



Univerzita Hradec Králové  
Fakulta informatiky a managementu

# Start systému

Mgr. Josef Jan Horálek, Ph.D. & Ing. Tomáš Svoboda, Ph.D.





- = Hardwarová kontrola systému (self-test)
- = Načtení a start zavaděče
- = Načtení a rozbalení jádra
- = Inicializace jádra, základní detekce HW
- = Spouštění programu init
- = Postupné spouštění různých procesů



- = Spouští se firmware, který kontroluje:
  - = paměť, základní součásti systému a periferie,
  - = zda je možnost vstoupit do setup a nastavit další parametry, provádět další kontroly a detekce,
  - = na závěr se řízení předává BIOSu nebo UEFI. Ty mají za úkol řídit další etapu startu.



= BIOS

- = Limit 2.1TB disku (MBR schéma)
- = 16 bitový režim kompatibility
- = 1MB RAM!
- = Většinou v textovém režimu
- = Aktualizace možná, ale obtížná

= UEFI

- = Teoreticky bez limit HDD
- = Využívá GPT tabulku oddílů
  - = GUID Partition Table
  - = Legacy režim
- = Většinou grafickém režimu
- = Umožňuje rychlejší start
- = Jednodušší aktualizace



FIM UHK

## Jak vstoupit do BIOS nebo UEFI

- = Klávesy DELETE, F2, F12, F1 atd.
- = Problémy prvního světa:
  - = Počítače startují tak rychle, že nestihneme přečíst informaci o klávese.



ASUS UEFI BIOS Utility - EZ Mode

Exit/Advanced Mode

10:24

P0H61-M LX3 PLUS R2.0

BIOS Version : 0609

CPU Type : Intel(R) Core(TM) i3-3210 CPU @ 3.20GHz Speed : 3200 MHz

Monday (09/23/2013)

Total Memory : 8192 MB (DDR3 1333MHz)

