

Programowanie Systemowe

Laboratorium 2 - Wprowadzenie

Magdalena Pastuła

Do wykonania ćwiczenia wykorzystano dostarczoną maszynę wirtualną z systemem Fedora w wersji 25.

1. LXR i dokumentacja.

1.1 Znajdź definicję struktury `file_operations`. Zapoznaj się z jej polami.

Znaleziona definicja struktury:

```
struct file_operations {
    struct module *owner;
    loff_t (*llseek) (struct file *, loff_t, int);
    ssize_t (*read) (struct file *, char __user *, size_t, loff_t *);
    ssize_t (*write) (struct file *, const char __user *, size_t, loff_t
*);
    ssize_t (*read_iter) (struct kiocb *, struct iov_iter *);
    ssize_t (*write_iter) (struct kiocb *, struct iov_iter *);
    int (*iopoll)(struct kiocb *kiocb, bool spin);
    int (*iterate) (struct file *, struct dir_context *);
    int (*iterate_shared) (struct file *, struct dir_context *);
    __poll_t (*poll) (struct file *, struct poll_table_struct *);
    long (*unlocked_ioctl) (struct file *, unsigned int, unsigned long);
    long (*compat_ioctl) (struct file *, unsigned int, unsigned long);
    int (*mmap) (struct file *, struct vm_area_struct *);
    unsigned long mmap_supported_flags;
    int (*open) (struct inode *, struct file *);
    int (*flush) (struct file *, fl_owner_t id);
    int (*release) (struct inode *, struct file *);
    int (*fsync) (struct file *, loff_t, loff_t, int datasync);
    int (*fasync) (int, struct file *, int);
    int (*lock) (struct file *, int, struct file_lock *);
    ssize_t (*sendpage) (struct file *, struct page *, int, size_t, loff_t
*, int);
    unsigned long (*get_unmapped_area)(struct file *, unsigned long,
unsigned long, unsigned long, unsigned long);
    int (*check_flags)(int);
    int (*flock) (struct file *, int, struct file_lock *);
    ssize_t (*splice_write)(struct pipe_inode_info *, struct file *, loff_t
*, size_t, unsigned int);
    ssize_t (*splice_read)(struct file *, loff_t *, struct pipe_inode_info
*, size_t, unsigned int);
    int (*setlease)(struct file *, long, struct file_lock **, void **);
    long (*fallocate)(struct file *file, int mode, loff_t offset,
                    loff_t len);
    void (*show_fdinfo)(struct seq_file *m, struct file *f);
#ifdef CONFIG_MMU
```

```

    unsigned (*mmap_capabilities)(struct file *);
#endif
    ssize_t (*copy_file_range)(struct file *, loff_t, struct file *,
                               loff_t, size_t, unsigned int);
    loff_t (*remap_file_range)(struct file *file_in, loff_t pos_in,
                              struct file *file_out, loff_t pos_out,
                              loff_t len, unsigned int remap_flags);
    int (*fadvise)(struct file *, loff_t, loff_t, int);
} __randomize_layout;

```

1.2 Spróbuj odnaleźć użycie tej struktury w systemie plików `hostfs`.

Znalezione użycie struktury `file_operations` w systemie plików `hostfs` w pliku `hostfs_kern.c`:

```

static const struct file_operations hostfs_file_fops = {
    .llseek      = generic_file_llseek,
    .splice_read = generic_file_splice_read,
    .read_iter   = generic_file_read_iter,
    .write_iter  = generic_file_write_iter,
    .mmap        = generic_file_mmap,
    .open        = hostfs_open,
    .release     = hostfs_file_release,
    .fsync       = hostfs_fsync,
};

static const struct file_operations hostfs_dir_fops = {
    .llseek      = generic_file_llseek,
    .iterate_shared = hostfs_readdir,
    .read        = generic_read_dir,
    .open        = hostfs_open,
    .fsync       = hostfs_fsync,
};

```

1.3 Czy potrafisz zinterpretować zastosowanie poszczególnych pól?

Poszczególne pola odpowiadają operacjom możliwym do wykonania na pliku (`hostfs_file_fops`) i katalogu (`hostfs_dir_fops`). Możliwe operacje to między innymi:

- `llseek` -> ustawienie wskaźnika w pliku
- `read_iter` -> czytanie z pliku
- `write_iter` -> pisanie do pliku
- `open` -> otwarcie pliku
- `release` -> zamknięcie pliku

1.4 Zlokalizuj w dokumentacji (katalog `Documentation` lub wyszukiwarka) informacje o tym, jak przechodzić po strukturze katalogów w jądrze.

