**Plug and Play la Windows**

Studenti: Nicolae Catalin Ionut 432A

Ungureanu Teodora 432A

Profesor coordonator: dr.ing. Ştefan Stăncescu

Cuprins

[Introducere 3](#_Toc358042855)

[Ce este plug and play 3](#_Toc358042856)

[Dezvoltarea in timp PnP 4](#_Toc358042857)

[Componentele PnP 5](#_Toc358042858)

[Functionare PnP in Windows 6](#_Toc358042859)

[Windows driver model (WDM) 7](#_Toc358042860)

[Introducere 7](#_Toc358042861)

[Tipuri de WDM 7](#_Toc358042862)

[Bus driver 8](#_Toc358042863)

[Function driver 9](#_Toc358042864)

[Filter driver 10](#_Toc358042865)

[Implementare Plug and Play in Windows 11](#_Toc358042866)

[Initializarea driver-ului (Driver\_Entry) 11](#_Toc358042867)

[Initializarea dispozitivului (AddDevice) 11](#_Toc358042868)

[Pornirea dispozitivului (IRP\_MN\_START\_DEVICE) 12](#_Toc358042869)

[Oprirea dispozitivului (IRP\_MN\_STOP\_DEVICE) 12](#_Toc358042870)

[Eliminarea dispozitivului (IRP\_MN\_REMOVE\_DEVICE) 13](#_Toc358042871)

[BIOS-ul Plug and Play 14](#_Toc358042872)

[Introducere Bios PnP 14](#_Toc358042873)

[Obiectivele implementarii PnP in Bios 16](#_Toc358042874)

[Concluzii 17](#_Toc358042875)

[Blibliografie 18](#_Toc358042876)

# Introducere

## Ce este plug and play

Plug and play (PnP) reprezinta o combinatie de suport hardware si software care permite sistemului unui calculator sa recunoasca si sa se adapteze la modificari ale configuratiei hardware cu sau fara mici modificari facute de utilizator. Un utilizator poate sa adauge sau sa stearga diferite dispozitive unui sistem fara a consulta manuale de configurare sofisticate si fara a avea cunostinte solide de hardware.

Desi este o tehnologie promovata de Microsoft si companiile care produc echipamente prezentate ca fiind compatibile cu Plug-and-Play, mecanisme asemanatoare exista si in celelalte sisteme de operare disponibile pe piata. Kernel-ul Linux ofera un aceleasi facilitati ca si plug-and-play prin feature-ul de incarcare automata a modulelor de kernel disponibile in sistem, atunci cand un device este atasat computer-ului. Totusi, sistemele Windows interactioneaza mai bine cu echipamentele atasate prin plug-and-play, datorita faptului ca acestea vin de cele mai multe ori cu drivere specifice, furnizate de dezvoltator, si nu ruleaza sub un modul de kernel care ofera suport generic.

PnP-ul are nevoie de sprijin de la hardware-ul dispozitivului, software-ul de sistem, și drivere. Inițiative în industria hardware au definit standarde (cum ar fi ISA definirea PnP și standardul PC Card) pentru identificarea ușoară a plăcilor și componentelor de bază ale sistemului folosite.

Suportul software-ul de sistem pentru PnP, împreună cu driverele PnP prevede următoarele:

• Recunoastere automata si dinamica a hardware-ului instalat;

• Alocarea resurselor hardware si realocare;

• Incarcarea driver-elor adecvate;

• O interfata de programare pentru drivere pentru a interactiona cu sistemul PnP;

• Mecanisme pentru drivere și aplicații pentru a afla schimbările din mediul hardware și pentru a lua măsurile adecvate;

## Dezvoltarea in timp PnP

Plug-and-play a fost introdus de sistemul de operare Widows 95, dar de atunci a suferit schimbari importante o data cu specializarea sistemelor computationale si cerintelor

utilizatorilor. Maturizarea specificatiilor PnP pentru componentele hardware a fost un proces destul de anevoios, dar sprijinul din partea celor doi giganti, Microsoft si Intel, a fost de bun augur.