Temperature

CPU +93.2°F/+34.0°C

MB +91.4°F/+33.0°C

Voltage

CPU 1.040V 5V 5.160V

3.3V 3.376V 12V 11.904V

Fan Speed

CPU\_FAN 1015RPM CHA\_FAN N/A

Q-Fan Control

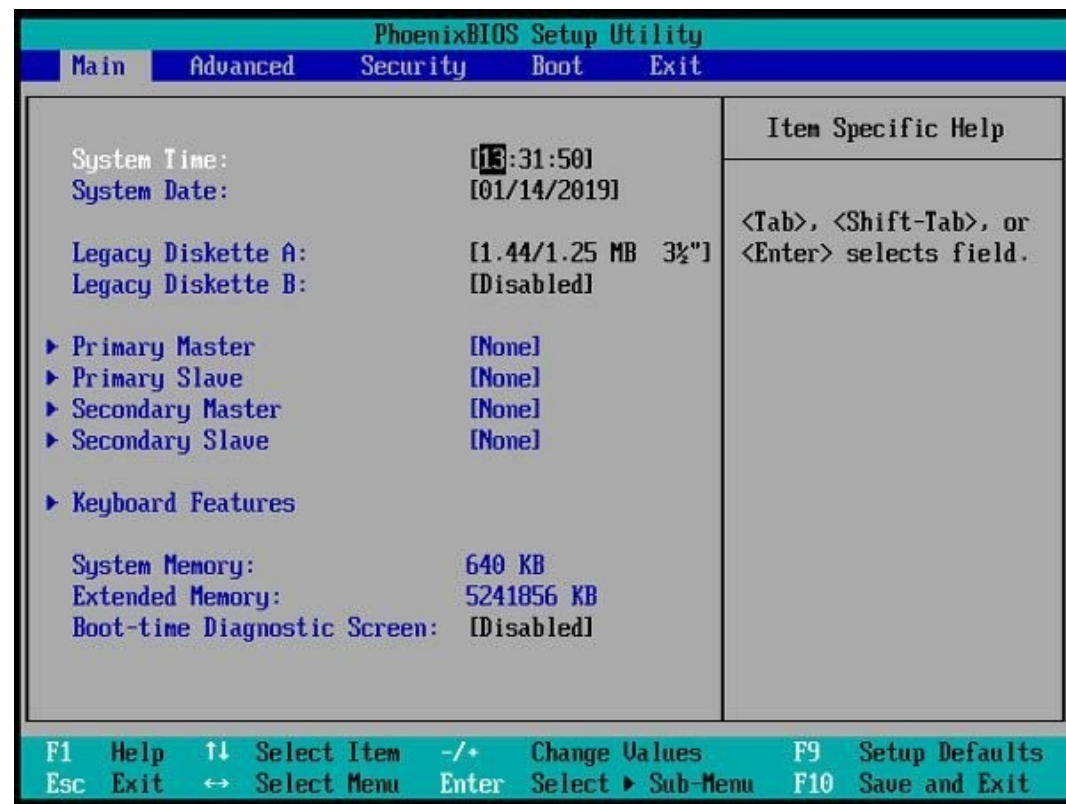
Quiet

Performance Energy Saving Turbo

Boot Priority

Use the mouse to drag or keyboard to navigate to decide the boot priority.

Shortcut (F3) **Advanced Mode (F7)** Boot Menu (F8) Default (F5)



PhoenixBIOS Setup Utility

Main Advanced Security Boot Exit

|                              |                    | Item Specific Help                               |
|------------------------------|--------------------|--|
| System Time:                 | [13]:31:50         | <Tab>, <Shift-Tab>, or<br><Enter> selects field. |
| System Date:                 | [01/14/2019]       |  |
| Legacy Diskette A:           | [1.44/1.25 MB 3½"] |  |
| Legacy Diskette B:           | [Disabled]         |  |
| ▶ Primary Master             | [None]             |  |
| ▶ Primary Slave              | [None]             |  |
| ▶ Secondary Master           | [None]             |  |
| ▶ Secondary Slave            | [None]             |  |
| ▶ Keyboard Features          |                    |  |
| System Memory:               | 640 KB             |  |
| Extended Memory:             | 5241856 KB         |  |
| Boot-time Diagnostic Screen: | [Disabled]         |  |

F1 Help ↑↓ Select Item -/+ Change Values F9 Setup Defaults  
Esc Exit ↔ Select Menu Enter Select ▶ Sub-Menu F10 Save and Exit



- = BIOS obsahuje základní funkce pro práci s HW. Podle seznamu zařízení začne hledat zavaděč systému (boot loader).
- = Zavaděč získá načtením startovacího záznamu (boot record).
- = Dříve byl zavaděč uložen jen přímo v rámci startovacího záznamu.
- = Dnes jen hlavní část – natažení celého zavaděče odjinud.
- = Úkol zavaděče – načíst jádro systému
  - = moderní zavaděče jako je GRUB nepotřebují znát konkrétní uložení jádra.

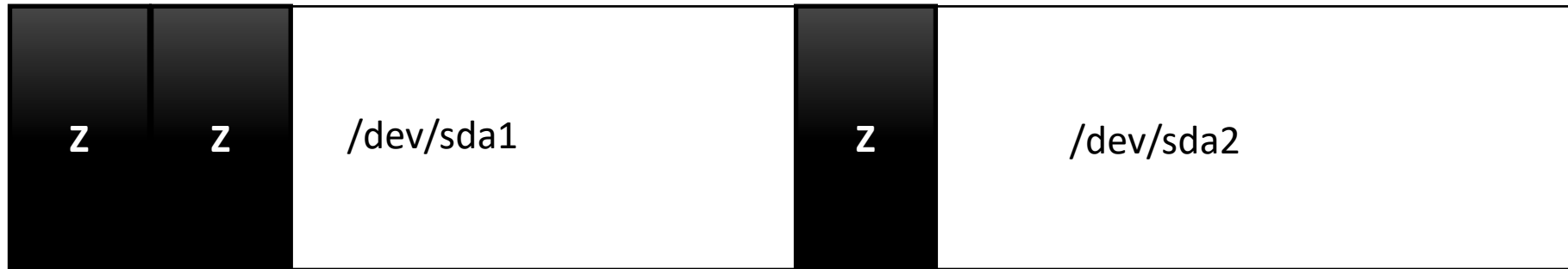


- = Disk je formátovaný jako GPT
- = Na začátku disku je seznam všech oddílů a jejich pozice na disku
- = UEFI hledá EFI oddíl, kde je zavaděč OS
- = Realita je dnes komplikovanější, díky kompatibilitě, Secure Boot a dalším technologiím

