

Ovladač LPT PcKit

OVLADAČ PRO PRÁCI S PORTY A
PŘERUŠENÍM PRO APLIKACE URČENÉ
PRO DOS A ROZHRANÍ WIN32

Příručka uživatele a programátora



SofCon[®] spol. s r.o.
Střešovická 49
162 00 Praha 6
tel/fax: +420 220 180 454
E-mail: sofcon@sofcon.cz
www: <http://www.sofcon.cz>

Informace v tomto dokumentu byly pečlivě zkontrolovány a SofCon věří, že jsou spolehlivé, přesto SofCon nenese odpovědnost za případné nepřesnosti nebo nesprávnosti zde uvedených informací.

SofCon negarantuje bezchybnost tohoto dokumentu ani programového vybavení, které je v tomto dokumentu popsáno. Uživatel přebírá informace z tohoto dokumentu a odpovídající programové vybavení ve stavu, jak byly vytvořeny a sám je povinen provést validaci bezchybnosti produktu, který s použitím zde popsaného programového vybavení vytvořil.

SofCon si vyhrazuje právo změny obsahu tohoto dokumentu bez předchozího oznámení a nenese žádnou odpovědnost za důsledky, které z toho mohou vyplynout pro uživatele.

Datum vydání: 22.05.2003

Datum posledního uložení dokumentu: 22.05.2003

(Datum vydání a posledního uložení dokumentu musí být stejné)

Upozornění:

V dokumentu použité názvy výrobků, firem apod. mohou být ochrannými známkami nebo registrovanými ochrannými známkami příslušných vlastníků.

Obsah :

1.O dokumentu	5
1.1. Revize dokumentu	5
1.2. Účel dokumentu	5
1.3. Rozsah platnosti	5
1.4. Související dokumenty	5
2.Termíny a definice	5
3.Úvod	6
4.Instalace	6
4.1. Co dělat pokud zařízení nefunguje	7
5.Umístění souborů ovladače	7
6.Nastavování vlastností ovladače	7
6.1. Zákaz a povolení ovladače	7
6.2. Zjištění stavu připojeného zařízení PcKIT	10
6.2.1. Zjištění stavu zařízení pomocí stránky stavu LPT PcKit	10
6.2.2. Zjištění stavu zařízení pomocí systémového protokolu	12
6.3. Popis zdrojů ovladače	13
6.4. Popis nastavení zdrojů ovladače	14
6.5. Konfigurace a registry	14
6.5.1. Přidání zdroje	15
6.5.2. Odebrání zdroje	15
6.5.3. Změna nastavení zdroje	15
6.6. Systémové prostředky	15
7.Rychlost zápisu na porty	16
8.Příloha	17
8.1. Stručně o operačním systému	17
8.2. Uživatelská a systémová úroveň běhu programu	17
8.3. Rozhraní operačního systému	17
8.4. Paměťový podsystém	18
8.5. Aplikace pro WIN32	19
8.5.1. Použití knihovny HalPcKit	19
8.5.2. Použití funkcí rozhraní WIN32	21
8.6. Aplikace pro DOS	22
8.7. Popis VDM	23
8.7.1. Obecně	23
8.7.2. Uživatelská zařízení	23
8.8. Adresáře a registry týkajících se ovladačů	24
8.8.1. Popis adresářů	24
8.8.2. Popis registru	24
9.Literatura	29

1. O dokumentu

1.1. Revize dokumentu

Verze dokumentu	Verze SW	Autor	Datum vydání	Popis změn
1.00	1.XX	Hv		První vydání.
1.13	1.XX	Tu	22.05.2003	Úprava dokumentu dle ISO9000.

1.2. Účel dokumentu

Tento dokument slouží jako popis instalace a použití ovladače pro operační systém Windows 2000 a Windows XP, umožňujícího pracovat s porty a přerušením (například při vývoji aplikací s pomocí desky PcKIT) podobně jako v prostředí DOS.

1.3. Rozsah platnosti

Určen pro programátory a uživatele programového vybavení SofCon.

1.4. Související dokumenty

Pro čtení tohoto dokumentu není potřeba číst žádný další manuál.

2. Termíny a definice

Používané termíny a definice jsou popsány v samostatném dokumentu Termíny a definice.

3. Úvod

V současnosti většina operačních systémů spuštěných na 32b procesorech fy. Intel využívá chráněný režim. Přestože tento režim má oproti reálnému režimu několik specifických vlastností, bude se dále uvažovat jenom vykonávání programu na různých úrovních a problémy při přístupu k portům. Toto chování platí především pro operační systémy Windows používající platformu Windows NT, počínaje operačními systémy Windows 4.0 a konče Windows XP.

Systémy s touto platformou se od zbývajících systémů fy. Microsoft liší lepší stabilitou systému, souborovým systémem NTFS a zavedením přístupových práv k jednotlivým částem operačního systému (objekty jádra, soubory a paměť). Proto každá aplikace, která bude chtít pracovat s porty, bude muset mít v jádru operačního systému službu (zpravidla ovladač), která tyto zápisy bude provádět. Komplikovanější situace nastává, pokud na porty bude chtít zapisovat aplikace napsaná pro dosové prostředí (např. Borland Pascal 7.0).

Protože naše firma používá při vývoji aplikací desku *PcKit* a vývojové prostředí *Borland Pascal 7.0*, vyvinuli jsme ovladač, který Vám umožňuje v operačních systémech Windows 2000 a Windows XP pracovat s porty a přerušením podobným způsobem, jak jste byli zvyklí dosud.

Pro aplikace určené pro dosové prostředí se program nemění a veškeré operace spojené se zápisem na porty (instrukce IN, OUT) a přerušením se provádí skrytě. Pro správnou činnost se musí pouze nastavit vlastnosti ovladače ve správci zařízení.

Pro aplikace určené pro rozhraní Win32 se musí použít dodávaná knihovna, která Vám poskytne řadu funkcí umožňujících pracovat s porty (instrukce IN, OUT).

Při používání driveru mohou nastat situace, které překvapí i zkušeného programátora. Proto je k manuálu přiložena příloha, která se bude věnovat popisu činnosti ovladačů, aby uživatel věděl, jak se má zachovat případně co všechno ovladač ovlivňuje. V této příloze budou také vysvětleny některé termíny, které nejsou obecně známy a jsou použity v manuálu k ovladači LPT PcKit.

4. Instalace

Při instalaci zařízení musíte být přihlášen jako uživatel, který má oprávnění instalovat zařízení – zpravidla musíte být členem skupiny administrátor.

Po spuštění programu *Setup.exe* se překopírují všechny soubory z instalačního média do dočasného adresáře určeného proměnnou TEMP (např. „c:\Documents and Settings\\Local Settings\Temp“). Vlastní instalace se bude provádět z dočasného adresáře a bude používat pomocný soubor *inf.txt*. Pokud při kopírování a vytvoření souboru nedošlo k chybě, spustí se instalace zařízení. Po jejím ukončení se dočasný adresář smaže. (Pozn.: Zadáte-li při spuštění programu *Setup.exe* parametr */d* tak se dočasný adresář nesmaže.)

Při instalaci je vytvořena nová třída zařízení (ovladačů), která se později zobrazuje ve správci zařízení a na konci instalace je instalované zařízení spuštěno s default parametry. Vzhledem k tomu, že při standardní instalaci pomocí správce zařízení (pomocí souboru INF) se zařízení nejdříve přiřadí prostředky a poté se vlastní zařízení spustí, nelze tuto instalaci použít pro ovladač LPT PcKit. Ovladač totiž vyžaduje, aby byl nejdříve spuštěn a poté si mohl dle nastavení vlastností vybrat používané systémové prostředky. **Z tohoto důvodu musíte při instalaci použít soubor *Setup.exe*.**

Po ukončení instalace může dojít k situaci, kdy je u zařízení žlutý vykřičník. Tento symbol znamená, že se ovladač pokusil získat systémový prostředek, který je používán jiným zařízením. V tomto případě musíte změnit nastavení u jednoho z kolidujících zařízení.

Pozn.: Implicitně používané prostředky ovladače lze před instalací měnit na instalačním médiu v souboru *tmp.txt* v sekcích [LptPcKit_Hw_AddReg] a [LptPcKit.LC]. V sekci [LptPcKit_Hw_AddReg] by vždy měla být podmnožina prostředků definovaných v [LptPcKit.LC]. V případě nevhodného nastavení nelze ovladač spustit, případně může dojít k pádu celého systému.

4.1. Co dělat pokud zařízení nefunguje

Pokud instalace zařízení proběhla bez chyby a ve správci zařízení není hlášen žádný konflikt zařízení (žlutý vykřičník) případně zařízení LPT Pckit je bez chybně spuštěno (detekce proběhne bez chyby – viz. obrázek 6: Stav připojeného zařízení LPT Pckit), lze tuto kapitolu přeskočit.

1. *Ve správci zařízení je u zařízení žlutý vykřičník.*

Provést kontrolu přidělených prostředků. Změna používaných zdrojů zařízení je popsána v Popis zdrojů ovladače.

2. *Zařízení je spuštěno, ale zařízení nelze detekovat (viz. obrázek 6: Stav připojeného zařízení LPT Pckit a kapitola Zjištění stavu připojeného zařízení Pckit)*

V případě problémů s alokací přerušení, nastavení portu LPT nebo detekce zařízení provést kontrolu nastavení LPT portu. V Setup BIOS musí být nastaveny následující kombinace buď EPP (standard 1.7 nebo 1.9) nebo ECP + EPP. (U některých základních desek, dle chipset, lze případně nastavit ECP). Dále se musí ve správci zařízení povolit používání přerušení viz. obrázek 8: Nastavení používání IRQ pro LPT port.

V případě problémů s alokací portu LPT provést kontrolu všech zařízení (služeb) v systému, případně neznámé zařízení (služby) vypnout..

V případě problémů s detekcí zařízení provést kontrolu zasunutí LPT Pckit do portu LPT.

5. Umístění souborů ovladače

Protože pro zpřístupnění portů jak pro rozhraní WIN32 tak VDM je potřeba mít vždy dva soubory, jeden pro systémovou úroveň a druhý pro rozhraní dle dané aplikace, je ovladač rozdělen do několika souborů. Při instalaci jsou tyto soubory uloženy v adresáři „%SystemRoot%\system32\drivers\sofcon”.

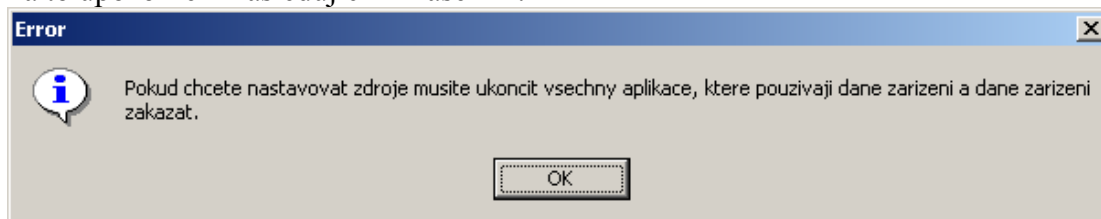
6. Nastavování vlastností ovladače

6.1. Zákaz a povolení ovladače

Při nastavování prostředků ovladače musíte být přihlášen jako uživatel, který má oprávnění nastavovat vlastnosti zařízení – zpravidla musíte být členem skupiny

administrátor.

