

CPU, RAM, kernel

Datum zpracování: 1.10.2023

Zpracovali: Kevin Daněš

Zadání

- CPU
 - /proc/cpuinfo
 - lscpu
- RAM
 - free
 - /proc/meminfo
 - vmstat; vmstat -s
- kernel
 - /proc/modules
 - lsmod
 - modinfo

úkoly

1. Pomocí výše uvedených zdrojů a příkazů zjistěte maximum údajů o CPU, relevantní část výstupů zkopírujte do elaborátu, souhrn okomentujte.
2. Pomocí výše uvedených zdrojů a příkazů zjistěte maximum údajů o RAM, relevantní část výstupů zkopírujte do elaborátu, souhrn okomentujte.
3. Pomocí výše uvedených zdrojů a příkazů zjistěte následující informace o jádru systému:
 1. instalované moduly - porovnejte výstupy z různých zdrojů
 2. moduly využívané modulem **i915**
 3. detailní informace o modulech **drm**, **i2c_core** a **video**

Veškeré postupy dokumentujte pomocí screenshotů nebo kopírujte stav terminálu do elaborátu. Elaborát prosím pro urychlení hodnocení odevzdávejte ve formátu PDF.

Úkol zpracujte individuálně!



Informace o CPU

- Soubor **/proc/cpuinfo** obsahuje informace o logických procesorech, se kterými jádro pracuje
- Příkaz **lscpu** zjišťuje informace o architektuře CPU právě ze souboru **/proc/cpuinfo** a dalších souborů, které poskytuje sysfs¹

Z příkazu **lscpu** můžeme například vyčíst, že

- procesor běží na 64 bitové x86 architektuře.
- procesor pracuje s malou endianitou, nebo-li na nejnižší adresu se ukládá nejméně významný byte
- model procesoru nese název Intel(R) Core(TM) i5-6400 CPU @ 2.70GHz
- na jedno jádro procesoru připadá jedno vlákno
- procesor pracuje s rychlostmi od 0.8GHz až do 3.3GHz

```
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Address sizes:         39 bits physical, 48 bits virtual
Byte Order:            Little Endian
CPU(s):                4
On-line CPU(s) list:   0-3
Vendor ID:              GenuineIntel
Model name:             Intel(R) Core(TM) i5-6400 CPU @ 2.70GHz
CPU family:            6
Model:                 94
Thread(s) per core:    1
Core(s) per socket:    4
Socket(s):              1
Stepping:               3
CPU max MHz:           3300,0000
CPU min MHz:           800,0000
BogoMIPS:               5399.81
```

Část výstupu příkazu lscpu

Pokud bychom začali zkoumat obsah souboru **/proc/cpuinfo**, zjistíme, že se tolik od obsahu výstupu příkazu **lscpu** neliší. Rozdíl je primárně v tom, že jsou informace děleny pro každý logický procesor zvlášť - můžeme tedy vidět rychlosti jednotlivých jader, jejich příznaky a kolik mají "sourozenců" - tedy dalších výpočetních jednotek v CPU.