Un punct de referinta a fost reprezentat de introducerea arhitecturii “OnNow” care cauta sa definesca o abordare globala a problemelor legate de configurarea sisitemului si a dispozitivelor cu care acesta interactioneaza. Un prim rezultat al acestei initiative a fost

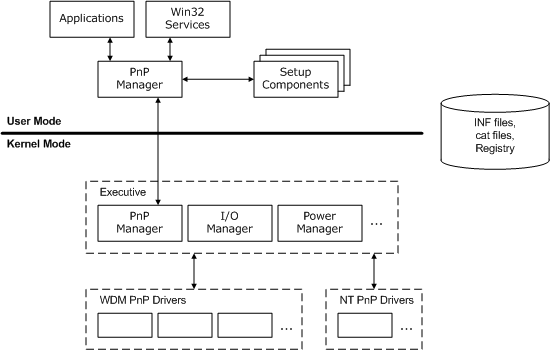
implementarea interfetei ACPI "Advanced Configuration and Power Interface" care defineste un nou sistem gestiune al resurselor, atat din punct de vedere al accesului si performantei, dar mai ales din punctul de vedere al consumului de energie. Prin aceste schimbari Plug and play-ul capata noi capabilitati cum ar fi: managementul consumului de curent si noi posibilitati de configurare, toate sub controlul total al sistemului de operare. Performante remarcabile au fost obtinute in domeniul notebook-urilor, remarcandu-se astfel o crestere a autonomiei dispozitivelor prin folosirea facilitatilor noi introduse.

Incepand cu Windows 2000 sistemul care pana atunci purta numele de Plug and Play se va numi Windows Driver Model (WDM). Noul sistem de operare Windows 8 aduce si mai multe imbunatatiri, introducand serviciul UpnP (Universal Plug and Play), care contine o colectie de drivere care permit controlarea unor dispozitive precum instalatii de aer conditionat controlabile peste IP.

## Componentele PnP

In Windows (incepand cu Win2000), implementarea plug and play are mai multecomponente software:

1. Managerul PnP - are doua parti : manager PnP kernel-mode si manager PnP user-mode. Managerul PnP kernel-mode interactioneaza cu componentele sistemului de operare si drivere pentru a configura , gestiona si mentine dispozitive . Managerul PnP user-mode interactioneaza cu componente de configurare user-mode , cum ar fi instalatori de clasa pentru a configura si instala dispozitive. De asemenea, managerul PnP user-mode interactioneaza cu aplicatii pentru, de exemplu, inregistrarea unei cereri de notificare a modificarilor unui dispozitiv .
2. Driverele PnP - suporta dispozitivele fizice , logice si virtuale pe o masina. Termenul “driver PnP” se refera la orice driver Windows care accepta interfetele descrise in acesta sectiune. In timp ce majoritatea driver-elor PnP sunt, de asemenea, drivere WDM si , astfel, surse compatibile pe platformele Windows, doar cateva drivere suporta PnP fara implementarea deplina WDM.
3. Managerul de consum (power manager) - se ocupa cu managementul consumului (pentru a reduce consumul de energie al sistemului, anumite dispozitive pot fi trecute in stari speciale de economisire a energiei).
4. Registrii (registry) - contin o baza de date a componentelor hardware si software instalate in sistem si sunt folositi la identificarea si localizarea resurselor de catre dispozitive.
5. Fisierele .inf (INF file) - descriu un dispozitiv, fiind necesar cate un astfel de fisier pentru fiecare dispozitiv la instalarea driver-ului; fiecare pereche dispozitiv/driver trebuie sa aiba un astfel de fisier.



**Fig1**. Componente PnP

## Functionare PnP in Windows

In Windows (incepand cu Win 95) kernel-ul (sau portiuni componente ale acestuia) se ocupa de configurarea dispozitivelor si in acest sens este cu adevarat un sistem plug and play. Dispozitivele sunt descoperite automat in timpul secventei de boot sau la inserare (hotplug), determinand incarcarea automata a driver-elor corespunzatoare. De cele mai multe ori, driverele disponibile sub Windows sunt dezvoltate de catre producatorul echipamentului si ofera performante optime. Tot datorita driverelor specializate, consumul de energie este mai bine optimizat in Windows. Arhitectura OnNow impune o serie de interfete stricte care permit controlarea setarilor dispozitivelor de tip plug-and-play intr-un detaliu mult mai fin.

