

## КАК ФУНКЦИОНИРАТ ОПЕРАЦИОННИТЕ СИСТЕМИ?

**Себастиан Янев<sup>1)</sup>**

*Висше училище по телекомуникации и пощи*

**Резюме.** Операционната система управлява хардуера, програмите, файловете системи. В доклада е описан процесът на инсталиране и зареждане на една операционна система. Представени са MBR и GPT файлови таблици за физическите твърди дискове, както и начинът на запис и първоначалното зареждане. Дефинирано е понятието „ядро“, като е представена неговата структура и предлаганите от него услуги. Описани са възникналите процеси и начините за тяхното управление. Направен е анализ на файловете системи. Представени са командният и графичният интерфейс като начин за управление на една операционна система.

*Keywords:* operating system; kernel; tree structure; BIOS; MBR; GPT

Операционната система (ОС) управлява хардуера, програмите и файловете системи в нея. BIOS намира първия зареждащ сектор от твърдия диск и операционната система се стартира в оперативната памет. ОС комуникира с хардуера с помощта на драйвери. Съществуват два типа таблици на дисковете: MBR и GPT<sup>2)</sup>.

Основен елемент на всяка операционна система е ядрото, което отговаря за връзката на потребителските програми с прилежащия хардуер и съставляващия софтуер (Lavington, 1980). ОС осигурява процесите и съответните прекъсвания. Съществуват два типа интерфейси: команден и графичен. Командният интерфейс се използва предимно от програмисти, като по-бърз начин за осъществяване на съответните задачи с помощта на команди. При графичния интерфейс се използват прозорци, лента със задачи, основни програми за работа. Информацията се съхранява във външни носители.

Операционната система е проектирана на принципа на дървовидната структура, по-лесно се намира информацията. Чрез файловата система потребителите могат да разглеждат своите файлове, а администраторът задава правата към собственика, групата и всички останали потребители. Инсталацията на всяка ОС се осъществява чрез външен носител – CD или DVD, като

тя може да бъде записана и в iso файл. От BIOS се задава носителят за инсталация. Създават се дяловете, часовата зона, езикът и др.

За да се опише структурата на една операционна система, като се опише начинът на конфигурирането ѝ, са поставени следните задачи.

- Дефиниране на понятието ядро, като основен елемент на всяка ОС.
- Описание на процеса на изпълнение на поставените задачи чрез графичен или команден интерфейс.
- Представяне на структура за съхранение на данните в ОС.
- Описание на видовете файлови системи, както и дефиниране на правата на потребителите.

### **Зареждане на операционната система**

При стартирането на всеки компютър се стартира BIOS (Basic Input/Output System), записана на чип на дънната платка (ROM чип). Тази програма идентифицира и инициализира всички периферни и системни устройства. Основната функционалност на BIOS е проверката на нормалното функциониране на хардуера POST (Power-On Self Test) (Bandel, Napier, 2010). BIOS е 16-битова програма, която съдържа таблица, като определя реда на зареждане на дисковете. Следваща стъпка е зареждането на операционната система. За тази цел BIOS открива първия 512-битов сектор от диска с операционната система (цилиндър 0, глава 0, сектор 1) – така наречения Master Boot Record (MBR). На този сектор е записан код, който продължава зареждането на инсталираната операционна система. След като BIOS открие MBR кода, започва действителното зареждане на ОС в паметта и неговото изпълнение, като открива първичния активен дял с указания за разчитане на файловата система. От този момент нататък операционната система използва драйвери за достъп до хардуера (Bandel, Napier, 2010). Процесът при EFI е идентичен с тази разлика, че EFI има указания за разчитане на файловата система и може да започне директно зареждането на системата. При Mac компютрите това се нарича BootX. Щом започне зареждането на ОС, BootX инициализира различните системи, включително процесор, памет, устройства, свързани с дънната платка, приложения и потребителски интерфейси.

### **Файлови таблици за физическите твърди дискове MBR и GPT**

MBR (Master Boot Records) е запис в първия физически сектор на хард-диск, като този запис се прочита от BIOS веднага след проверката на паметта, дисковете и периферните устройства. MBR стандарта запазва информацията за адресирането на логическите блокове и техния размер в 32 бита, докато GPT разполага с 64 бита за същата информация (нарича се още LBA адресиране). По подразбиране GPT таблицата има възможност за именуване на до 128 дяла, докато MBR таблицата – само четири дяла, или три основни дяла (Primary Partitions) и един разширен дял (Extended Partition). Този така наре-

чен Extended дисков дял може да бъде разделян на няколко логически дяла (Logical Partitions) (Bandel, Napier, 2010).