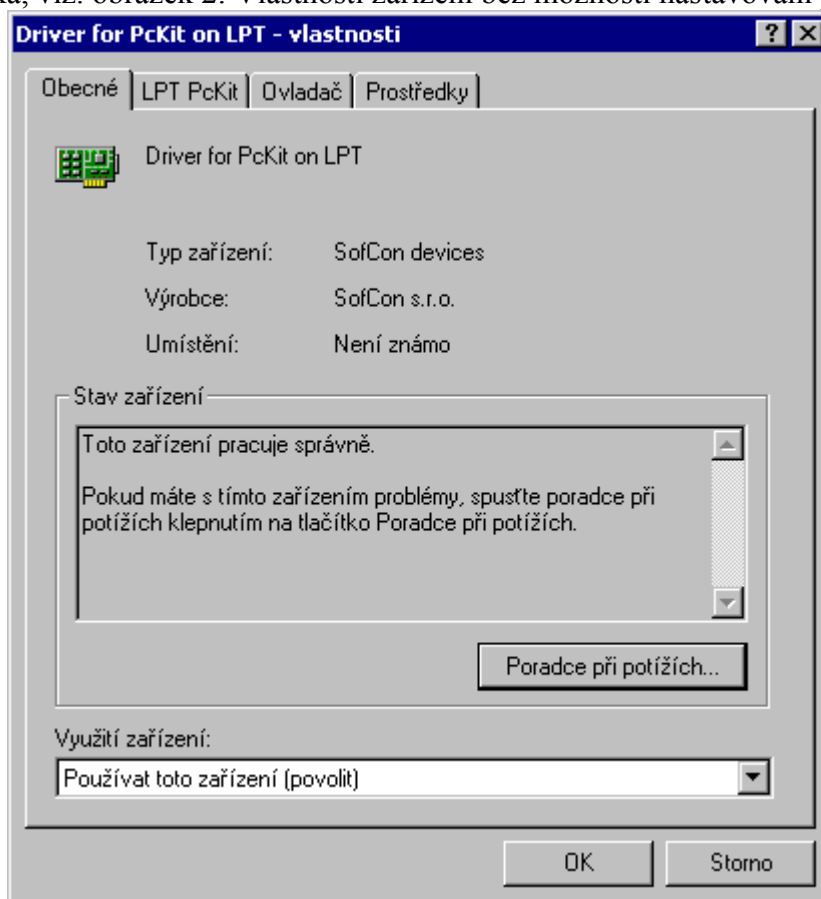
Nastavení prostředků můžete provést pouze u zakázaného (vypnutého) zařízení. Pokud byste chtěli měnit nastavení prostředků u aktivní (zapnuté) služby, jste na to upozorněni následujícím hlášením.



obrázek 1: Chybové hlášení při spuštění zařízení

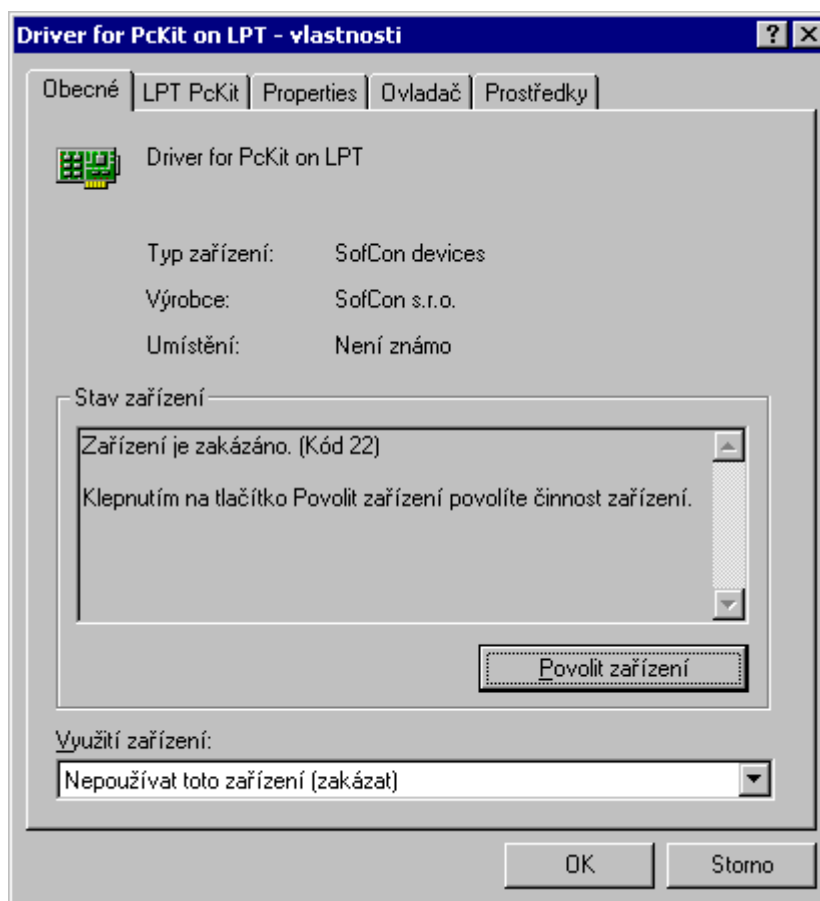
Zákaz zařízení provedete výběrem zařízení ve správci zařízení a v pop-up menu, který se zobrazí po stisknutí pravého tlačítka myši, volbou položky „Zakázat“.

Je-li u ikony umístěn žlutý vykřičník, je zařízení sice vypnuto, ale správce zařízení Vám nemusí nabízet položku „Povolit“. V tomto případě se musí nejprve vybrat položka „Zakázat“, poté nastavit nové vlastnosti a následně zařízení „Povolit“. Při zakázaném zařízení v pop-up menu chybí položka „Zakázat“ a na jejím místě je položka „Povolit“. Pokud zařízení není zakázáno zobrazí se Vám následující obrazovka, viz. obrázek 2: Vlastnosti zařízení bez možnosti nastavování prostředků.



obrázek 2: Vlastnosti zařízení bez možnosti nastavování prostředků

V případě, že zařízení není spuštěné nebo je zakázané, zobrazí se následující obrazovka, viz. obrázek 3: Vlastnosti zařízení s možností nastavování prostředků.



obrázek 3: Vlastnosti zařízení s možností nastavování prostředků

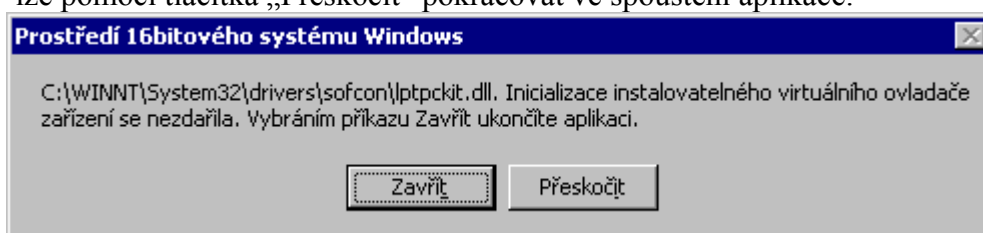
Tipy pro nastavování:

1. Pokud máte v systému nainstalován nestabilní nebo neověřený ovladač, je možné, že Vám systém při povolení zařízení bude padat.
2. Pro snadnější práci s ovladačem doporučuji zařízení v každém případě zakázat. Tzn. v pop-up menu se Vám jako první položka zobrazuje „Povolit“.
3. Systém Vás bude při každé změně nastavení prostředků vyzývat, abyste systém restartovali. Tento restart není zpravidla nutný, pokud měníte nastavení prostředků zakázaného (červený křížek) nebo nespuštěného (žlutý vykřičník) zařízení. V případě, že zařízení je používáno, tak ho nelze vypnout a musíte nejdříve aplikaci ukončit a poté se pokusit zařízení znovu zakázat. Nepodaří-li se zařízení ani poté vypnout, musíte počítač restartovat. Systém po každé změně nastavení prostředků provede inicializaci všech zařízení a jejich novou alokaci jednotlivým zařízením. V případě ovladače LPT PcKit se zapnutou podporou dosových aplikací nesmí být spuštěna žádná dosová úloha, protože by se změny nastavení do této aplikace nepromítly.
4. Pokud je zařízení ve vypnutém stavu, přes ikonu má červený křížek, nealokuje žádné systémové zdroje, ale je možné, že zůstane zapnutá inicializace dosových ovladačů. V tomto případě se bude zobrazovat chybové hlášení, viz obrázek 4: Chybové hlášení při spuštění dosové aplikace.



obrázek 4: Chybové hlášení při spuštění dosové aplikace

Po stisku tlačítka „OK“ se zobrazí dialogové okno, viz. obrázek 5: Dialog při chybě v inicializaci ovladače, které Vám umožní spuštění dosové aplikace zastavit pomocí tlačítka „Zavřít“. V případě, že spuštěná aplikace nepotřebuje obsluhu systémových prostředků případně virtuálních zařízení, lze pomocí tlačítka „Přeskočit“ pokračovat ve spuštění aplikace.



obrázek 5: Dialog při chybě v inicializaci ovladače

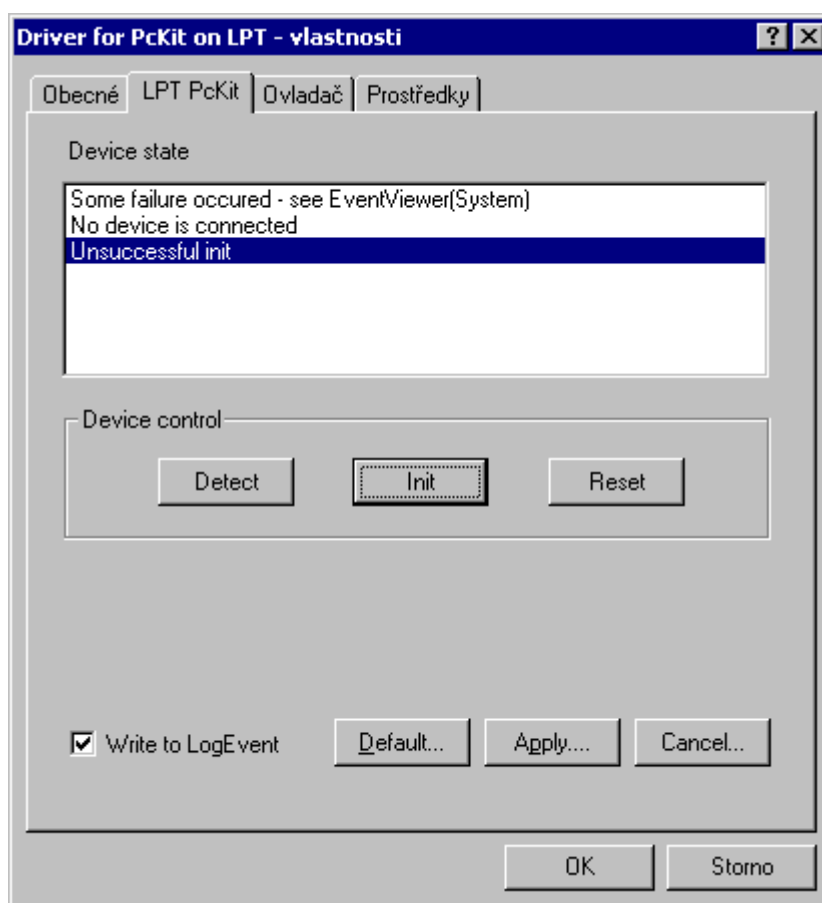
5. Podle nastavení ovladače se mohou tato chybová hlášení zobrazit několikrát pro různé systémové prostředky a virtuální zařízení.
Vypnutí inicializace ovladačů se provede v kapitole Popis nastavení zdrojů ovladače.
6. Pokud je spuštěno více jak 5 DOS aplikací zobrazí se předchozí dialog. Toto chování je dáno ovladačem, který generuje přerušení pro DOS aplikace.