Ca si in cazul Linux-ului, in Windows (versiunile de dupa Windows 95) se aplica un algoritm de rezolvare a conflictelor ce apar la alocarea de resurse (ex. conflicte de IRQ) . In modelul anterior din Windows (legacy drivers), era necesara incarcarea explicita a driver-elor si initializarea dispozitivelor asociate acestuia la incarcare . Folosind plug and play, acest lucru nu mai este necesar, intrucat sistemul de operare se ocupa de aceste operatii (la detectarea unui dispozitiv se va apela o metoda speciala a driver-ului care va adauga dispozitivul).

# Windows driver model (WDM)

## Introducere

Pentru a permite dezvoltatorilor driver-elor sa scrie drivere de dispozitiv cu codul sursa compatibil cu toate sistemele de operare Microsoft Windows, a fost introdus Windows Driver Model (WDM). Driver- ele kernel-mode, care respectă normele de WDM se numesc drivere WDM. Toate driverele WDM trebuie:

• Sa includa Wdm.h, nu Ntddk.h. (Wdm.h este un subset de Ntddk.h.)

• Sa aiba unul din tipurile de drivere WDM (bus driver, function driver, filter driver) si sa creeze dispozitive cu unul din tipurile WDM (Physical Device Object, Functional Device Object, Filter Devive Object).

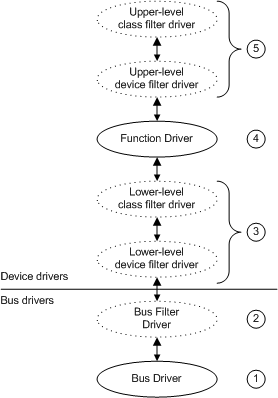
• Sa suporte Plug and Play (PnP);

• Sa suporte managementul consumului (power management);

• Sa suporte Windows Management Instrumentation (WMI); WMI este un mecanism prin care kernel-ul pune la dispozitia aplicaţiilor din user-mode informaţii (permite publicarea informaţiilor, configurarea dispozitivelor, un mecanism de notificări, logarea evenimentelor, etc.).

## Tipuri de WDM

Windows Driver Model organizeaza driverele si dispozitivele intr-o stiva . Astfel , exista trei categorii de drivere: Bus driver, function driver, filter driver.



**Fig.2** Relatia dintre bus driver, function driver si filter driver pentru un dispozitiv

### Bus driver

Bus Drivers – reprezinta drivere asociate magistralelor din sistem. Microsoft ofera bus drivers pentru cele mai comune magistrale cum ar fi PCI, PnpISA, SCSI si USB. Este obligatoriu sa existe un astfel de driver pentru fiecare magistrala din sistem. Un bus driver poate deservi mai multe magistrale daca exista mai mult de o magistrala de acelasi tip pe masina .

Responsabilitatile principale ale unui bus driver sunt:

• Enumerarea dispozitivelor de pe magistrala acestuia;

• Sa raspunda la IRP-urile Plug and Play si IRP-urile gestionarii consumului;

• Multiplexarea accesului la magistrala;

• Administrarea generica a dispozitivelor pe magistrala;

Un bus driver efectueaza anumite operatiuni in numele dispozitivelor de pe magistrala , incluzand accesarea registrilor dispozitivului pentru a schimba fizic starea de consum a unui dispozitiv. De exemplu , atunci cand dispozitivul intra in stand-by , bus driver-ul seteaza registrii dispozitivului sa puna dispozitivul intr-o stare corespunzatoare de consum.

### Function driver

Function Drivers – reprezinta principalul driver pentru un dispozitiv . Un function driver este de obicei scris de catre producatorul dispozitivului si este necesar. Managerul PnP incarca cel putin un function driver per dispozitiv. Un function driver poate deservi unul sau mai multe dispozitive.

Un function driver asigura interfata operationala pentru dispozitivul sau.

• Class drivers – sunt un tip de function drivers care asigura interfEtele intre diferite nivele ale arhitecturii WDM . Functionalitatea comuna intre diferitele clase de drivere pot fi scrise in class driver si utilizate de catre alte class drivers si drivere miniport. Marginea inferioara a class driverului va avea interfata expusa la driverul miniport , in timp ce marginea superioara a class driverelor de nivel inalt este sistemul de operare specific. Driverele pot fi incarcate dinamic si descarcate dupa bunul plac.