¹ Jedná se o souborový systém poskytovaný jádrem, ve kterém se nachází informace o relevantních věcech pro jádro (např. informace o ovladačích, připojených perifériích či hardwaru)





```
processor      : 0
vendor_id     : GenuineIntel
cpu family    : 6
model         : 94
model name    : Intel(R) Core(TM) i5-6400 CPU @ 2.70GHz
stepping      : 3
microcode     : 0xf0
cpu MHz       : 800.030
cache size    : 6144 KB
physical id    : 0
siblings      : 4
core id       : 0
cpu cores     : 4
apicid        : 0
initial apicid : 0
fpu           : yes
fpu_exception : yes
cpuid level    : 22
wp            : yes
flags         : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr
pge mca cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm
pbe syscall nx pdpe1gb rdtscp lm constant_tsc art arch_perfmon pebs
bts rep_good nopl xtopology nonstop_tsc cpuid aperfmperf pni
pclmulqdq dtes64 monitor ds_cpl vmx est tm2 ssse3 sdbg fma cx16 xtpr
pdcml pcid sse4_1 sse4_2 x2apic movbe popcnt tsc_deadline_timer aes
xsave avx f16c rdrand lahf_lm abm 3dnowprefetch cpuid_fault epb
invpcid_single pti ssbd ibrs ibpb stibp tpr_shadow vnmi flexpriority
ept vpid ept_ad fsgsbase tsc_adjust bmi1 avx2 smep bmi2 erms invpcid
mpx rdseed adx smap clflushopt intel_pt xsaveopt xsavec xgetbv1
xsaves dtherm ida arat pln pts hwp hwp_notify hwp_act_window hwp_epp
md_clear flush_lld arch_capabilities
vmx flags     : vnmi preemption_timer invvpid ept_x_only ept_ad
ept_lgb flexpriority tsc_offset vtptr mtf vapic ept vpid
unrestricted_guest ple shadow_vmcs pml
bugs          : cpu_meltdown spectre_v1 spectre_v2
spec_store_bypass l1tf mds swapgs itlb_multihit srbds mmio_stale_data
retbleed
bogomips      : 5399.81
clflush size   : 64
cache_alignment : 64
address sizes  : 39 bits physical, 48 bits virtual
power management:
```

Ukázka informací o jednom logickém jádře v souboru /proc/cpuinfo



Informace o RAM

Příkaz **free** slouží k vypsání informací o kapacitě paměti a swapu - poskytuje rychlý přehled o celkově dostupné paměti, kolik jí je dostupné a kolik použité. Výstup příkazu **free** lze ovlivnit několika příznaky, v příkladu níže byly použity následující:

- **-h** pro formátování do lidmi čitelného formátu
- **-t** pro zobrazení řádku se souhrnem
- **--si** pro použití základu 1000 místo 1024 při výpočtu velikosti

V příkladu níže můžeme například vidět, že

- Celkově je v systému 17 GB paměti
- V době volání příkazu bylo využito uživatelem 524 MB paměti
- V době volání příkazu bylo volných 11GB
- Pro swap jsou dedikovány 2 gigabyty, které nejsou vůbec využity

```
kevin.danek@A0322:~$ free -h --si -t
              total        used        free      shared  buff/cache   available
Mem:           15G         524M          11G         83M           4,3G          14G
Swap:           2,0G           0B           2,0G
Total:          17G         524M          13G
```

Ukázka výstupu příkazu free

Soubor **/proc/meminfo** poskytuje podrobnější informace o využití paměti než příkaz **free**, podobně jako **cpuinfo** se jedná o soubor poskytovaný jádrem OS. Tento soubor obsahuje hodně metrik, které pro nás nejsou nyní relevantní, ovšem z některých z nich můžeme vyčíst podobné údaje, co z příkazu **free**. V tomto případě dokonce vidíme i rozdělení velikostí vyrovnávací paměti a cache, což jsou údaje, které jsou v příkazu **free** spojeny do jednoho.

```
kevin.danek@A0322:~$ cat /proc/meminfo
MemTotal:        16228968 kB
MemFree:         11287908 kB
MemAvailable:    15250120 kB
Buffers:          427360 kB
Cached:          3809100 kB
Active:          2726368 kB
...
```

Informace o jádře systému

Stejně jako pro CPU a paměť, i zde jádro OS poskytuje informace do souboru ve složce **/proc/** - jedná se konkrétně o soubor **/proc/modules**, který je v následujícím formátu:

- První sloupec označuje **název modulu**
- Druhý sloupec obsahuje **velikost modulu v bytech**
- Třetí sloupec obsahuje **počet instancí modulu**, tj. kolikrát je někde spuštěn
- Čtvrtý sloupec obsahuje (většinou neúplný) seznam závislých modulů.
 - Seznam je oddělen čárkami
 - Pokud modul žádné závislosti nemá, je místo něho pomlčka
- Pátý sloupec říká, v jakém je modul stavu (Live, Loading, ...)
- Šestý sloupec říká adresu v jádře, na kterém se modul nachází

```
kevin.danek@A0322:~$ cat /proc/modules
tls 147456 0 - Live 0x0000000000000000
rfcomm 98304 4 - Live 0x0000000000000000
nfsv3 61440 1 - Live 0x0000000000000000
rpcsec_gss_krb5 40960 0 - Live 0x0000000000000000
nfsv4 1077248 0 - Live 0x0000000000000000
nfs 544768 3 nfsv3,nfsv4, Live 0x0000000000000000
fscache 385024 1 nfs, Live 0x0000000000000000
netfs 61440 1 fscache, Live 0x0000000000000000
```