6.2. Zjištění stavu připojeného zařízení PcKIT

6.2.1. Zjištění stavu zařízení pomocí stránky stavu LPT PcKit

Před popisem zjišťování stavu zařízení bych se ještě rád zastavil u používané terminologie. Dosud byl vždy pod pojmem zařízení popisován ovladač, který vytvořil funkční instanci nazývanou zařízení. Tato instance vždy před spuštěním provedla patřičnou detekci obsluhovaného hardwaru a poté se dokončila její inicializace. Vzhledem k možným stavům, v kterých se může PcKIT připojený na LPT nacházet, je rozlišováno mezi zařízení (instance ovladače) a připojeným zařízením (obsluhovaný hardware).

Zavedením tohoto rozlišení může nastávat situace, kdy ovladač je spuštěn aniž by v systému bylo připojené zařízení. Proto ovladač umožňuje zjišťovat stav připojeného zařízení na stránce LPT PcKit, viz. obrázek 6: Stav připojeného zařízení LPT PcKit.



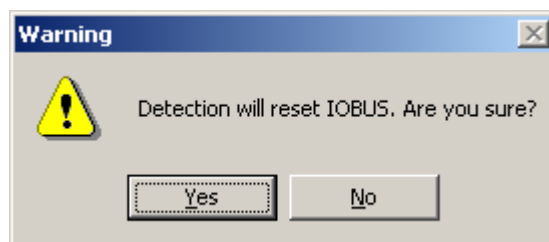
obrázek 6: Stav připojeného zařízení LPT PcKit

Stránka LPT PcKit je rozdělena do dvou částí. Horní část, okno *Device state*, a tlačítka *Detect*, *Init* a *Reset*, slouží pro zjišťování okamžitého stavu zařízení. Spodní část, zatrhávací tlačítko *Write to LogEvent* a tlačítka *Default*, *Apply* a *Cancel*, slouží pro nastavování specifických vlastností zařízení.

V další části bude uveden popis jednotlivých prvků této stránky. (Pozn.: Při RESET sběrnice IOBUS dochází také k RESET připojeného zařízení (LPT PcKit), jenž je v případě stisku tlačítka *Detect* následován dalšími operacemi. Jejich výsledkem je nastavení připojeného zařízení (LPT PcKit) do stavu pro vykonávání operací na sběrnici IOBUS.)

Device state je okno pro zobrazování výsledného stavu provedených operací.

Detect – stiskem tohoto tlačítka způsobíte RESET celé sběrnice IOBUS, proto byste toto tlačítko měli používat s opatrností. Před spuštěním detekce se zobrazí dialog, kterým tuto akci musíte potvrdit. (Pozn.: V rámci detekce se provede inicializace připojeného zařízení.). Výsledkem této operace je rozsvícení LED na zařízení LPT PcKit.



obrázek 7: Potvrzení detekce zařízení a RESETu sběrnice IOBUS

Init – stiskem tlačítka se provede inicializace připojeného zařízení. Výsledkem této operace je rozsvícení LED na zařízení LPT PcKit.

Reset – stiskem tlačítka se provede RESET připojeného zařízení a sběrnice IOBUS. Pro správnou činnost připojeného zařízení by se v následujícím kroku měla provést buď detekce nebo inicializace zařízení. Pokud se tento krok neprovede nebude připojené zařízení (LPT PcKit) pracovat se zařízením na sběrnici IOBUS. Výsledkem této operace je zhasnutí LED na zařízení LPT PcKit.

Zatrhávací tlačítko *Write to LogEvent* slouží pro aktivaci zápisu chybových stavů zařízení do souboru událostí (EventLog).

Apply... provede uložení změn v nastavení zařízení.

Cancel... veškeré změny v nastavení zařízení jsou zrušeny.

Default... nastaví a uloží implicitní stav nastavení zařízení.

6.2.2. Zjištění stavu zařízení pomocí systémového protokolu

Pokud je na stránce LPT PcKit, viz. obrázek 6: Stav připojeného zařízení LPT PcKit, zatrženo tlačítko *Write To EventLog* zapisuje zařízení chyby při spuštění nebo chodu zařízení do *Systémového protokolu (Systém Log)*. Při nezatrženém tlačítku se chyby do protokolu nezapisují.

V protokolu mohou být zaznamenány tyto události.

Device could not be initialized!

Zařízení se nepodařilo inicializovat v důsledku neplatně přiřazených prostředků nebo špatného zasunutí zařízení do portu LPT. Zakažte zařízení, zkontrolujte přiřazené prostředky a povolte zařízení. Zkontrolujte zasunutí zařízení do portu LPT.

Device could not be created - either device is already running or not enough memory.

Tato chyba je způsobena pokud je zařízení už v systému jednou spuštěno nebo je v systému nedostatek paměti. Pokud je zařízení odinstalováno a tato chyba se neustále zobrazuje, je potřeba restartovat systém.

Device could not acquire handle to LPT port - either other device is using LPT port or LPT driver is not running.

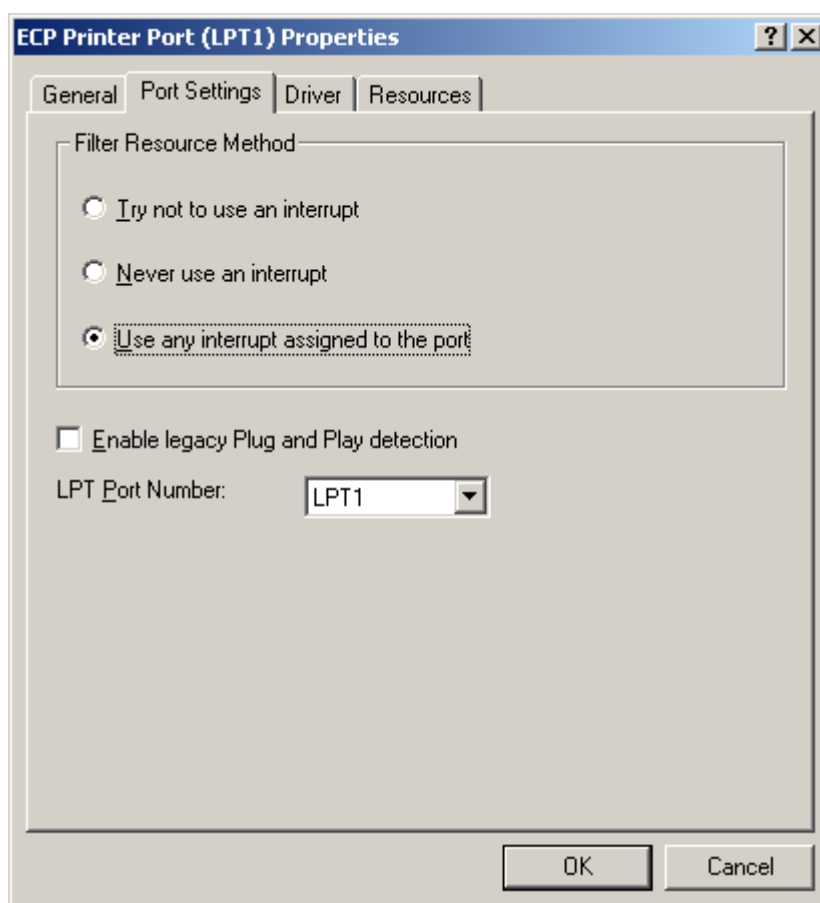
Tato chyba je způsobena pokud už v systému je spuštěno nějaké zařízení (služba) používající tento LPT port případně ovladač LPT portu je zakázán.

Device could not get information about LPT port.

Tato chyba je způsobena ovladačem LPT portu. V systému je nainstalována nekompatibilní verze.

Device could not allocate IRQ at LPT port. Check setting of LPT port whether IRQ is enabled for using.

Tato chyba je způsobena nastavením ovladače LPT portu neumožňujícím používání IRQ. Doporučené nastavení LPT portu je zobrazeno na následujícím obrázku, viz. obrázek 8: Nastavení používání IRQ pro LPT port. Dalším zdrojem problémů může být nastavení portu v Setupu BIOSu, viz. Co dělat pokud zařízení nefunguje.



obrázek 8: Nastavení používání IRQ pro LPT port

Device was not detected.

Při spuštění zařízení (instance ovladače) nebylo detekováno připojení LPT PcKit. Při této chybě by se mělo zkontrolovat připojení zařízení a pokusit se o detekci zařízení pomocí stránky LPT PcKit, viz. obrázek 6: Stav připojeného zařízení LPT PcKit. Dalším zdrojem problémů může být nastavení portu v Setupu BIOSu, viz. Co dělat pokud zařízení nefunguje.

6.3. Popis zdrojů ovladače

Ovladač umožňuje obsluhovat tři typy zařízení – IO Ports, Black Ports a přerušení.

IO Ports jsou porty, které má zařízení (instance ovladače) zaregistrováno v systému, tj. žádné další zařízení tyto porty nemůže používat.