• Miniport drivers – sunt de asemenea function drivers pentru USB, SCSI, audio si adaptori de retea. Acestea ar trebui sa fie de obicei sursa si binar compatibile intre Windows 98 si Windows 2000 si sunt specifice hardware dar controleaza accesul la hardware printr-un class driver magistrala specific.

### Filter driver

Filter Drivers - sunt drivere optionale care adauga o valoare sau modifica comportamentul unui dispozitiv . Un filter driver poate deservi unul sau mai multe dispozitive .

• **Bus filter driver** – de obicei adauga o valoare la magistrala si sunt furnizate de Microsoft. Acestea sunt optionale si pot fi oricate bus filter drivers pentru o magistrala . Pentru dispozitivele descrise de un BIOS ACPI , managerul de consum introduce un filtru ACPI(bus filter driver) furnizat de Microsoft deasupra bus driver-ului pentru fiecare astfel de dispozitiv. Filtru ACPI porneste si opreste dispozitivele ,este transparent pentru alte drivere dar nu este prezent la masini non-ACPI .

• **Lower-Level Filter Drivers** – de obicei modifica comportamentul dispozitivelor hardware. Pot fi oricate filter drivers de nivel jos pentru un dispozitiv .

• **Upper-Level Filter Drivers** – adauga valoare pentru un anumit dispozitiv. De exemplu , un upper level filter driver pentru o tastatura ar putea impune controale suplimentare de securitate . Pot fi oricate drivere de aceste tip pentru un dispozitiv .

# Implementare Plug and Play in Windows

## 

## Initializarea driver-ului (Driver\_Entry)

Driverele WDM se initializeaza in rutina DriverEntry. Functiile ce trebuiesc initializate reprezinta functiile de dispatch pentru operatii de deschidere, scriere, citire, control, inchidere dispozitiv. Pe langa acestea, mai trebuie initializata functia pentru initializarea dispozitivelor (AddDevice) si functia pentru mesajele plug and play (IRP\_MJ\_PNP).

## Initializarea dispozitivului (AddDevice)

Functia AddDevice este apelata de managerul PnP atunci cand se descopera un dispozitiv. Prototipul functie AddDevice este:

NTSTATUS AddDevice(

IN PDRIVER\_OBJECT DriverObject,

IN PDEVICE\_OBJECT PhysicalDeviceObject

);

DriverObject este un pointer catre obiectul asociat driver-ului, iar PhysicalDeviceObject este un pointer catre dispozitivul fizic

## Pornirea dispozitivului (IRP\_MN\_START\_DEVICE)

Managerul PnP trimite o cerere IRP\_MN\_START\_DEVICE in cazul in care se doreste pornirea unui dispozitiv nou adaugat sau se doreste repornirea unui dispozitiv existent, care a fost oprit pentru reechilibrarea resurselor.

Daca un driver pentru un dispozitiv nu reuseste cererea IRP\_MN\_START\_DEVICE, managerul PnP trimite o solicitare IRP\_MN\_REMOVE\_DEVICE catre stiva dispozitivelor. Ca raspuns la acest IRP, driverele de dispozitive anuleaza operatiunea de start si cea de AddDevice si il desprinde de stiva dispozitivelor.

Ca rezultat al acestui start nereusit, managerul PnP marcheaza dispozitivul cu “fail start”.

## Oprirea dispozitivului (IRP\_MN\_STOP\_DEVICE)

La primirea unui IRP cu codul minor IRP\_MN\_STOP\_DEVICE, se vor executa operatii pentru oprirea dispozitivului.

