az operációs rendszer múltja

A Symbian operációs rendszer gyökerei az 1980-as évek végéig nyúlnak vissza. Ekkor kezdte kifejleszteni a Psion Computers cég a kézisámítógépeihez a SIBO, majd az EPOC operációs rendszert. Az EPOC figyelemre méltó tulajdonságokkal rendelkezett, a kézisámítógépek teljes irodai szoftverkészletet tartalmaztak, és gyakori

elemcsere nélkül is meglehetősen hosszú üzemidőt értek el. Az EPOC 1998-ra az 5. verzióig jutott el. Ekkor vette át a fejlesztését a Symbian, és az operációs rendszer is új nevet kapott: Symbian OS lett belőle.

Magát a Symbian céget 1998 júniusában alapította a Nokia, a Motorola, a Psion, valamint az Ericsson. 1999-ben csatlakozott hozzájuk a Matsushita, majd 2002-ben a Sony-Ericsson és a Siemens.

A Symbian számára az áttörést a 2000-ben megjelentetett 6.0-s verzió jelentette. Az első erre épülő eszköz a Nokia 9210 (kommunikátor) telefonja volt. A 2001 januárjában elkészült Symbian OS 6.1, majd a valamivel több mint egy évvel későbbi 7.0-s verzió már számos okos telefon alapját képezte.

1.2. Az okos telefonok képességei

Az okos telefonok, mint a nevük is mutatja, rengeteg fejlett tulajdonsággal rendelkeznek a régebbi mobilokhoz képest. Erős processzoruk, nagyméretű, színes kijelzőjük, viszonylag nagy memóriájuk lehetővé teszi személyes információkezelő programok kényelmes alkalmazását (például e-mail kliens, fejlett naptár, címjegyzék), kommunikációs szoftverek (MMS, WAP- és HTML-böngészők stb.) gyors alkalmazását, egyéni alkalmazások (például szótárak, térképek, útikalauzok, üzleti programok stb.) telepítését, a telefon testreszabását. Az 1.1. ábrán bemutatunk néhány Symbianra épülő okos telefont.



1.1. ábra: Symbian okos telefonok

Fontos ugyanakkor észrevenni, hogy az okos telefonok felhasználói a készülékeiktől a régi mobiltelefonoknál megszokott egyszerű kezelhetőséget és megbízhatóságot várják el. A PC-k világában elfogadottnak nevezhető működés – például a számítógép újraindulása, az erőforrások pazarló használata, a hibás szoftver javítócsomagokkal való tökéletesítés – az okos telefonoknál nem megengedhető. A meglévő operációs rendszerek leskálázása a felhasználói utasításokra való reagálás idejének, a válaszidőnek sem kedvez: senki nem akar másodperceket várni például egy hívás kezdeményezésére amiatt, mert az operációs rendszer éppen valamilyen belső működést végez.

A Symbian filozófiája szerint a mobiltelefon-készülékek piacát öt jellemző kulcselem teszi egyedivé, és az ezek alapján összeálló koncepció határozza meg az operációs rendszerrel szemben támasztott alapvető igényeket.

A mobiltelefon kicsi és hordozható, ami jelentős felhasználói elvárásokat eredményez. Egyrészt a hardver méretének kezelhetőnek kell lennie. A technológia megengedi az eszközök miniaturizálását, azonban ez sem mehet a használhatóság rovására. A telefonra minél nagyobb kijelzőt és kényelmes beviteli eszközt (billentyűket) kell elhelyezni. Másrészt elvárható, hogy a telefonra töltött alkalmazások együtt tudjanak működni. A névjegyzékben rögzített adatokat szeretnénk e-mail küldéshez, telefonáláshoz, SMS küldéséhez is használni.

A hordozhatóság elkerülhetetlenül felveti az energiafogyasztás problémáját. A telefonnak úgy kell gazdálkodnia az energiával, hogy az ellátást biztosító akkumulátort minél ritkábban kelljen újratölteni. Emiatt az operációs rendszer nem lehet erőforrás-pazarló, minél hatékonyabban kell futnia, hiszen nagy teljesítményű processzort a fogyasztása miatt nem lehet az eszközökbe építeni. A telefon működtetésével, az alkalmazások egyidejű futtatásával és váltásával kapcsolatos teendőket minél kisebb ráfordítással kell tudnia elvégezni. Mindezzel ellentétes követelmény, hogy a telefonnak az energiatakarékos, készenléti állapotból gyorsan aktívvá kell válnia, a felhasználói parancsra, bejövő hívásra stb. minél kisebb válaszidővel kell reagálnia.