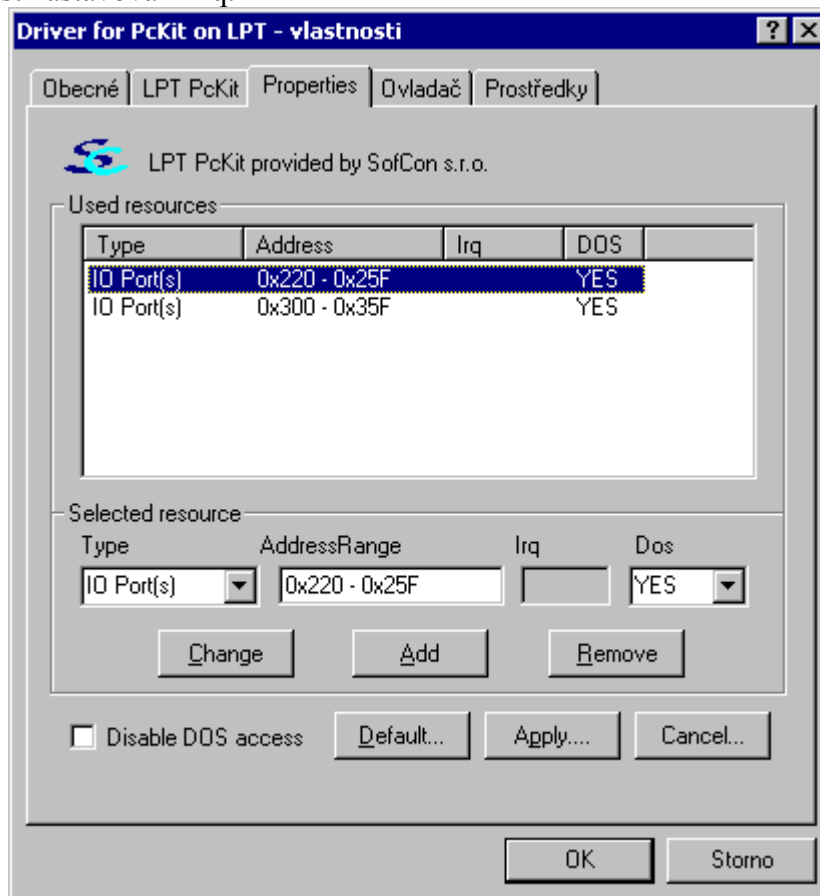
Black Ports jsou porty, které zařízení nemá v systému registrováno a jsou obsluhovány jinými zařízení. V případě nevhodného zápisu na tyto porty může dojít k pádu systému.

Přerušení. Vlastní připojené zařízení generuje pouze jedno přerušení podle nastavení portu LPT. Toto přerušení je zařízením zpracováno a podle nastavení připojeného zařízení se pro NT VDM vygeneruje jako IRQ5 nebo IRQ7. Tyto generované přerušení jsou přiřazena přerušením sběrnice IOBUS následujícím způsobem (**INT3iobus = IRQ7, INT4iobus = IRQ5**).

6.4. Popis nastavení zdrojů ovladače

Nastavování zdrojů se provádí na obrazovce, která se zobrazí po stisku záložky „Properties“.

Pozn.: Pro obsluhované zdroje ovladače LPT PcKit není v této verzi dostupná možnost nastavování Irq.



obrázek 9: Nastavování zdrojů ovladače

Pozn.: V operačních systémech Windows XP bylo zjištěno, že v dosových programech nelze používat obsluhu portů (IO Ports) na adresách 0x330 a 0x331. Tyto porty jsou automaticky obsluhovány dosovým podsystémem. V případě programů pro WIN32 lze tyto porty (IO Ports) používat bez omezení.

6.5. Konfigurace a registry

V této části bude popsáno jak obnovit implicitní nastavení, zrušit veškeré provedené změny, uložení konfigurace do registru a zákaz inicializace ovladače v prostředí dosových aplikací. Tyto činnosti se provádí pomocí zatrhávacího políčka a tlačítek, která jsou umístěna v dolní části.

Po stisku tlačítka „Apply...“ se současně zobrazená konfigurace (část Used resources) nastavení prostředků ovladače uloží do registru. Tato konfigurace se potom použije při spouštění zařízení při inicializaci systému nebo po jeho povolení.

Po stisku tlačítka „Default...“ se zobrazí implicitní nastavení zařízení pro použití desky LPT PcKit.

Po stisku tlačítka „Cancel...“ se přečte nastavení prostředků zařízení uloženého

v registru. Tzn. veškeré změny, které jste v dialogu provedli jsou zrušeny.

Zatrhávací políčko „*Disable DOS access*“ způsobí zrušení inicializace všech obsluh ovladače v dosových aplikacích. Stav tohoto tlačítka je zapsán do registru až po stisku tlačítka „*Apply...*“. Pokud byste změnilí stav tlačítka a poté stisknuli tlačítko „*Default...*“, přečte se z registru původní nastavení.

6.5.1. Přidání zdroje

Přidávání zdroje se provádí po nastavení všech jeho parametrů pomocí tlačítka „*Add*“ v části okna označené „*Selected resource*“. Po jeho stisku se v horní části okna označeného jako „*Used resources*“ tento zdroj vypíše.

Před nastavením parametrů zdroje v části označené jako „*Selected resource*“ je nejlepší nejdříve vybrat typ zdroje v combo-box „*Type*“. Změnou typu se mění možnosti zadávání obsluhovaných zdrojů dle následující tabulky.

Type	AddressRange	Irq	DOS
IOPort(s)	Zadává se interval	/nelze zadat/	ANO/NE
BlackPort(s)	Zadává se interval	/nelze zadat/	ANO/NE

tabulka 1: Možnosti zadávání parametrů pro různé typy zdrojů

Zadávaná čísla se zpracovávají v hexadecimálním tvaru a mohou být zadávána ve tvaru 0x12 nebo 12.

Pro práci s registry platí pravidla uvedená v kapitole Konfigurace a registry.

Tipy při přidávání:

1. Při zadávání intervalu adres se kontroluje správnost zadávaných mezí a zda nedochází k jeho průniku s dalšími intervaly adres.

6.5.2. Odebrání zdroje

Odebrání zdroje se provede jeho výběrem v horní části okna označeného jako „*Used resources*“ a stiskem tlačítka „*Remove*“.

Pro práci s registry platí pravidla uvedená v kapitole Popis nastavení zdrojů ovladače.

6.5.3. Změna nastavení zdroje

Změna parametrů zdroje se provede jeho výběrem v horní části okna označeného jako „*Used resources*“. Po jeho výběru se současné nastavení zdroje překopíruje do části označené jako „*Selected resource*“. Zde se provede požadovaná změna parametrů a potvrdí se stiskem tlačítka „*Change*“. Po jeho stisku se změněné parametry zobrazí v části okna označené jako „*Used resources*“.

Pro práci s registry platí pravidla uvedená v kapitole Popis nastavení zdrojů ovladače.

6.6. Systémové prostředky

Při definování parametrů portů se zadává jenom interval obsluhovaných adres. Při obsluze jednoho portu se zadává opět interval adres o velikosti 1. V combo-box

„Dos“ se určuje zda má být port v dosových aplikacích obsluhován. Pokud vyberete „No“, tak se při inicializaci NT VDM neprovede nastavení obsluhy portů a dosová aplikace nemůže s těmito porty pracovat. Toto omezení, ale neplatí pro aplikaci určenou pro rozhraní WIN32, která při použití knihovny „HalPcKit.dll“ může s nastaveným porty pracovat vždy, pokud je dané zařízení zapnuto.

Black Port(s) slouží pro přístup k portům, které obsluhují jiné ovladače. Pokud uživatel začne zapisovat na porty, o kterých neví, jak je správně obsluhovat může dojít k pádu systému.

V současné době aplikace pro rozhraní WIN32 nemůže s přerušáním pracovat.

7. Rychlost zápisu na porty

Při měření jsme používali program, který ve smyčce volal OUT případně funkci WritePortByte poskytovanou knihovnou HalPcKit. Při testech jsme na uvedených počítačích naměřili následující hodnoty.

Uvedené rychlosti závisí na nainstalovaném systému a dalších komponentách, proto se na tyto časy nelze spolehnout.

	Aplikace pro DOS	Aplikace pro WIN32

tabulka 2: Rychlost přístupu na porty v systémech Windows 2000

Dále jsme se zaměřili na Windows XP, kde jsme naměřili následující hodnoty:

	Aplikace pro DOS	Aplikace pro WIN32

tabulka 3: Rychlost přístupu na porty v systémech Windows XP

8. Příloha

8.1. Stručně o operačním systému

Na rozdíl od systému DOS, který byl navržen jako jedno aplikační, jsou současné operační systémy navrženy jako více aplikační. V praxi to znamená, že můžete spustit několik aplikací najednou a mezi těmito aplikacemi přepínat.

V praxi to znamená, že systém obsahuje mechanismus, který na základě přidělených priorit nechá na procesoru po určitý čas běžet každou spuštěnou aplikaci. Z pohledu uživatele pak všechny aplikace běží současně, přestože ve skutečnosti v jednom časovém okamžiku běží pouze jedna aplikace.

Operační systémy obsahují množiny funkcí a datových typů, které slučují do programových rozhraní. Pomocí těchto rozhraní jsou systémy rozšiřovány o ovladače nových karet a aplikace dodávané různými firmami.

V současnosti nejrozšířenějšími operačními systémy jsou produkty fy. Microsoft. Vzhledem k tomu, že nové systémy platformy Windows NT nepodporují přímý přístup k portům, dodáváme ovladač, jenž Vám umožní na porty přistupovat. Bohužel při běhu ovladače mohou nastat situace, které vyžadují podrobnější znalost architektury systému. Cílem této přílohy není poskytnout podrobné informace o tom jak systém funguje, ale přiblížit činnost jednotlivých částí systému, které se na činnosti ovladače podílejí a ještě trochu více.

8.2. Uživatelská a systémová úroveň běhu programu

Aby se dosáhlo stavu, že při poškození jedné aplikace nedochází k poškození dalších aplikací, zavedly se různé úrovně běhu programu. Zpravidla se mluví o dvou úrovních – **uživatelské a systémové**.

Systémová úroveň (Kernel mode) je určena pro běh jádra operačního systému, které tvoří jednotlivý správci (např. správce paměti, úloh a IO). Na této úrovni neexistuje žádné omezení a pokud na této úrovni dojde k nějaké chybě, tak operační systém ukončí svou činnost. Programy běžící na této úrovni neumožňují žádnou přímou uživatelskou obsluhu. Tato úroveň je při vlastním běhu aplikace zodpovědná za přidělování paměti dle požadavků aplikace a důsledné oddělení jednotlivých aplikací.

Uživatelská úroveň (User mode) je určena pro běžné aplikace obsluhované uživatelem. Prostřednictvím těchto aplikací uživatel může ovládat programy, které běží na systémové úrovni. Pokud dojde na této úrovni k nějaké chybě, systém se přepne na systémovou úroveň, připraví data a zavolá obsluhu chyby. Protože na uživatelské úrovni neexistuje chyba, kterou operační systém nedokáže obsloužit, jsou chyby programu nazývány **výjimky**.

Pozn.: Přestože by k výjimce mělo docházet zřídka a program by se měl při jejím vyvolání ukončit, je možné naprogramovat aplikaci, která bude po obsluze výjimky pokračovat v dalším běhu. Tato vlastnost se používá především při práci s virtuální pamětí.