Managerul PnP directioneaza driverele sa opreasca un dispozitiv in urmatoarele situatii:

* Pentru a reechilibra resursele hardware folosite de dispozitiv. Reechilibrarea este de obicei necesara cand se introduce un dispozitiv nou care necesita resurse deja utilizate.
* Pentru a dezactiva dispozitivul ca raspuns la cererea managerului de dispozitive
* Dupa o cerere esuata de IRP\_MN\_STOP\_START

## Eliminarea dispozitivului (IRP\_MN\_REMOVE\_DEVICE)

Managerul PnP trimite driverelor cereri de eliminare a dispozitivelor cand dispozitivul a fost sau este in curs de scoatere din calculator. De asemenea, managerul PnP, trimite o solicitare de eliminare atunci cand utilizatorul doreste sa faca un update la driveri dispozitivului, sau atunci cand dispozitivul este dezactivat sau nu reuseste sa porneasca

# BIOS-ul Plug and Play

## Introducere Bios PnP

Rolul Bios-ului traditional este de a gestiona dispozitivele esentiale ale sistemului: hard disk, placa video, porturile seriale si paralele, tastatura. Celelalte dispozitive erau lasate sa se lupte pentru IRQ (cerere de intrerupere). Cand Windows 95 a fost introdus, rolul Bios-ului s-a schimbat dramatic. Pentru a sprijinii Windows 95 a fost introdus Plug and Play Bios care a schimbat modul in care sunt instalate si gestionate dispozitivele.

Bios-ul Plug and Play executa urmatorii pasi pentru configurarea dispozitivelor PnP inainte de a pornii masina:

* Construieste o structura numita harta resurselor pentru a determina resursele sistemului disponibile – IRQ’s, porturi I/O, canale DMA si ferestre de memorie.
* Harta resurselor este initializata sa indice resursele disponibile
* Bios-ul determina resursele folosite de dispozitivele de pe placa de baza, fie prin cautarea informatiilor in ESCD (Extended System Configuration Data), fie prin verificarea resurselor utilizate de dispozitivele de pe placa de baza
* Bios-ul PnP interogeaza ESCD pentru a determina resursele care sunt alocate pentru cardurile Legacy, apoi actualizeaza harta resurselor. In aceasta etapa, harta resurselor contine toate informatiile necesare cu privire la dispozitivele de tip Legacy.
* Cardurile ISA PnP sunt examinate pentru a stabilii resursele necesare.
* Dupa identificare se porneste cu ultima configuratie care a mers pentru fiecare card din ESCD. Pentru card-uri noi, sistemul va folosi prima configuratie alternativa care se potriveste cu cerintele de resurse ale card-ului. Cu toate acestea, daca nici unul nu este gasit, se incearca urmatorul set de resurse disponibile. Daca sistemul ramane fara resurse, se va reveni la card-urile configurate anterior si se incearca o re-configurare a lor pentru a face loc card-ului nou.
* Apoi Bios-ul scaneaza extinderea memoriei standard a Bios-ului pentru a cauta extinderi ale semnaturii Bios pentru Legacy ISA si pentru card-uri ISA PnP.
* Spatiul de configurare PCI este examinat si dispozitivele PCI initializate. Deoarece dispozitivele PCI prin definitie sunt complet configurabile, aceste dispozitive sunt configurate la modul primul venit, primul servit.
* In cazul in care unul sau mai multe card-uri PCI nu pot fi configurate, o eroare este generata si sarcina de configurare este preluata de managerul de configurare.

Unele dintre cele mai importante domenii abordate de Bios-ul sistemului sunt managementul resurselor si configuratia la rulare.

Managementul resurselor ofera posibilitatea de a gestiona resursele de sistem fundamentale, si anume, DMA(direct memory acces), IRQ(interrupt request line), I/O si adresele de memorie. Aceste resurse, denumite si resurse ale sistemului sunt in cerere mare si de obicei sunt supra alocate sau alocate intr-o maniera conflictuala cu arhitectura standard a sistemului, ceea de duce de obicei la imposibilitatea bootarii sau la esecuri de configuratie ale sistemului. Un sistem de tipul plug and play Bios joaca un rol vital in gestionarea acestor resurse si asigura o lansare de succes a sistemului de operare. Bios-ul Plug and Play configureaza dispozitivele placii de baza in timpul procesului de pornire. Dupa ce procesul POST (power-on self test) s-a incheiat, controlul configurarii dispozitivelor plug and play trece din sistemul Bios in sistemul software. Bios-ul, cu toate acestea, ofera servicii de configurare pentru dispozitivele placii de baza chiar daca procesul POST s-a terminat. Aceste servicii se numesc runtime services (servicii la rulare). Configuratia la rulare este un concept ce nu a mai existant intr-un sistem de Bios pana la introducerea in Bios-ul Plug and Play. Bios-ul sistemul nu a mai avut capacitatea de a modifica intr-un mod dinamic resursele alocate dispozitivelor placii de baza dupa ce sistemul de operare a fost incarcat.