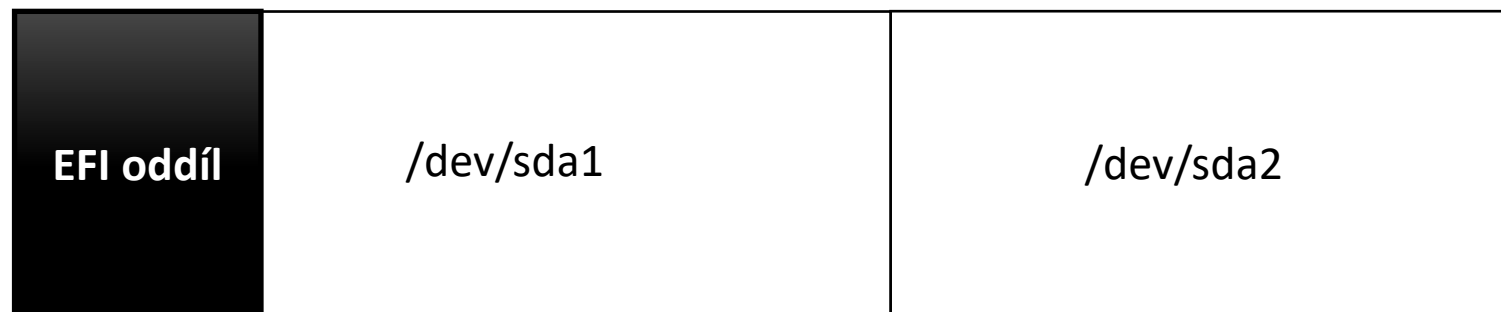




BIOS



UEFI





- = Najde-li zavaděč jádro, musí se před spuštěním načíst a rozbalit do cílové paměti.
- = Rozbalení zajišťuje rutina na začátku jádra.
- = Nejčastější metoda komprese jádra je LZ4, dříve deflate.
- = Při rozbalování je kontrolována integrita jádra.
- = Příprava na inicializaci
  - = příprava tabulky paměťových stránek a detekce procesoru.



- = Inicializace jádra = složitý proces
- = Nastavení přerušení, vytvoření datových struktur, inicializace ovladačů a detekce zařízení,
  - = ovladače v modulech nejsou inicializovány
  - = Některé ovladače musí být zakompilovány přímo v jádře, aby mohl systém běžet
- = Vytvoření speciálního vlákna, z něhož se stává první proces –init



- = Rozlišujeme tři skupiny programů
  - = Jednoznačně patřící do jádra
    - = Jaderné subsystemy, ovladače zařízení, plánovače úloh, plánovače blokových operací, základní bezpečnostní komponenty, primární souborové systémy
  - = Programy na rozhraní
    - = Souborové systémy, mechanismy meziprocesové komunikace, monitorovací a řídicí komponenty, firewally apod.
- = Ostatní
  - = Běžné aplikace, síťové servery, ....



Primární souborové  
systémy

Speciální aplikace

Sítě (filtrace, směrování)

Ovladače zařízení

Bezpečnostní technologie

Souborové systémy

Meziprocesová  
komunikace

Souborové systémy  
procfs, sysfs, ..