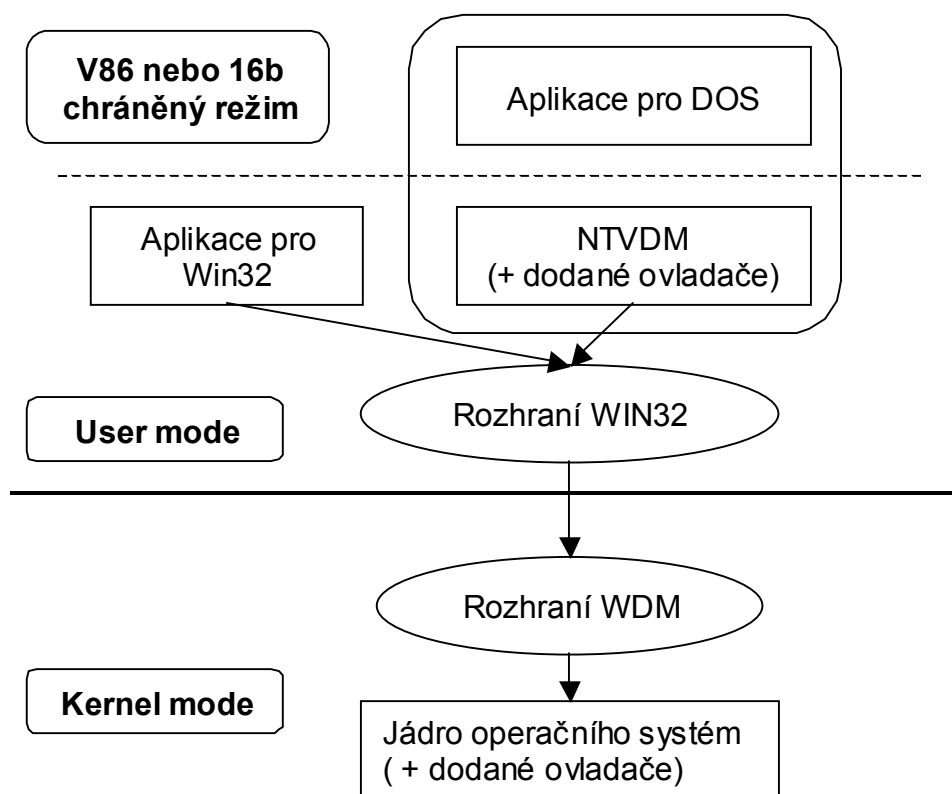
8.3. Rozhraní operačního systému

Aby operační systémy mohly být snadno rozšiřovány a doplňovány o další

aplikace, lze v každém systému najít definovanou množinu datových typů a funkcí, které mohou programátoři používat. Tuto množinu datových typů a funkcí budeme dále nazývat **programové rozhraní**.

Protože operační systémy platformy Windows NT podporují aplikace pro DOS jsou v těchto systémech tři různá rozhraní.

- WIN32 rozhraní pro psaní uživatelských aplikací. Obsluha aplikací je buď pomocí grafického prostředí (GUI) nebo pomocí textové řádky (CONSOLE). Pro tyto programy platí omezení týkající se uživatelské úrovně.
Programy na této úrovni mohou prostřednictvím služeb operačního systému volat programy na rozhraní WDM.
- VDM rozšíření rozhraní Win32 určené pro obsluhu aplikací běžících v prostředí NT VDM.
V tomto prostředí běží především dosové aplikace a aplikace určené pro Win3.x.
- WDM rozhraní pro psaní programů, které úzce pracují s vlastním operačním systémem. Tyto programy běží na systémové úrovni a zpravidla to jsou ovladače karet.

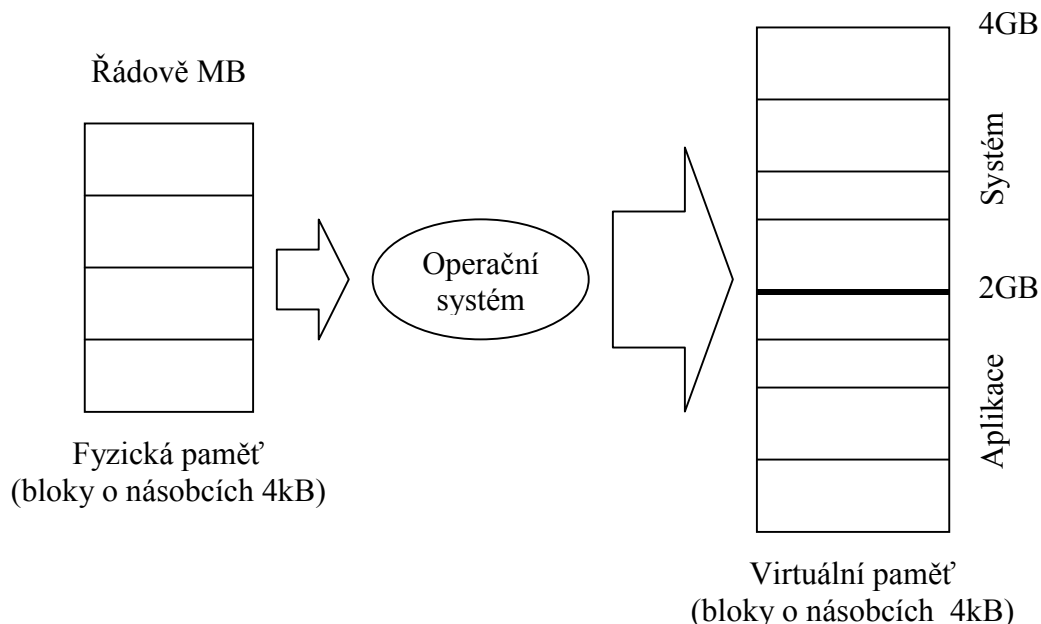


obrázek 10: Rozhraní v operačním systému

8.4. Paměťový podsystém

Operační systémy Windows přidělí každé aplikaci paměť 4 GB, tzv. virtuální paměť. Tato virtuální paměť nemusí odpovídat skutečné fyzické paměti, která je operačním systémem mapována pod virtuální paměť. Pokud dojde k situaci, že v systému není žádná volná fyzická paměť, pak systém vybere části fyzické paměti,

keré uloží do souboru označovaného jako SWAP soubor. Vybrané a uložené části fyzické paměti odpovídající virtuální paměti jsou označeny jako neplatné a mohou být systém použity při mapování. Pokud aplikace později k této neplatné paměti přistoupí, spustí se v jádře obsluha, která požadovanou paměť přečte do fyzické paměti a poté tuto paměť přiřadí virtuální paměti. (Pozn.: Přiřazení virtuální a fyzické paměti umožňuje používat různé části fyzické paměti pro stejnou část virtuální paměti.).



obrázek 11: Mapování a rozdělení paměti operačním systémem

Při vykonávání programu se používají registry DS, ES, CS, SS (případně FS a GS), které systém nastaví tak, aby umožňovaly práci s 4GB pamětí. Dále se budou označovat jako selektory. Tato paměť je pak spravována podle výše uvedeného popisu. Při vlastním přiřazování reálné paměti k virtuální paměti jsou také nastavována přístupová práva, např. zda je paměť pro čtení nebo zápis.

Vlastní paměť je pak rozdělena na dvě části po 2GB. Jedna část je určena pouze pro aplikaci a jsou v ní uložena jak data tak program. Druhá část je uložena v horní části a je vyhrazena pro operační systém (zpravidla jádro a systémové komponenty). Tento model paměti se nazývá **FLAT MODEL** a umožňuje každým selektorem adresovat celých 4GB.

Tato koncepce platí pouze pro aplikaci pracující v rozhraní WIN32, což se zjistí podle příznaku v hlavičce spustitelného souboru.

8.5. Aplikace pro WIN32

8.5.1. Použití knihovny HalPcKit

Při požadavku práce s porty v aplikaci pro rozhraní WIN32, se musí implicitně nahrát dynamická knihovna „HalPcKit.dll“. Pokud při jejím nahrávání nedošlo k chybě musí se zavolat funkce AttachToDevice, kterou se provede propojení knihovny se zadaným ovladačem. V tomto případě se jako vstupní parametr zadá „\\.\LptPcKit“. Pro uvolnění používaného ovladače se musí zavolat funkce

DetachFromDevice.

V současnosti knihovna neposkytuje žádnou funkci pro podporu přerušení.

BOOL WINAPI AttachToDevice(PTSTR pszDeviceName)

Funkce propojí knihovnu se zadaným ovladačem. Pokud ovladač není připojen nebo se připojení nepodařilo vrací výkonné funkce knihovny FALSE, operace se nezdařila.

BOOL WINAPI DetachFromDevice(void);

Funkce uvolnění ovladač. Při nepřipojeném ovladači nelze volat výkonné funkce knihovny.

○ **Množina funkcí pro práci se zaregistrovanými prostředky – Port(s)**

BOOL ReadPortByte(WORD port, PBYTE pValue)

Funkce přečte z portu 1BYTE a zapíše ho do proměnné pValue, tzn. chová se jako instrukce INB. V případě chyby vrací FALSE.

BOOL ReadPortWord(WORD port, PWORD pValue)

Funkce přečte z portu 1WORD a zapíše ho do proměnné pValue, tzn. chová se jako instrukce INW. V případě chyby vrací FALSE.

BOOL ReadPortDword(WORD port, PDWORD pValue)

Funkce přečte z portu 1DWORD a zapíše ho do proměnné pValue, tzn. chová se jako instrukce IND. V případě chyby vrací FALSE.

BOOL ReadPortBufferByte(WORD port, PBYTE pValue, WORD count)

Funkce přečte z portu několik BYTE a zapíše je do pole proměnných pValue, tzn. chová se jako instrukce INSB. V případě chyby vrací FALSE.

BOOL ReadPortBufferWord(WORD port, PWORD pValue, WORD count)

Funkce přečte z portu několik WORD a zapíše je do pole proměnných pValue, tzn. chová se jako instrukce INSW. V případě chyby vrací FALSE.

BOOL ReadPortBufferDword(WORD port, PDWORD pValue, WORD count)

Funkce přečte z portu několik DWORD a zapíše je do pole proměnných pValue, tzn. chová se jako instrukce INSD. V případě chyby vrací FALSE.

BOOL WritePortByte(WORD port, BYTE value)

Funkce zapíše na port 1BYTE, tzn. chová se jako instrukce OUTB. V případě chyby vrací FALSE.

BOOL WritePortWord(WORD port, WORD value)

Funkce zapíše na port 1WORD, tzn. chová se jako instrukce OUTW. V případě chyby vrací FALSE.

BOOL WritePortDword(WORD port, DWORD value)

Funkce zapíše na port 1DWORD, tzn. chová se jako instrukce OUTD. V případě chyby vrací FALSE.

BOOL WritePortBufferByte(WORD port, PBYTE pValue, WORD count)

Funkce zapíše na port několik BYTE z pole pValue o počtu count, tzn. chová se jako instrukce OUTSB. V případě chyby vrací FALSE.

BOOL WritePortBufferWord(WORD port, PWORD pValue, WORD count)

Funkce zapíše na port několik WORD z pole pValue o počtu count, tzn. chová se jako instrukce OUTSW. V případě chyby vrací FALSE.

BOOL WritePortBufferDword(WORD port, PDWORD pValue, WORD count)

Funkce zapíše na port několik DWORD z pole pValue o počtu count, tzn. chová se jako instrukce OUTSD. V případě chyby vrací FALSE.