GPT (Guid Partition Table) е формат на файловата таблица, на който използва GUID (Globally Unique Identifiers). GPT е произволен номер за дисковете и дяловете в зависимост от определения размер за файловата таблица, което премахва нуждата от разширени и логически дялове.

### **Дефиниция, организация и услуги, изпълнявани от ядрото на една ОС**

Ядрото (kernel) е елемент на системата, изпълняван в защитен режим, осигуряващ връзката на потребителските програми с прилежащия хардуер и софтуерните съставляващи (файлова система, мрежови протоколи). Ядрото предоставя основните функционалности: създава и управлява процесите, предоставя функции за достъп до файловата система и средствата за комуникация (system calls) (Lavington, 1980). Те са единственият интерфейс към функционалностите на ядрото.

Ядрото е софтуер, който предоставя функционалност, необходима за имплементацията на допълнителното софтуерно обкръжение. Така например достъп до файлова система и мрежовите функции се изпълняват като допълнителни услуги. Съвременната тенденция при ОС е запазването на малък размер на ядрото, като услугите се изпълняват от отделни приложения. За тази цел се използва монолитно ядро за постигане на необходимата производителност и лекота при проектиране.

Операционната система Linux, като UNIX базирана, е с отворен код и предоставя възможността за едновременната работа на няколко процеса от потребителя. Тя осигурява мощност, гъвкавост и икономичност <sup>3)</sup>.

Първите версии на Windows са базирани на MS-DOS, а по-късните са NT базирани. ОС Windows осигурява едновременната работа на няколко процеса на един потребител и използва затворен код. Голяма част от ядрото е написано на машинно независим код. Основните характеристики на тази ОС са:

- стартов процес на ниско ниво;
- управление на прихващанията и грешките;
- управление на ниско ниво на изпълнимото съдържание на процес;
- конфигуриране и инициализиране на хардуера;
- изпълнима поддръжка за I/O.

Основна функционалност на ядрото са: таймер и управление на системния часовник, управление на дескриптора и системни процеси, управление на виртуалната памет (paging и swapping), основни системни интерфейси (управление на всички дискове) – I/O, управление, файлова система, поддръжка на терминали и мрежова комуникация.

Границата между ядрото и потребителските програми се налага от защитен режим на прилежащия хардуер. Ядрото оперира в отделно адресно поле,

недостъпно за потребителските процеси (Bandel, Napier, 2010). Привилегирвани операции, като започването на I/O или изключването на процесора, са достъпни само за ядрото. Процесите изискват услуги от ядрото чрез system calls. Тези заявки са механизмът, чрез който потребителските процеси изискват от ядрото извършване на сложни операции, като запис върху носител, или прости операции, като връщането на текущото време. Всички заявки изглеждат синхронни за приложението. То не се изпълнява, докато действието, свързано със system call, се изпълнява. Ядрото може да свърши с изпълнението на някои операции, свързани със заявката, и след като се е възстановил от system call. Например при заявка за запис ядрото копира нужната информация в буфера си, но се възстановява, преди да е приключил записът от буфера на носител.

### **Управление на процесите**

Всяка задача или ред от изпълнения се нарича процес. Контекстът на процес се състои от потребителско ниво, включващо съдържанието на неговото адресно поле и средата за изпълнение, както и от кернелско ниво, което включва параметри на планирането, контроли на ресурсите и идентификационна информация. Процесът включва всичко използвано от ядрото при доставяне на необходимите му услуги. Потребителят може да изпълнява процес, да контролира неговото състояние по време на изпълнение и да получава информация за статуса по време на изпълнението. На всеки процес се присвоява уникална стойност, наречена идентификатор на процеса (PID). Тази стойност се използва от ядрото, за да идентифицира процес, когато докладва неговото състояние на потребителя, или от потребителя, когато той изпраща system call чрез ядрото към процеса (Bandel, Napier, 2010).

Ядрото създава процес, като дублира контекста от друг процес. Новият процес се нарича child process на оригинала, който, от своя страна, се нарича parent process. Контекстът, дублиран при създаването на процес, включва както потребителското състояние на изпълнение, така и системното положение на процеса, управляван от ядрото.

Системата определя набор от сигнали, които може да се подават на процес. Когато се генерира сигнал, той се блокира от по-нататъшно появяване, докато не бъде обработен (caught). Стандартното действие на сигнал е да прекрати процеса. В някои случаи това е съпроводено със създаването на core files, които съдържат копия на адресното пространство на процеса. Обработката на сигнала може да не е задължителна. Той може да се игнорира. Някои сигнали не могат да се игнорират – това са SIGKILL, който прекратява текущия процес, и SIGSTOP, който регулира работата на процеса.