A mobiltelefonok gyakran napokig be vannak kapcsolva, eközben nem szabad újraindulniuk, összeomlaniuk. Elinduláskor (booting) szintén nem várakoztathatják hosszan a felhasználót.

A minél szélesebb felvevőpiac megcélzása számos további követelményt von maga után. A tömegpiacra szánt mobiltelefonoknak megbízhatónak kell lenniük. Semmilyen körülmények között nem léphet fel adatvesztés (például a személyes információkezelő programok adatai), a telefon nem dobhatja el a vonalat, és az operációs rendszernek nem szabad összeomlania erőforráshiány, szélsőséges körülmények vagy hibásan megírt alkalmazások esetén. Ezt a mikrokernél-megközelítéssel lehet leginkább elérni: ekkor csak egy igen kis méretű kód (a mikrokernél) fut privilegizált módban, minden más rendszerkomponens erre épül, így alakítja ki a moduláris, biztonságos és könnyen bővíthető rendszert.

A modularitás a memóriafelhasználásnak is kedvez, mivel az általuk biztosított szolgáltatásokat nemcsak a rendszer, hanem az alkalmazások is egyaránt felhasznál-

hatják. A memóriafelhasználás egyébként is kényes kérdés e mobileszközöknél. Amennyiben a programok szivároztatják a memóriát (memory leaking), az gyorsan elfogyhat egy viszonylag kevés erőforrással rendelkező, de állandóan bekapcsolt készülék esetén. Ezért fontos egy könnyen alkalmazható hibakezelő keretrendszer bevezetése, amely az efféle hibákat kiküszöböli, és a defenzív programozói stílus, vagyis a kivételes helyzetekre való felkészülés azokban az esetekben, amelyekben az operációs rendszertől nem várhatunk el támogatást.

A hibakezelő keretrendszer mellett a jó szoftvertervezés is segít a robusztus, hibamentes kódok elkészítésében. Ehhez tervezési mintákkal, illetve jól megtervezett alkalmazás-keretrendszerrel járul hozzá a Symbian.

Az okos telefonnak fel kell készülnie **nem állandó kapcsolatok kezelésére**, hiszen egy mobiltelefonnal a mozgó távközlési hálózatra szeretnénk csatlakozni, sőt a kis hatósugarú kommunikációt (PAN, Personal Area Network) is vezeték nélküli technológiákkal (például Bluetooth) szeretnénk megvalósítani. A vezeték nélküli kommunikáció azonban nem folyamatos: bármikor elmehet a térerő, megszűnhet a rálátás az infraeszközre, a telefonnak azonban ekkor is működőképesnek kell maradnia, offline szolgáltatásokat kell nyújtania, vagyis nem működhet vékony kliensként. Kezelnie kell az átvitelből adódó hibákat, szükség esetén a felhasználót is tájékoztatva.

Egy fejlett mobiltelefonról elvárjuk, hogy ismerje és használja az összes fontos távközlési protokollt. Az alkalmazás szintjét nem érintheti az, hogy alatta éppen milyen kommunikációs technológiát alkalmaznak. Az operációs rendszernek bővíthetőnek kell lennie újabb, esetlegesen alkalmazásspecifikus protokollokkal is.

Az operációs rendszerre épülő **eszközök sokfélesége** látszólag ellentmondásos helyzet elé állítja a fejlesztői közösséget. Mindegyik telefontyártó cég szeretné, ha a készüléke hordozná az ő egyedi arculatát mind felépítésében, mind megjelenésében. Az ellentmondás feloldását az operációs rendszer alapjának és a felhasználói felületnek (User Interface, UI) a szétválasztása jelenti. A piaci igények is megkövetelik különböző interfészekkel ellátott telefonok megjelenését. Nyilván teljesen más alkalmazási területei lehetnek egy viszonylag kis képernyős, külön alfanumerikus billentyűzetet nem tartalmazó okos telefonnak, mint egy érintőképernyős, PDA-szerű modellnek vagy egy teljes billentyűzetet és irodai szoftvercsomagot is tartalmazó készüléknek.