○ **Množina funkcí pro práci s nezaregistrovanými prostředky – BlackPort(s)**

BOOL ReadBlackPortByte(WORD port, PBYTE pValue)

Funkce přečte z portu 1BYTE a zapíše ho do proměnné pValue, tzn. chová se jako instrukce INB. V případě chyby vrací FALSE.

BOOL ReadBlackPortWord(WORD port, PWORD pValue)

Funkce přečte z portu 1WORD a zapíše ho do proměnné pValue, tzn. chová se jako instrukce INW. V případě chyby vrací FALSE.

BOOL ReadBlackPortDword(WORD port, PDWORD pValue)

Funkce přečte z portu 1DWORD a zapíše ho do proměnné pValue, tzn. chová se jako instrukce IND. V případě chyby vrací FALSE.

BOOL ReadBlackPortBufferByte(WORD port, PBYTE pValue, WORD count)

Funkce přečte z portu několik BYTE a zapíše je do pole proměnných pValue, tzn. chová se jako instrukce INSB. V případě chyby vrací FALSE.

BOOL ReadBlackPortBufferWord(WORD port, PWORD pValue, WORD count)

Funkce přečte z portu několik WORD a zapíše je do pole proměnných pValue, tzn. chová se jako instrukce INSW. V případě chyby vrací FALSE.

BOOL ReadBlackPortBufferDword(WORD port, PDWORD pValue, WORD count)

Funkce přečte z portu několik DWORD a zapíše je do pole proměnných pValue, tzn. chová se jako instrukce INSD. V případě chyby vrací FALSE.

BOOL WriteBlackPortByte(WORD port, BYTE value)

Funkce zapíše na port 1BYTE, tzn. chová se jako instrukce OUTB. V případě chyby vrací FALSE.

BOOL WriteBlackPortWord(WORD port, WORD value)

Funkce zapíše na port 1WORD, tzn. chová se jako instrukce OUTW. V případě chyby vrací FALSE.

BOOL WriteBlackPortDword(WORD port, DWORD value)

Funkce zapíše na port 1DWORD, tzn. chová se jako instrukce OUTD. V případě chyby vrací FALSE.

BOOL WriteBlackPortBufferByte(WORD port, PBYTE pValue, WORD count)

Funkce zapíše na port několik BYTE z pole pValue o počtu count, tzn. chová se jako instrukce OUTSB. V případě chyby vrací FALSE.

BOOL WriteBlackPortBufferWord(WORD port, PWORD pValue, WORD count)

Funkce zapíše na port několik WORD z pole pValue o počtu count, tzn. chová se jako instrukce OUTSW. V případě chyby vrací FALSE.

BOOL WriteBlackPortBufferDword(WORD port, PDWORD pValue, WORD count)

Funkce zapíše na port několik DWORD z pole pValue o počtu count, tzn. chová se jako instrukce OUTSD. V případě chyby vrací FALSE.

8.5.2. Použití funkcí rozhraní WIN32

V této části bude naznačeno jakými funkcemi a jak je volat. Podrobnější popis je nad rámec této přílohy.

Jak bude naznačeno později, práce s programy na systémové (dále *ovladač*) úrovni probíhá pomocí přístupu označovaného *klient-server*. Tzn. že aplikace na uživatelské úrovni vnesou požadavek na ovladač, který tento požadavek vykoná. Při zpracovávání požadavku jsou používány dva přístupy – *asynchronní* a *synchronní*.

Při *asynchronním přístupu* je požadavek zaznamenán a řízení se vrací aplikaci na uživatelské úrovni. Tato aplikace může potom pokračovat v provádění programu až

do místa, kde potřebuje znát výsledek operace. V případě, že operace ještě není dokončena, program se zastaví a čeká dokud není operace dokončena. Oznámení o ukončení operace se provádí pomocí synchronizačního prostředku, který se předal při volání operace.

Při *synchronním přístupu* se program na uživatelské úrovni zastaví, dokud operace není dokončena.

Abyste mohli s ovladačem začít pracovat, musíte na tento program získat ukazatel, dále se označuje *HANDLE* a znát jméno ovladače. Jméno ovladače potom použijete při volání funkce *CreateFile*, která vrátí handle na požadovaný ovladač.

Pozn.: Jméno ovladače začíná `\\.\<jmenný prostor>\<jméno program>`. Na přiloženém obrázku jsou pro ilustraci zobrazeny jmenné prostory na systémech platformy NT. Pro ovladač LPT PcKit se používá jméno `\\.\LptPcKit`, což je přezdívka pro `\\.\GLOBALROOT\DosDevices\LptPcKit` nebo `\\.\GLOBALROOT\Device\LptPcKit`.

Pozn.: V případě práce s ovladači by jste neměli používat náhodné požadavky (příkazy), protože by mohlo dojít k poškození operačního systému.

??	Directory
ArcName	Directory
BaseNamedObjects	Directory
Callback	Directory
Device	Directory
DosDevices	SymbolicLink
Driver	Directory
FileSystem	Directory
KnownDlls	Directory
NLS	Directory
ObjectTypes	Directory
RPC Control	Directory
Security	Directory
Windows	Directory
WmiGuid	Directory

obrázek 12: Jmenné prostory na platformě NT

Po získání handle na ovladač můžete s tímto ovladačem začít komunikovat, tj. posílat požadavky. Požadavky se posílají pomocí *DeviceIoControl*. V případě, že ovladač poskytuje určité rozhraní lze také použít *WriteFile* či *ReadFile*.

Ukončení práce s tímto ovladačem provedete pomocí *CloseFile*.

8.6. Aplikace pro DOS

V případě, že se spouští aplikace pro DOS, přechází procesor INTEL do speciálního módu – tzn. V86. Tento mód vytvoří za pomoci systému prostředí DOS a zároveň systému dovoluje spravovat veškeré specifické instrukce a přístupy. Mezi takové instrukce patří zakázání a povolení přerušení, práce s vstupně výstupními zařízení a práce s určitými oblastmi paměti (např. VGA paměť) atd.

V systému jsou standardně obsluhována zařízení, které jsou běžné dostupné jako různé řadiče spravované systémem (např. přerušení a DMA), zobrazovací zařízení, přístupy na disky atd. Obsluhy těchto zařízení se chovají jako byste pracovali se skutečným zařízením, dále se budou označovat jako **virtuální zařízení**.

Aby bylo možné do tohoto prostředí přidat další zařízení, je vytvořeno rozhraní VDM (virtual dos machine). Toto prostředí je umístění v prostředí WIN32 a umožňuje vytvářet virtuální zařízení pro reálné aplikace.

8.7. Popis VDM

8.7.1. Obecně

Při spuštění aplikace pro DOS se spustí aplikace, která vytvoří dosové prostředí, které uživateli poskytuje pouze 1MB paměti. Systém nastaví selektory (CS, SS, DS, ES) tak, aby umožňovaly pracovat pouze s 64kB. Aplikace, která toto prostředí vytváří, umožňuje toto prostředí rozšiřovat pomocí definovaných funkcí rozšiřujících prostředí Win32 a označuje se jako NT VDM. V prostředí NT VDM je k dispozici 4GB a může používat rozhraní Win32.

Při spouštění NT VDM se do paměti načítají a inicializují uživatelem dodané zařízení. Popis inicializace bude popsán dále. Po jejich úspěšné inicializaci se spustí vlastní aplikace. Při ukončení aplikace se nejdříve ukončí uživatelem dodané zařízení a potom se ukončí vlastní prostředí. Možná si kladete otázku, kam se v paměti dodaná zařízení umístí a kolik zaberou z konvenční paměti dosové aplikace? Dodaná zařízení Vám v konvenční paměti nezaberou žádný prostor a jsou operačním systémem umístěna v prostoru aplikace NT VDM. Přestože pro dosový prostor máte k dispozici 1MB paměti, dosová aplikace má dostupných mnohem méně. Toto omezení je dáno definicí DOS. Část paměti je vždy rezervována pro BIOS, VGA a další aplikace dle nastavení **autoexec.nt** a **config.nt**.

Tipy:

1. V systémech platformy Windows NT je možné při spouštění dosové aplikace určit, které konfigurační soubory použít. Nejsou-li tyto soubory definovány, použijí se implicitní soubory.)
2. Při spouštění NT VDM (systémová komponenta) systém přepne procesor do režimu V86, který mu umožní vytvoření dosového prostředí. Tzn v tomto prostoru se připraví paměť 1MB a do této paměti se umístí požadované služby (BIOS a různé služby DOS). Tato koncepce umožňuje spouštění aplikací v reálném režimu nebo aplikací pro 16b chráněný režim.

8.7.2. Uživatelská zařízení

Po vytvoření NT VDM a před vytvořením dosového prostředí se inicializují služby, které jsou volány při každé speciální instrukci – např. IN, OUT, přístupy do určité paměti, atd. Služby pro standardní zařízení (určené fy. Microsoft) jsou součástí VDM. V případě, že aplikace nebo uživatel vyžaduje přístup na zařízení, které není v dosovém prostředí standardně podporováno, jsou dodávány uživatelské služby. Tyto služby se inicializují po standardních obsluhách a můžou používat standardní funkce rozhraní Win32. Podle stavu zařízení je požadavek obslužen v obsluze tj. v uživatelské úrovni nebo je předán operačnímu systému tj. ovladači. Při předání požadavku ovladači (systému) dochází ke změně úrovně programu a k nezanedbatelné režii při ukládání a obnovení současné úrovně.

Pozn.: Přestože dosové aplikace se tváří jako samostatné aplikace, můžou mít částečně sdílenou paměť. Toto sdílení je zavedeno kvůli kompatibilitě se staršími aplikacemi psanými pro Windows 3.x. Proto při chybě jedné aplikace může dojít k pádu více takto propojených aplikací.

8.8. Adresáře a registry týkajících se ovladačů

8.8.1. Popis adresářů

Pokud bychom se na operační systém podívali pohledem běžného uživatele zjistíme, že systém je tvořen spousty souborů. Tyto soubory jsou uloženy přímo nebo v podadresářích adresáře, který se označuje jako systémový a jeho cesta je nastavena v systémové proměnné „*SystemRoot*“. V tomto adresáři je dále adresář „*System32\drivers*“, který obsahuje ovladače používané a dodávané ovladače.

Pozn.: V případě nastavení této proměnné na jiný adresář než systémový nebude ovladač LptPcKit fungovat.

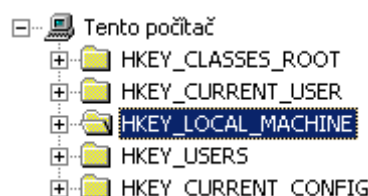
Při instalaci ovladačů se používá soubor s koncovkou *INF*, který obsahuje informace o instalovaném zařízení. Tyto informace se později ukládají do registrační databáze. Tato databáze je v systémech platformy Windows NT jednou z nejlépe chráněných částí systému. Její ochrana je založena na souborovém systému NTFS, který obsahuje bezpečnostní prvky a navíc při změně této databáze jsou vytvářeny žurnály. Tyto žurnály umožňují při chybě a nedokončení požadované operace vše vrátit do původního stavu. Vlastní soubory *INF* jsou uloženy v podadresáři „*inf*“.