## Obiectivele implementarii PnP in Bios

Bios-ul Plug and Play are urmatoarele obiective:

* **Eliminarea conflictelor de resurse din timpul procedurii de POST**

O problema majora este ca exista mai multe dispozitive disponibile decat exista resurse de sistem, de aceea este foarte probabil sa existe conflicte ale resurselor. Bios-ul sistemului va preveni aparitia acestor conflicte prin nepornirea dispozitivelor care se afla in conflict cu dispozitivele de boot primar, sau prin relocarea dispozitivelor de boot, pentru a permite o pornite de succes a sistemului de operare.

* **Permiterea configurarii dinamice a dispozitivelor placii de baza**

Dispozitivele placii de baza au fost tratate ca avand configuratii statice. Este obiectivul Bios Plug and Play sa ofere un mecanism standard prin care dispozitivele placii de baza sa fie configurate dinamic de software-ul sistemului. Acest lucru va acorda pentru sistemului software o mare flexibilitate a managementului configurarii atunci cand resursele sistemului din folosite si configuratii alternative sunt necesare.

* **Sa furnizeze notificari ale sistemului**

Bios-ul sistemului este capabil de a detecta anumite evenimente hardware ce ar putea afecta configurarea sistemul. Prin furnizarea unui mecanism de notificare, sistemul de operare poate recunoaste evenimentul si sa opereze toate modificarile de configuratie necesare.

* **Independenta intre hardware si sistemul de operare**

Extensile ale Bios-ului sistemului izoleaza partea de hardware a placii de baza prin interfete si structuri bine definite. Nodurile dispozitivelor de sistem reprezinta dispozitive care sunt controlate de Bios-ul sistemului. Sistemul de operare nu necesita cunostinte specifice ale placii de baza in vederea controlarii acestor dispozitive ci mai degraba se baseaza pe Bios-ul sistemului pentru al izola de hardware-ul care sta la baza lui.

# Concluzii

Plug and Play a evoluat foarte mult cu siguranta de la versiunea Windows 95. Cuplat cu interfata sistemului ACPI, putem sa-l utilizam pentru a gestiona dinamic toate componentele care alcatuiesc calculatorul. Acest lucru ne permite sa uitam ce IRQ sau ce driver sa utilizat si ne ofera mai mult timp pentru a folosi calculatorul pentru lucrurile pentru care l-am achizitionat initial.

Pentru Windows cele mai multe drivere sunt furnizate de producatorii componentelor respective, si din cauza aceasta uneori functia Windows-ului de Plug and Play nu poate fi folosita, fiind necesar sa introducem noi driverul necesar functionarii componentei.

In lucrarea de fata, am incercat sa oferim o imagine si o intelegere cat mai ampla asupra conceptului si asupra mecanismului de functionare al Plug and Play Windows dar si modul de functionare al acestuia in Bios-ul sistemului.

# Blibliografie

* Andrew S. Tanenbaum, Sisteme de operare moderne, 3'rt ed., Byblos, 2004
* Programming the Microsoft Windows Driver Model, Second Edition
* <http://www.quepublishing.com/articles/article.aspx?p=27188&seqNum=8>
* <http://www.geocities.com/~budallen/pnp.html>
* <http://pvtridvs.net/pool/miscbooks/Programming_with_Plugplay.pdf>
* <http://msdn.microsoft.com/en-us/windows/hardware/gg463317#ESAAC>
* <http://msdn.microsoft.com/en-us/library/windows/hardware/ff547125(v=vs.85).aspx>
* <http://msdn.microsoft.com/en-us/library/windows/hardware/ff548158(v=vs.85).aspx>
* <http://msdn.microsoft.com/en-us/library/windows/hardware/ff564862(v=vs.85).aspx>
* <http://en.wikipedia.org/wiki/Legacy_Plug_and_Play>
* <http://en.wikipedia.org/wiki/Plug_and_play>
* <http://www.osronline.com/article.cfm?article=61>
* <http://tldp.org/HOWTO/Plug-and-Play-HOWTO.html>
* <http://www.computerhope.com/jargon/p/pnp.htm>