Всички сигнали са с еднакъв приоритет. Когато възникват едновременно, обработката им зависи от конкретната имплементация. Има специални ме-

ханизми, които предпазват критични части от кода от получаването на някои сигнали. За да се осигури контрол над много зависими процеси, се създават групи от процеси – process groups. Групирането на процеси води до улеснено изпращане на сигнали до много процеси, както и до контролиране на достъпа до терминали. Има методи, чрез които да се промени текущата група. Създаването на нова група не е сложен процес, като стойността на новата група е идентификаторът на създавания процес.

### **Команден и графичен интерфейс**

Когато една ОС приключи с изпълнението на дадено указание, командният интерпретатор извежда на екрана съобщение за нови разпореждания. Потребителят може да въведе от клавиатурата редица от знакове, завършваща със специалния знак за край на въвеждането Enter. Тази редица от знакове се нарича команден ред. Той определя коя команда трябва да се изпълни и кои файлове ще се използват при това изпълнение. Този начин често е предпочитан от програмистите, защото някои задачи могат да се изпълнят само с команди, като се спести време при изпълнението (Lavington, 1980). Команден интерфейс използват MS-DOS и някои LINUX дистрибуции.

Основни команди в MS-DOS са: извеждане на съдържание в директория (dir); смяна на текущата директория (cd); връщане в предишна директория (cd); команда за създаване на нова директория (md); изтриване на празна директория (rd); стартиране на изпълним файл ([устройство:][път]име[.разширение]); копиране на файл (copy); преименуване на файл (ren); изтриване на файл (del) и др.

Основни команди в LINUX: извеждане на съдържание в директория (ls | ls -l | ls -a); смяна на текущата директория (cd); връщане в предишна директория (cd ..); команда за създаване на нова директория (mkdir); команда за изтриване на директория без потвърждение (rm -rf); копиране на файл (cp); преименуване на файл (mv); изтриване на файл (rm) и др.

За да работи една ОС с файлове, се използва дървовидна структура (каталог, папка, справочник с файлове и директории, включени в нея), чрез тази структура се улеснява търсенето на файлове.

При Windows текущите устройства се дефинират с буква от латинската азбука (c:\;d:\;e:\), докато при Linux всичко е файл и устройствата се отличават като: /dev/sda(sdb)/ номер.

Графичен интерфейс съдържа всички основни програми на една операционна система, като определя как ще изглежда ОС. Те се показват на екрана и се управляват от посочващи устройства (мишка, клавиатура, touch-screen). Графичният интерфейс използва лентата на задачите (Bandel, Napier, 2010).

В Windows графичният интерфейс се променя в различните версии, но основните компоненти и програми остават неизменени. Windows Explorer се

използва за отваряне на файлове, с програмата „calculator“ пресмята математически задачи, с *notepad* се пишат текстове и т.н.

В Linux съществуват различни дистрибуции и графични интерфейси. Едно от тях е GNOME, което има 28 програми, с които да работи потребителят.

### **Видове файлови системи. Структура**

Потребителите в една ОС могат да създават, модифицират и унищожават своите файлове, като ги разделят един от друг, както и да осигурят ефективното им взаимодействие при изпълняване на общата работа. Потребителите помежду си могат да си предават информация, като се обръщат към файловете със символически имена, а не чрез физическите устройства. Необходимо е файловете да бъдат защитени от неправилен достъп, като за целта администраторът (*root*) променя правата на файловете за достъп (Linux).

Видове файлови системи.

- File Allocation Table (FAT) е файлова система на Microsoft, при която в началото на дисковия носител се разполага файл, който е една таблица, съдържаща информация за физическото местонахождение на всеки един файл в диска, на всеки фрагмент от този файл, на всяка директория, и описва йерархията от файлове и директории в едно дисково устройство.

- New Technology File System (NTFS) е пряк наследник на HPFS, която е използвана в OS/2 (по-късно позната като IBM OS/2 Warp). Тя позволява адресиране до 256 TB. NTFS е файловата система по подразбиране за операционните системи на Microsoft Windows NT, Windows 2000.

- Extended File Allocation Table (exFAT) е файлова система на Microsoft, специално създадена за флаш устройства. Използва се при Windows XP, Vista, 7, 8.