Poznámky a tipy:

1. Protože sběrnice PCI je od začátku koncipována jako Plug&Play, může systém detekovat všechna připojená zařízení. Při získávání informací o těchto zařízeních se vrací řetězce, které zařízení jednoznačně identifikují. Tyto řetězce jsou systémem dále používány při vyhledávání vhodného ovladače. Při hledání vhodného ovladače systém prohledává *INF* soubory a obsahuje-li nějaký soubor identifikační řetězec, nabídne se jeho použití při instalaci.
2. Při instalaci systému jsou do adresáře „*inf*“ umístěny *INF* soubory standardních zařízení, které jsou dodávány se systémem. Proto je možné, že při vložení nové karty do počítače, systém zjistí identifikační řetězec a nalezne standardní *INF* soubor. V dalším kroku instalace zjistí, že zařízení ještě nebylo instalováno a může Vás vyzvat k vložení instalačního CD se soubory ovladačů.
3. Každá instalace zařízení je systém monitorována a zapisována do soubory *SETUPAPI.LOG*, který je uložen v systémovém adresáři.

8.8.2. Popis registru

Registrační databáze je tvořena 5 větvemi – *HKEY_USERS*, *HKEY_CLASSES_ROOT*, *HKEY_CURRENT_CONFIG*, *HKEY_CURRENT_USER* a *HKEY_LOCAL_MACHINE*. Dále se zaměříme pouze na poslední zmiňovanou větev ***HKEY_LOCAL_MACHINE***, která obsahuje informace o počítači a systému, tj. službách a ovladačích.



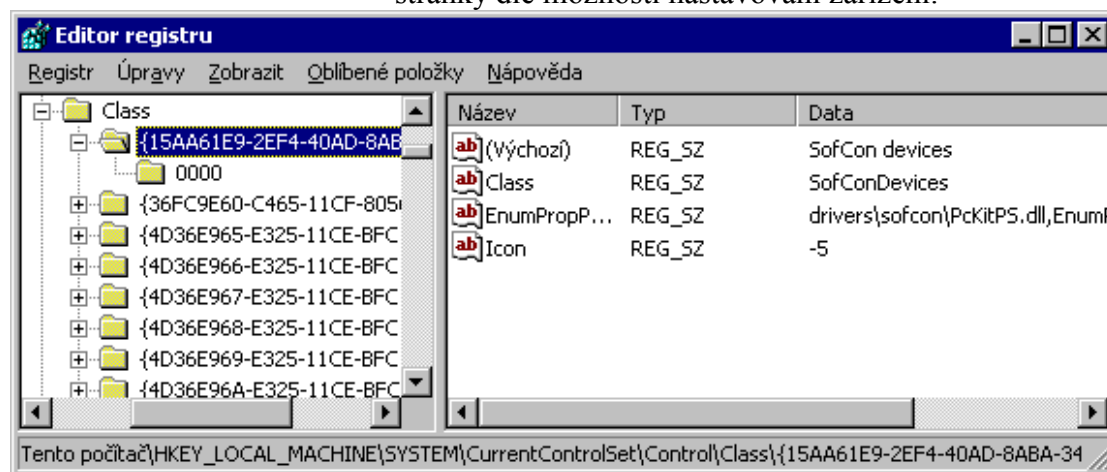
obrázek 13: Větvě registrační databáze

- o Třída zařízení

Každé zařízení můžeme zařadit do jedné z kategorií, dále **tříd**, popisujících zařízení, např. BATTERY, CD ROM, DISK, MULTIMEDIA, NETWORK, PRINTERS atd. Tyto třídy jsou definovány systémem a určují požadavky na chování ovladače zařízení. Třídy lze také rozšiřovat dodavateli ovladačů. Popis aktuálních tříd je ve větvi „*System\CurrentControlSet\Control\Class*“. Kompletní popis všech možných nastavení je nad rámec tohoto textu. Proto bude popsáno pouze několik položek, které používá ovladač LPT PcKit.

Každá třída má vlastní GUID (Globally unique identifier), který ji jednoznačně identifikuje a je určen buď systémem nebo vygenerován pomocí guidgen. (Pozn.: Dále popisovaný ovladač má „15AA61E9-2EF4-40ad-8ABA-348F3ED5F747“.) Položky v této větvi mají následující význam:

<no name>	název třídy, který se bude zobrazovat ve správci zařízení, např. SofCon devices
Class	název třídy, který se použije v databázi registrů, např. SofConDevices
Icon	ikona, která se bude zobrazovat ve správci zařízení
EnumPropPages32	odkaz na knihovnu, která rozšiřuje možnosti nastavování ovladače pomocí správce zařízení, tj. přidá stránky dle možností nastavování zařízení.



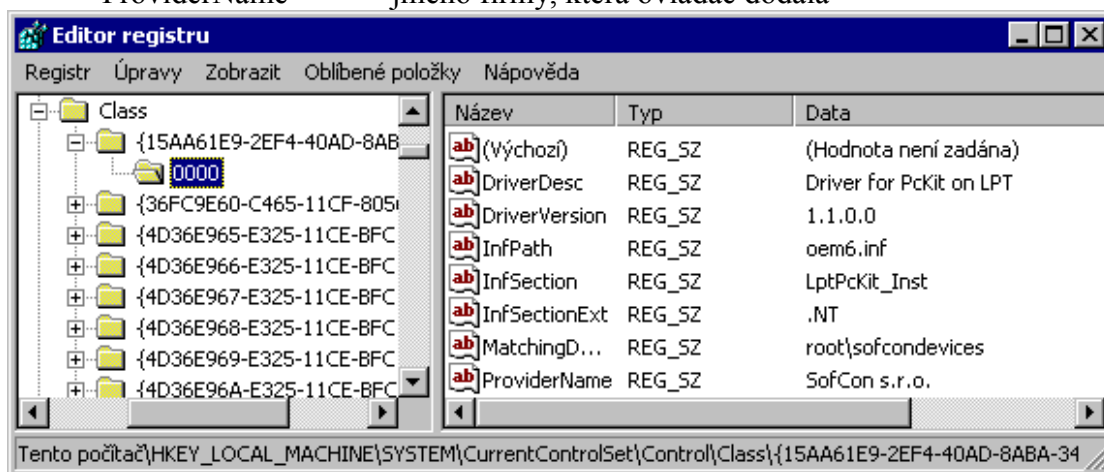
obrázek 14: Třídy zařízení

Protože do jedné třídy může patřit více ovladačů, je každému ovladači vytvořena instance. Tyto instance jsou vzestupně číslovány. Pokud bude v systému nainstalován pouze jeden ovladač třídy SofConDevices bude zde pouze větev „0000“.

Kompletní popis všech možných nastavení je nad rámec tohoto textu. Proto bude popsáno pouze několik položek, které používá ovladač LPT PcKit.

DriverDesc	řetězec, který se zobrazuje při instalaci ovladače
DriverVersion	verze ovladače
InfPath	název inf souboru, pod kterým je uložen v adresáři „inf“

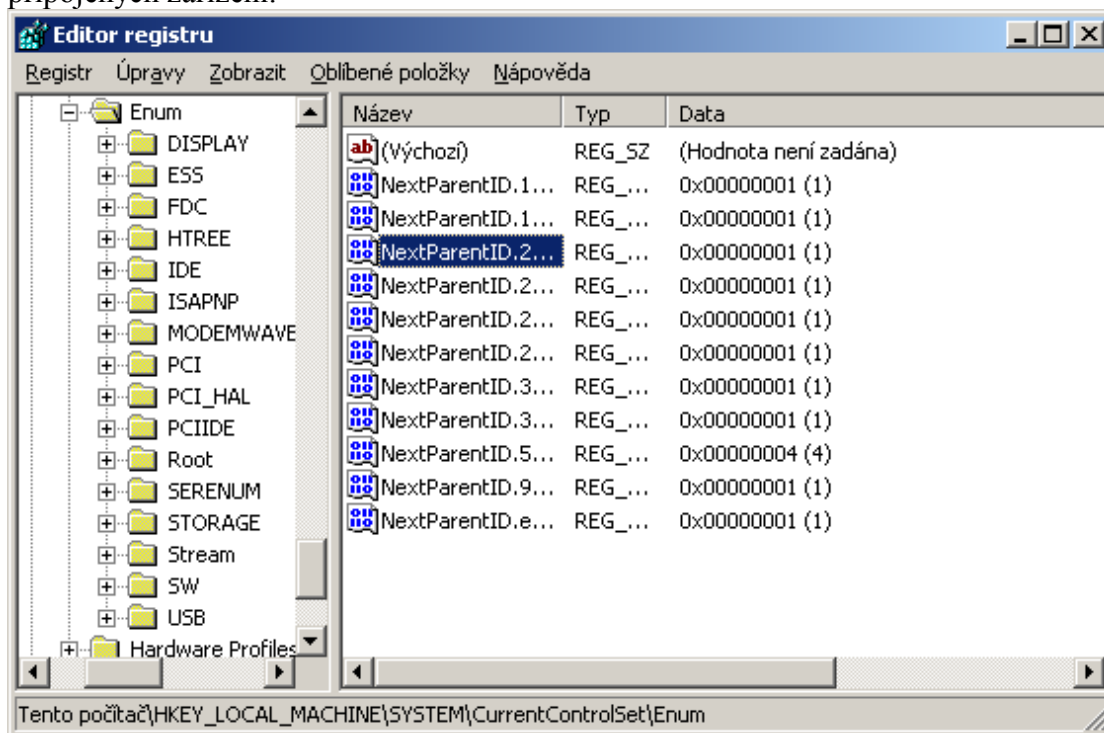
InfSection	název sekce v souboru INF, která obsahuje informace o ovladači
InfSectionExt	platforma – 32b systém x86 případně 64b systém x86.
MatchingDeviceId	lokalizace instancí tohoto ovladače
ProviderName	jméno firmy, která ovladač dodala



obrázek 15: Instance třídy

- o Instance spuštěných zařízení

Obecně jsou ovladače koncipovány tak, aby jeden ovladač mohl ovládat několik zařízení. Toho je dosaženo pomocí oddělení datových informací ovladače od vlastního kódu. Tyto datové informace jsou uloženy v registru pod větví „System\CurrentControlSet\Enum“. V této větvi jsou uloženy informace o jednotlivých komponentách, které sledují sběrnice. Termín sběrnice je zde použit ve významu různých komunikačních protokolů, které můžou umožnit sledování připojených zařízení.