A könyv által megcélzott platform, a Series60 az okos telefonokhoz a Nokia által kifejlesztett UI, amelyet azonban már további gyártóknak is továbblicencelt.

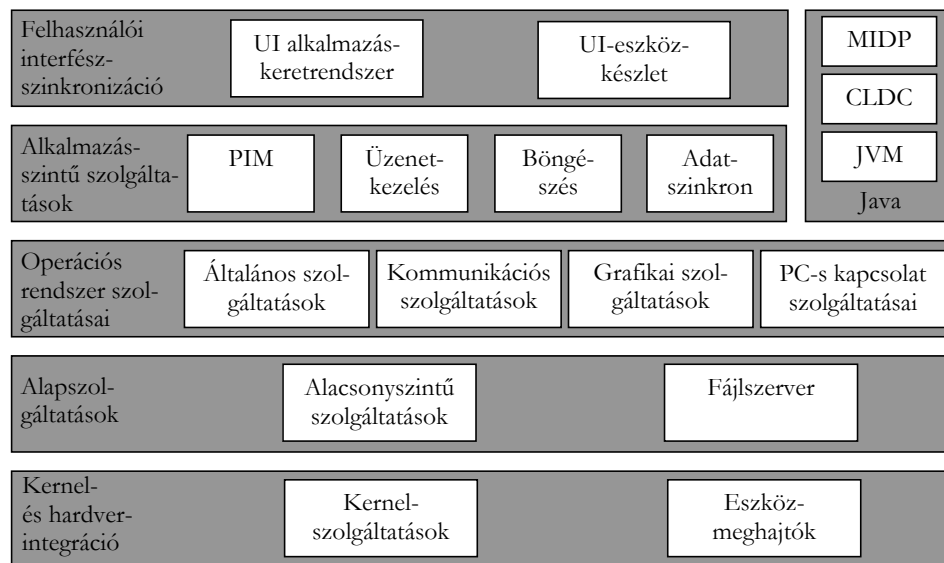
Elvárható, hogy egy tömegpiacot célzó mobiltelefon **nyitott legyen** a külső fejlesztések irányába. Az operációs rendszer elfogadottságát és népszerűségét növeli, ha számos alkalmazás kapható hozzá a kommunikáció, a produktivitás növelése vagy szórakoztatás céljaira. Ahhoz, hogy a platform a fejlesztők körében is népszerű legyen, a nyitottságon kívül egyéb körülmények is szükségesek. Az egyik legfontosabb, hogy az operációs rendszer programozásának megismerése időtálló befektetés legyen, vagyis újabb eszközök és technológiák megjelenésével ne kelljen teljesen új tudást felvenni. A tanulási folyamatot könyvek, letölthető anyagok, közösségi portálok,

kurzusok is segítik. A programozás elterjedt nyelveken (C++, Java), objektumorientált megközelítésben folyik, és az interneten bárki számára letölthető fejlesztőkészletek (Software Development Kit, SDK), eszközök, illetve eszközemulátorok állnak rendelkezésre.

A tanulási időt csökkentő tényező továbbá a kommunikációs és egyéb szabványok széles körű alkalmazása.

1.3. A Symbian felépítése

Az 1.2. ábrán látható a Symbian operációs rendszer felépítésének vázlatja. A legelső, integrációs rétegre alacsony szintű szolgáltatások épülnek, ezt egy middleware szint követi (grafikai, kommunikációs és egyéb szolgáltatásokkal), majd az alkalmazásszintű szolgáltatások jönnek, végül legfelül a felhasználói interfész (igény szerint testre szabható) elemei találhatóak. Az architektúra ugyancsak tartalmaz egy fejlett, mobil környezetre optimalizált Java futtatókörnyezetet is.



1.2. ábra: A Symbian operációs rendszer felépítése

A legalsó réteg nem más, mint egy kernel- és hardverintegrációs szint, amely a különböző eszközök felépítésbeli eltéréseit fedi el. Így lehetővé válik, hogy az operációs rendszer új hardvereszközre való telepítésekor (base porting) ne kelljen a felsőbb szintekben módosítani. Az integráció nagyfokú robusztusságot és megfelelő teljesítményt biztosít.