Základní technologie  
jádra

Subsystemy jádra



- = INIT má výhradní postavení – všechny uživatelské procesy vznikají právě pod ním
- = Rodičem init je samotné jádro.
- = Role init:
  - = řízení startu systému,
  - = řízení a přepínání úrovní běhu (runlevels),
  - = správa procesů jejichž rodič skončil,
  - = řízení vypínání/restartu systému,



FIM UHK

- = Typické spouštění:
  - = připojení speciálních souborových systémů
  - = aktivace bezpečnostních technologií
  - = nastavení parametrů pro konzoly
  - = inicializace síťových rozhraní
  - = kontrola souborových systémů
  - = úklid v adresářích pro dočasné soubory
  - = zapnutí paketového filtru
  - = start základních systémových démonů
  - = start konzol, případně grafického rozhraní

- = V dřívějším systému nastartoval program init (resp. upstart) prostředí systému podle zvoleného runlevelu.
  - = Ten představuje množinu služeb, které mají být spuštěny nebo zastaveny.
  - = Například runlevel 0 znamená vypnutí systému a runlevel 5 spuštění grafického režimu.



- = Jeho myšlenka je tedy nastavení systému podle předem definovaných úrovní a jeho rychlého přepnutí.
- = Toho bylo docíleno za využití tzv. symbolických linků v `/etc/init.d/rcX.d` (kde X reprezentuje využitý runlevel).
- = Služby byly spouštěny sériově, a pokud bylo následně potřeba provést určitou akci, využil se k tomu příkaz (další snímek)

```
service <název-init-skriptu> <akce>  
[root@nagios ~]# service httpd status  
httpd (pid 1833) is running...
```

- = V Ubuntu 15.04 došlo k podstatné změně a program init zde byl nahrazen na systemd
- = Náhrada runlevelů na tzv. systemd targets
- = I v nové verzi lze stále využít příkazu init ke změně runlevelu.
- = Pro ovládání a kontrolu nad systemd se využívá příkazu systemctl.
  - = Restart systému lze tedy provést buď pomocí zavolání runlevel 6 příkazem `init 6` nebo nově `systemctl isolate reboot.target`.
  - = Například `target /usr/lib/systemd/system/graphical.target`
  - = je ekvivalent runlevelu 5 (příkaz `init 5`), což představuje grafický režim.

- = Systemd targets je jeden z typů systemd units.
  - = Tyto units slouží jako konfigurační soubory například pro zmíněné systemd targets nebo pro služby (přípona `.service`), zařízení (přípona `.device`), přípojný body (přípona `.mount`) a mnohé další části systému.
  - = Target je tedy konfigurační soubor, ve kterém je definováno, jaký režim má na starosti, jaké další units musí být aktivovány, či s jakými je naopak v rozporu.



- = Kromě standardních procesů, které v Linuxu fungují, existuje i speciální typ, tzv. démon.
- = Jedná se o typ procesu, který (nejčastěji) běží od startu systému po jeho ukončení.
- = Probíhá bez povšimnutí na pozadí a čeká na obslužení nějaké události, pro kterou je naprogramován.
- = Důležité je, že probíhá na pozadí, i když v dané chvíli není přihlášen žádný uživatel.
- = Vyznačuje se tím, že PID jeho rodiče je jedna.
  - = V Ubuntu 15.04 je tedy jeho předek systemd, který má vždy PID jedna (dříve init proces)



- = Systemd je démon určený pro správu systému a služeb na novějších vybraných distribucích Linuxu.
- = Nahrazuje starší init, který byl do té doby široce používán, a mnozí uživatelé mají s touto evoluční změnou problém.
  - = Změna = čas na přeučení.
  - = Novější démon totiž vyžaduje jiné metody a přístupy v používání.



- = Systemd, stejně jako kdysi init, je rodič všech dalších procesů, protože je spuštěn hned po inicializaci jádra a stejně jako jeho předchůdce má PID jedna.
- = Pokud tedy dojde k problémům při startu tohoto démona, nedojde ke spuštění dalších navazujících procesů a systém vyvolá tzv. „Kernel panic“ (ve Windows zvaný jako „Blue Screen of Death“).
- = Když skončí nějaký rodič procesu, stává se jeho novým rodičem právě systemd, a až je potřeba, postará se o jeho odklizení.