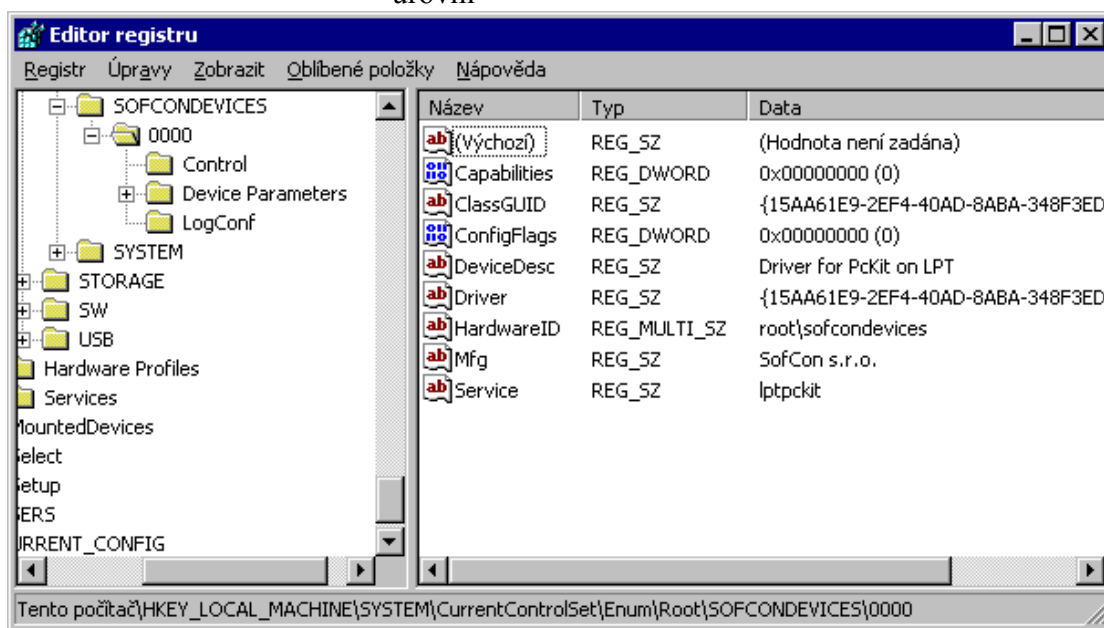
obrázek 16: Enumerátory sběrnic

Dále se zaměříme na větev „Root“, která obsahuje informace o zařízeních spuštěných systémem. Pokud bude spuštěn ovladač LPT PcKit najdeme zde větev

„SofConDevices“. Zde jsou obsaženy všechny instance zařízení, které patří třídy SofConDevices. Jednotlivé instance jsou potom vzestupně číslovány. Při jednom spuštěném zařízení třídy SofConDevices zde bude pouze větev „0000“.

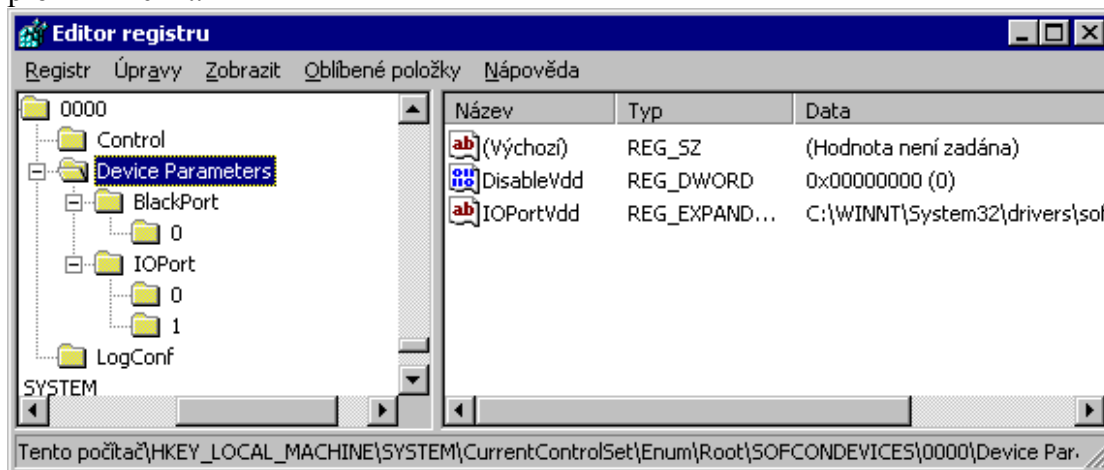
Kompletní popis všech možných nastavení je nad rámec tohoto textu. Proto bude popsáno pouze několik položek, které používá dále popisovaný ovladač.

Class	jméno třídy, do které zařízení patří
ClassGUID	GUID třídy, do které zařízení patří
DeviceDesc	řetězec, který se zobrazuje ve správci zařízení
Driver	lokalizace ovladače
Service	označení ovladače, který prostřednictvím systému (rozhraní WDM) komunikuje s aplikací na uživatelské úrovni



obrázek 17: Instance spuštěného zařízení

V každé větvi, která popisuje jednotlivé instance zařízení jsou tři větve popisující stav případně nastavení, které se použije při prvním spuštění zařízení. Z větvi „Control“, „LogConf“ a „DeviceParameters“ je pro nás důležitá pouze větev „DeviceParams“ obsahující parametry zařízení, které lze nastavovat pomocí správce zařízení. Parametry jsou poplatné danému zařízení a budou součástí popisu ovladače pro LPT PcKit.



obrázek 18: Parametry ovladače LPT PcKit

Pozn.: „LogConf“ obsahuje nastavení zdrojů, kterými zařízení může disponovat. Zpravidla se toto nastavování provádí u zařízení, které nepodporují Plug&Play.

Pozn.: „Control“ obsahuje systémové informace a aktuální stav zařízení.

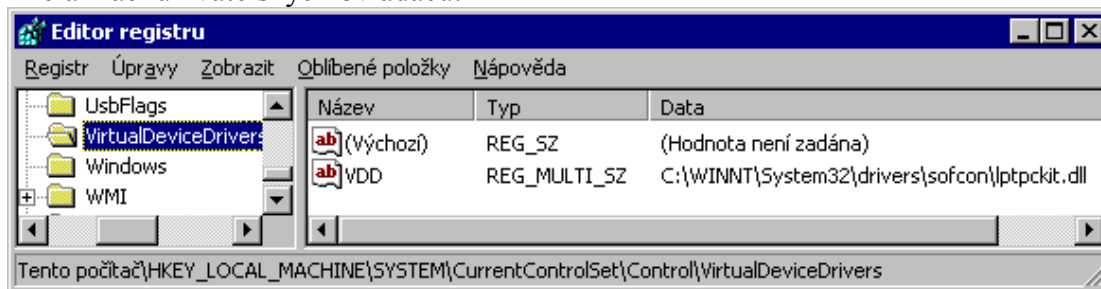
Pozn.: Dále bude stručný popis parametrů větve „DeviceParameters“, jejíž složky a položky odpovídají nastavení ve správci zařízení.

DisableVdd popisuje zda se mají inicializovat dosové ovladače

IOPortVdd cesta na ovladač obsluhující IOPort a Irq

Absolutní cesty na ovladače se pak podle nastavení správce zařízení odstraňují nebo přidávají do položky

„*System\CurrentControlSet\Control\VirtualDeviceDrivers\Vdd*“, která se používá pro inicializaci uživatelských ovladačů.



obrázek 19: Uložení uživatelských ovladačů

Ve větvích „*IOPort*“, „*BlackPort*“ jsou definovány používané nebo už někdy použité definice virtuálních zařízení a systémové prostředky. V každé této větvi je položka „*Count*“, ve které je poznamenán počet aktuálně používaných zařízení a podvětve jednotlivě definovaných systémových prostředků a virtuálních zařízení. Jejich význam odpovídá nastavování ve správci zařízení.

o Služby – výkonná část ovladače

Ovladač jedné desky může tvořit několik různých souborů, které implementují různé vrstvy ovladače. Soubory by měly být pojmenovány jednoznačnými názvy, aby při instalaci nedocházelo ke kolizím. S takto pojmenovanými ovladači se může komunikovat prostřednictvím rozhraní WDM, pokud tento ovladač nemá nikdo exklusivně otevřen.

Jednotlivé služby jsou v registru uloženy pod „*System\CurrentControlSet\Services*“. V případě nainstalovaného ovladače pro LPT PcKit ve větvi „*lptpckit*“ najdete několik parametrů a další větvi „*Enum*“ a „*Security*“. Kompletní popis všech možných nastavení je nad rámec tohoto textu. Proto bude popsáno pouze několik položek, které používá dále popisovaný ovladač.

DisplayName zobrazované jméno služby

ErrorControl určuje jak se chovat pokud dojde k chybě.

ImagePath uložení souboru, který implementuje tuto službu.

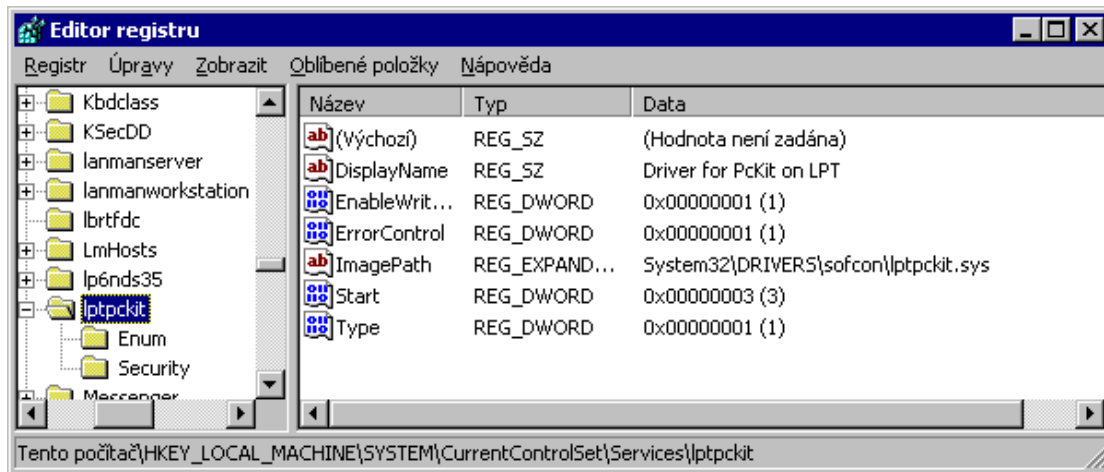
Start určuje, kdy a jak tuto službu spouštět.

Type určuje, na jaké úrovni a jak služba bude spuštěna.

EnableWriteLog povoluje zápisy do protokolu událostí

Pozn.: „*Enum*“ obsahuje systémové informace o používání této služby.

Pozn.: „*Security*“ obsahuje informace o přístupových právech k této službě.



obrázek 20: Nastavení služby

9. Literatura

- [1] Microsoft Corp., MSDN Library
- [2] Microsoft Corp., Microsoft Windows 2000 Driver Development Kit
- [3] W. Oney, Programming the Microsoft Windows driver model, Microsoft Press, ISBN 0-7356-0588-2
- [3] J. Richter, Windows pro pokročilé a experty, Computer Press, ISBN 80-85896-89-3
- [4] D. A. Solomon, Windows NT pro administrátory a vývojáře, Computer Press, ISBN 80-7226-147-9