A mikrokernel közvetlenül a processzorban fut (natív környezete a 32 bites ARM processzor), privilegizált üzemmódban. Az operációs rendszer a kernel-, illetve felhasználói (user) processzek ütemezéséhez preemptív stratégiát alkalmaz. A kernel többszálú programvégrehajtást biztosít, a legnagyobb prioritást a kernelszálnak tartja fenn, amely a kliensek kéréseit szolgálja ki. Felhasználói kód csak a User könyvtáron keresztül (*euser.dll*) kerülhet kernel módba.

Az eszközmeghajtók szerepe a hardver–szoftver együttműködés megteremtése, így például a kijelző, a billentyűzet vagy akár a kommunikációs csatornák elérésének biztosítása, a konkrét hardverfüggő részek implementálásával. A meghajtók általában két részre bonthatók: fizikai meghajtókra (Physical Device Drivers, PDD), illetve logikai meghajtókra (Logical Device Drivers, LDD). Ez utóbbiak egy adott eszköztípusra vonatkozó hasonló tulajdonságok kezelésére tartalmazznak magasabb szintű funkcionalitást.

Az alapszolgáltatásokat nyújtó szint leginkább az operációs rendszer többi komponensének az alapját adja. Egyik legfontosabb eleme a fájlserver, amely a fájlrendszerhez biztosít megosztott hozzáférést. Beépített moduljai mellé (VFAT, Logging Flash FS) újraindítás nélkül, dinamikusan adhatunk újabb elemeket. Az alacsony szintű szolgáltatások közé tartoznak többek között a különböző biztonsági szolgáltatások, a beépített adatbázis-kezelés, az energiagazdálkodás, a karakterkódolási eljárások, az XML-feldolgozás, valamint az alkalmazások beállításainak kezelése.

Az operációs rendszer legfontosabb elemét jelentő szolgáltatásokat egy middleware rétegbe gyűjtötték össze. A biztonsági szolgáltatások két alapvető része a kriptográfiai modul, valamint a tanúsítványkezelő modul. Az operációs rendszer biztosítja az összes fontos és elterjedt kriptográfiai algoritmust, mint a szimmetrikus (DES, 3DES, RC2, RC4, RC5) és aszimmetrikus (RS, DSA, DH) rejtjelező algoritmusokat vagy a különböző hashfüggvényeket (MD5, SHA1, HMAC).

A multimédia terén nyújtott funkcionalitást keretrendszerekbe és programozói könyvtárakba szervezték. A multimédia-szerver az audio- és videoanyagok felvételét és lejátszását támogatja, valamint különböző képmegjelenítési és -feldolgozási képességekkel rendelkezik. A támogatott formátumok egyébként is széles skálája természetesen bővíthető. A window server egy hatékony ablakkezelő rendszer, amelynek alapvető feladata, hogy megossza a képernyőt és a beviteli eszközöket az alkalmazások között.

Ugyancsak fontos eleme a middleware rétegnek a kommunikációt támogató szolgáltatások gyűjteménye. Ez három alapvető részre oszlik. A telefónia alrendszer a gyártók valósítják meg, ennek integrálása teszi lehetővé az operációs rendszer egyéb részei számára a komplex adatkommunikáció biztosítását. A második rész a soros, illetve a rövid hatótávú vezeték nélküli kommunikációt takarja (Personal Area Network, PAN). A pont–pont kapcsolat történhet USB-n, infrán vagy Bluetoothon.

A kommunikáció harmadik csoportjába a hálózati szolgáltatások tartoznak. Az operációs rendszerben megtalálható TCP/IP stack, HTTP, valamint WAP stack megvalósítás, és természetesen minden fontos protokoll implementálva van hozzájuk.

Ebben a rétegben található meg az alapvető grafikai szolgáltatásokat nyújtó könyvtár, amely a képernyő, a billentyűzet és a mutatóeszköz megosztott kezelését is biztosítja. A GDI-t (Graphical Device Interface) használhatjuk a képernyőre, nyomtató eszközre vagy memóriába (bitmap) történő rajzolás során. Ugyancsak az operációs rendszer szolgáltatásai közé tartozik a mobil eszköz számítógéppel való összekapcsolását biztosító eszközkészlet.