Ukázka souboru /proc/modules

Pro lépe formátovaný výstup můžeme použít příkaz **lsmod**, který sice odebere informaci o umístění v paměti jádra, avšak v běžných situacích tato informace stejně není potřeba:

```
kevin.danek@A0322:~$ lsmod
Module                Size  Used by
tls                   147456  0
rfcomm                 98304  4
nfsv3                  61440  1
rpcsec_gss_krb5       40960  0
nfsv4                 1077248  0
nfs                   544768  3 nfsv4,nfsv3
fscache               385024  1 nfs
netfs                 61440  1 fscache
```

Ukázka výstupu příkazu lsmod

Pokud chceme zkoumat konkrétní moduly, můžeme tak učinit pomocí příkazu **modinfo** (module info), kterému stačí před název modulu a vrátí nám kompletní výpis informací.

Závislosti modulu lze vyčíst z řádku, který začíná slovem “depends” (závisí). Například u modulu **i915** nám lsmod žádnou informaci o závislostech nepoví, ovšem modinfo již tuto informaci obsahuje.

```
kevin.danek@A0322:~$ modinfo i915 | grep depends
depends:
ttm,drm,drm_display_helper,drm_kms_helper,video,cec,drm_buddy,i2c-
algo-bit
```

Pole depends ve výstupu příkazu modinfo



Informace o vybraných modulech

Vybrané moduly jsou DRM (Direct Rendering Manager), modul pro sběrnici I2C a modul pro video.

```
kevin.danek@A0322:~$ modinfo drm
filename:          /lib/modules/6.2.0-33-generic/kernel/drivers/gpu/drm
/drm.ko
license:          GPL and additional rights
description:      DRM bridge infrastructure
author:           Ajay Kumar <ajaykumar.rs@samsung.com>
license:          GPL and additional rights
description:      DRM shared core routines
author:           Gareth Hughes, Leif Delgass, José Fonseca, Jon Smirl
import_ns:        DMA_BUF
license:          GPL and additional rights
description:      DRM panel infrastructure
author:           Thierry Reding <treding@nvidia.com>
srcversion:       FB874D1ACD4ACA3BAA44DD9
```

```
kevin.danek@A0322:~$ modinfo i2c_core
name:             i2c_core
filename:         (builtin)
license:          GPL
file:             drivers/i2c/i2c-core
description:      I2C-Bus main module
author:           Simon G. Vogl <simon@tk.uni-linz.ac.at>
```

```
kevin.danek@A0322:~$ modinfo video
filename:          /lib/modules/6.2.0-33-generic/kernel/drivers
/acpi/video.ko
license:          GPL
description:      ACPI Video Driver
author:           Bruno Ducrot
srcversion:       FBF3CDB89BC82DDD96455A8
alias:            acpi*:LNXVIDEO:*
depends:           wmi
retpoline:        Y
intree:           Y
name:             video
vermagic:         6.2.0-33-generic SMP preempt mod_unload modversions
```



Závěr

V linuxových distribucích je spousta souborů, které obsahují informace o daném zařízení a systému. Nad těmito soubory, které jsou často nečitelné pro člověka kvůli své hustotě informací, jsou vybudovány utility programy, které tyto soubory čtou a zpracovávají do lidmi čitelného formátu. Tento přístup je skvělý v tom, že pro rychlou diagnostiku jsou k dispozici tyto triviální programy, ale zároveň existuje jednoduchý přístup k datům pro další aplikace, které mohou vykonávat komplexnější úkony a nemusí se daný vývojář babrat se systémovými voláními či být omezen low-level systémovou knihovnou napsaném v konkrétním jazyce.