- = Mezi vybrané funkcionality systemd patří:
  - = Nový logovací systém tzv. journal
  - = Systémové zprávy se nyní ukládají pomocí systemd journal a to včetně událostí ze startu systému.
  - = Defaultně jsou uloženy v adresáři `/run/log/journal`, ale je možné je ukládat do standardního `/var/log` adresáře.
  - = Stačí vytvořit nový adresář `journal` ve `/var/log` a restartovat `systemd-journald`.

```
[root@ubuntu ~]# mkdir /var/log/journal
```

```
[root@ubuntu ~]# systemctl restart systemd-journald
```



= Následně dojde k vytvoření nového adresáře a v něm je uložen log.

```
[root@ubuntu /]# ll
var/log/journal/c952d245df494af0bd0acb75aa888d60/
total 8196
-rw-r----- 1 root root 8388608 Aug 17 14:31
system.journal
```





- = Log lze prohlížet pomocí příkazu `journalctl`. Ukázka výpisu pomocí zmíněného příkazu

```
Aug 17 14:24:21 ubuntu.local systemd-logind[612]: New session 1 of user root.
```

```
Aug 17 14:24:21 ubuntu.local login[623]:  
pam_unix(login:session): session opened
```

```
Aug 17 14:24:21 ubuntu.local login[623]: ROOT LOGIN ON  
tty1
```



- = Například pokud se vypíše obsah souboru se službou sshd pomocí příkazu `cat` na terminál, zobrazí se tento text (jeho část).

```
root@ubuntu system]# cat
/usr/lib/systemd/system/sshd.service

[Unit]

Description=OpenSSH server daemon

After=network.target sshd-keygen.service
```



- = Z popisu na předchozím snímku je zřejmé, že se jedná o „OpenSSH server daemon“, ale důležitější je položka „After“.
- = Ta značí závislosti – v tomto případě že služba má být načtena až potom, co jsou připraveny položky uvedené u „After“.
- = To umožňuje pohodlné definování závislostí mezi jednotlivými službami.



- = Mezi vybrané funkcionality systemd patří:
  - = Náhrada runlevelů pomocí targets
    - = Místo číslo určitého runlevelu se nyní využívá jméno příslušného targetu.
  - = Start více služeb najednou (paralelní zpracování)
  - = Zpětná kompatibilita s SysV init skripty



- = Crond je démon, který slouží k automatizování opakovaných událostí (typicky zálohování).
- = Může probíhat každou minutu nebo pokaždé v neděli večer.
- = Vše záleží na tom, kde je spouštěcí skript umístěn.
  - = Může být například v adresáři:
    - = `/etc/cron.daily` nebo `/etc/cron.weekly`
    - = nebo pokud je potřeba individuální nastavení, tak v `/etc/cron.d`, kde lze plně přizpůsobit interval, kdy má být daný kód proveden.



- = V adresáři `/etc/cron.d` je například definovaná úloha `raid-check`, která má proběhnout v neděli v 1 hodinu ráno. Zde je výpis příslušného skriptu:

```
[root@ubuntu etc]# cat /etc/cron.d/raid-check  
  
# Run system wide raid-check once a week on Sunday at  
1am by default  
  
0 1 * * Sun root /usr/sbin/raid-check
```

- = Hvězdička značí jakoukoliv přípustnou hodnotu. Výše zmíněný kód tedy doslova říká, že má být spuštěn nultou minutu v jednu hodinu ráno libovolné neděle v libovolném měsíce. Syntaxe je následující:

```
# .----- minute (0 - 59)
# | .----- hour (0 - 23)
# | | .----- day of month (1 - 31)
# | | | .----- month (1 - 12) OR jan,feb,mar,apr ...
# | | | | .---- day of week(0 - 6) (Sunday=0 or 7)OR sun,mon,tue,...
# | | | | |
# * * * * * user-name  command to be executed
```



- = Kromě plánování zálohování ho lze využít například k restartu služeb, vytváření hlášení o stavu systému nebo spuštění libovolného skriptu.
- = Pro editaci úloh je vhodné využít příkaz `crontab -e`, který uživatele přepne do editoru a dle výše zmíněných pravidel si zde nakonfiguruje potřebné úlohy.
- = Ke zjištění úloh v `crontab` slouží příkaz `crontab -l`.





Univerzita Hradec Králové  
Fakulta informatiky a managementu

**Děkuji za pozornost**

Další téma: Systémová volání