- Universal Disk Format (UDF) е файлова система, стандартизирана от асоциацията OSTA (Optical Storage Technology Association) за съхранение на данни на оптични носители. UDF е файлова система за четене и запис за всички оптични носители (*all optical media*) – най-често при DVD дискове.

В ОС Linux основните файлови системи са *ext2,3,4*, XFS, JFS, ReiserFS и *btrfs*, като за флаш памет се използва FTL, MTD, UBIFS, JFFS2, YAFFS<sup>6)</sup>.

HFS се използва в операционната система Mac OS. При нея томът се дели на по 512 бита логически блокове<sup>4)</sup>.

### **Инсталиране и зареждане на ОС**

Инсталацията на една ОС може да се осъществи локално или виртуално. Съществува възможност да се свали виртуално дисково изображение със съкращение *.iso*. За да се инсталира в BIOS, е необходимо да се укаже от кой носител да зарежда. По време на инсталацията се правят първоначалните настройки за език, часовата зона, дял от твърдия диск, на който да се инсталира. В ОС Linux може да се създаде */boot*, */home*, */swap*/. След

приключване на инсталацията се инсталират и драйверите на дънната платка, видеокартата и др.

Съществуват и други алтернативни начини за инсталиране и стартиране на операционната система: чрез флашка, чрез ram диск или по мрежата. Могат да се създават и група от дялове от различни харддискове, като по този начин да се увеличи бърздействието на системата (RAID).

За инсталирането и зареждането на една ОС на виртуална машина може да се използва част от съществуващия хардуер с помощта на Virtualbox, VMWare и др. Зареждането става чрез виртуални дискове, които представляват файл във файловата система. Потребителят сам задава колко оперативна памет, колко видео памет да използва от наличната му.

### **Заклучение**

Операционната система осигурява връзката между хардуера и софтуера, като осигурява и разпределя отделните процеси. За правилното ѝ инсталиране и конфигуриране са определени: ядрото, като основен елемент на ОС; възможностите за изпълнение на поставените от потребителя задачи с графичен или команден интерфейс; дървовидна структура за съхранение на данните.

Файловата система на всяка ОС се използва от потребителите за създаване, триене и модифициране на файловете, като се задава правото на администратора да променя правата върху файловете.

### **NOTES/БЕЛЕЖКИ**

1. Авторът е студент във Висше училище по телекомуникации и пощи. Консултант при разработване на настоящия текст е гл. ас. д-р. инж. Екатерина Оцетова-Дудин, катедра „Информационни технологии“, Факултет по телекомуникации и мениджмънт, Висше училище по телекомуникации и пощи. E-mail: e.dudin@utp.bg
2. [http://www1.znam.bg/zmonres/edu/IT%209/lessons/os\\_disk/page2.htm](http://www1.znam.bg/zmonres/edu/IT%209/lessons/os_disk/page2.htm)
3. <http://www.digital.bg/novini/%D0%BA%D0%B0%D0%BA%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B8%D0%BE%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%82%D0%B0%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0-news23400.html>
4. <http://www.referati.org/grafichen-potrebitelski-interfeis/85269/ref>
5. <http://alekz.sistemite.com/mbr-vs-gpt/>
6. <http://afks.kaloyan.info/pro/bsdkernel.html>
7. <http://it.souprovadia.info/files/8-MS-DOS.pdf>
8. <http://www.referati.org/grafichen-potrebitelski-interfeis/85269/ref>

## REFERENCES/ЛИТЕРАТУРА

- Lavington, S. (1980). *Early British Computers*. Manchester: Manchester University Press.
- Bandel, D. & Napier, R. (2010). *Linux. Pulno rakovodstvo. Tom I*. Sofia: SoftPress [Бандел, Д. & Нейпиър, Р. (2010). *Linux, Пълно ръководство, Том I*. София: СофтПрес].
- Bandel, D. & Napier, R. (2010). *Linux. Pulno rakovodstvo. Tom II*. Sofia: SoftPress [Бандел, Д. & Нейпиър, Р. (2010). *Linux, Пълно ръководство, Том II*. София: СофтПрес].

## HOW DO THE OPERATING SYSTEMS WORK?

**Abstract.** The operating system manages hardware, programs, file systems. This paper describes the process of installation and loading an operating system. There are shown MBR and GPT file tables for physical hard disks, how to record and how to boot from them. Defined is the term of kernel, its structure and its services. Described is the process and methods for their management, the graphic and command interface of an operating system.

✉ **Mr. Sebastian Yanev**

University of Telecommunications and Post  
Sofia, Bulgaria  
E-mail: sebastianyanev@gmail.com