Az operációs rendszer következő rétege alkalmazásszintű szolgáltatásokat biztosít. Ebben olyan alkalmazásmotorokat találunk, amelyek a mobiltelefonokban szokásos, a felhasználó adatait tároló és kezelő programok készítését könnyítik meg és teszik egységessé. A személyes információkezelő modul (Personal Information Management, PIM) különböző naptár- és határidő-funkciókhoz, személyes kapcsolatok (kontaktok) kezeléséhez, teendőlisták kialakításához ad támogatást. Az üzenetkezelő keretrendszer az elterjedt üzenettípusok (SMS, EMS, MMS, e-mail, fax, BIO) küldését és fogadását végző alkalmazások készítését könnyíti meg. A böngészést (browsing) segítő alrendszer általános szolgáltatásokat biztosít mind a beépített, mind a külön telepített tartalom megjelenítőknél. Az operációs rendszer tartalmaz még egy OMA (Open Mobile Alliance) SyncML 1.1.2-es motort adatszinkronizáció céljára.

A rendszer tartalmaz Java virtuális gépet is (Java Virtual Machine, JVM), amely a mobileszközökre írt, hordozható alkalmazások futtatását teszi lehetővé. Az operációs rendszertől függően a MIDP profil 1.0-s vagy 2.0-s verzióját valósítja meg.

A felhasználói interfész szintje nagyfokú rugalmasságot biztosít a gyártóknak a megjelenítéssel kapcsolatos különböző elképzeléseik megvalósításában, miközben az operációs rendszer szolgáltatásai a közös programozói interfész (API) miatt ugyanúgy érhetőek el. Ez a Symbian alapú készülékekre írt programok különböző eszközökre való átvitelét is megkönnyíti. Ebből következik, hogy a könyvben leírt alapelveknek az a része, amelyek a Series60 felhasználói interfészt nem érintik, bármely más Symbian alapú rendszerrel is felhasználhatóak.

1.4. A könyv tartalma

A további fejezetekben a Series60 alapú Symbianfejlesztésről lesz szó. A fejezetek felépítése a következő:

1. Bevezetés. Az operációs rendszer múltja. A Symbian felépítése.
2. A Symbianfejlesztés alapelemei. Elnevezési konvenciók, kivétel- és memória-kezelés. Tervezési minták Symbian alatt. A fejlesztési folyamat áttekintése.

3. A legkisebb Symbian alkalmazás felépítése. A Series60 alkalmazások felépítése, az MVC (Model-View-Controller) minta. A fordítás és futtatás folyamata. Telepítőcsomag készítése.
4. Egyszerű multimédiás alkalmazás készítése. Bittérképek használata és megjelenítése, rajzolás. Animáció készítése. Szinuszos és digitalizált hangok használata. A menük felépítése és használata, dinamikus felépítése.
5. A grafikus felhasználói interfész alapjai. Egyszerű és összetett controlok készítése. A felhasználó beavatkozásának kezelése. Panelek, a státuspanelt felépítő komponensek.
6. Grafikus beviteli elemek a felhasználói interfészen. Dialógusok előállítása erőforrásból, a dialógusok kezelése, űrlapok használata. Szerkesztőmezők és paramétereik. Lekérdezőablakok előállítása, értesítések készítése.
7. Listák típusai, listaelrendezések. Listák előállítása erőforrásból és programkóddal. Listaelemek manipulációja. Beállításlisták készítése. Kétdimenziós megjelenítés (rácsok).
8. A fájlserver kliensoldali használata, adatfolyamok használata. Store-ok készítése, a store-ok fajtái. Az adatbázis-kezelés módjai.
9. A Symbian kommunikációs és üzenetkezelő alrendszerének áttekintése. Infravörös átvitel. A Socket Server és a Messaging Server használata. Különböző üzenetek küldése és fogadása.
10. Párhuzamos programozás Symbian operációs rendszeren. Processz- és szálkezelés, processzek közötti kommunikáció. Időzítők, értesítők. Az ActiveObjectek működése, készítése és használata.