Głównym plikiem opisującym strukturę katalogów w jądrze jest plik `/filesystems/path_lookup.txt`.

1.5 Zlokalizuj plik `namei.h` a w nim funkcje `kern_path` i `user_path`. Czym się one różnią i kiedy mają zastosowanie?

Definicja funkcji `kern_path`:

```
int kern_path(const char *name, unsigned int flags, struct path *path)
{
    return filename_lookup(AT_FDCWD, getname_kernel(name),
                          flags, path, NULL);
}
```

Definicja funkcji `user_path`:

```
static inline int user_path_at(int dfd, const char __user *name,
                              unsigned flags, struct path *path)
{
    return user_path_at_empty(dfd, name, flags, path, NULL);
}
```

Funkcje `kern_path` oraz `user_path` wyszukują ścieżkę do pliku podanego w argumencie `name`. W tym celu funkcje te wyszukują strukturę `dentry`, która odpowiada danemu plikowi, a następnie zwracają tę strukturę poprzez argument `path`. Dodatkowo, zwiększany jest licznik odwołań do tych struktur.

Różnicą między tymi funkcjami jest środowisko: `kern_path` wyszukuje dla łańcucha znajdującego się w pamięci jądra, natomiast `user_path` dla łańcucha znajdującego się w pamięci użytkownika.

1.6 Znajdź definicję struktury `dentry`.

Znaleziona definicja struktury `dentry`:

```
struct dentry {
    /* RCU lookup touched fields */
    unsigned int d_flags;           /* protected by d_lock */
    seqcount_t d_seq;              /* per dentry seqlock */
    struct hlist_bl_node d_hash;    /* lookup hash list */
    struct dentry *d_parent;        /* parent directory */
    struct qstr d_name;
    struct inode *d_inode;          /* Where the name belongs to - NULL is
                                   * negative */
    unsigned char d_iname[DNAME_INLINE_LEN]; /* small names */
    /* Ref lookup also touches following */
    struct lockref d_lockref;       /* per-dentry lock and refcount */
    const struct dentry_operations *d_op;
    struct super_block *d_sb;       /* The root of the dentry tree */
    unsigned long d_time;           /* used by d_revalidate */
    void *d_fsdata;                /* fs-specific data */

    union {
```

```

        struct list_head d_lru;          /* LRU list */
        wait_queue_head_t *d_wait;      /* in-lookup ones only */
    };
    struct list_head d_child;            /* child of parent list */
    struct list_head d_subdirs;          /* our children */
    /*
     * d_alias and d_rcu can share memory
     */
    union {
        struct hlist_node d_alias;       /* inode alias list */
        struct hlist_bl_node d_in_lookup_hash; /* only for in-lookup
ones */
        struct rcu_head d_rcu;
    } d_u;
} __randomize_layout;

```

1.7 Co robi funkcja `dget` i po co? Znalezionej definicji funkcji `dget`:

```

/**
 *      dget, dget_dlock -      get a reference to a dentry
 *      @dentry: dentry to get a reference to
 *
 *      Given a dentry or %NULL pointer increment the reference count
 *      if appropriate and return the dentry. A dentry will not be
 *      destroyed when it has references.
 */
static inline struct dentry *dget_dlock(struct dentry *dentry)
{
    if (dentry)
        dentry->d_lockref.count++;
    return dentry;
}

static inline struct dentry *dget(struct dentry *dentry)
{
    if (dentry)
        lockref_get(&dentry->d_lockref);
    return dentry;
}

```

Jak wskazuje opis, funkcja ta zwiększa liczbę odwołań do struktury. Dzięki nowym odwołaniom struktura `dentry` nie zostanie usunięta.

2. Kompilacja jądra.

2.1 Zapoznaj się z plikiem `.config`.

Plik opisuje oraz zawiera konfigurację jądra.

2.2 Wykonaj polecenie `make help`.

Po wykonaniu polecenia `make help` wypisuje się przewodnik po Makefile'a do budowy jądra.

2.3 Co robi polecenie `make oldconfig`?

Polecenie `make oldconfig` dokonuje aktualizacji obecnej konfiguracji korzystając z pliku `.config`.

2.4 Co robi polecenie `make menuconfig`?

Polecenie `make menuconfig` wyświetla menu graficzne, za pomocą którego można uaktualnić konfigurację w pliku `.config`.

2.5 Ustaw dowolną, ale charakterystyczną wersję lokalną (`CONFIG_LOCALVERSION` albo `General setup/Local version`).

Ustawiono `JanTrzeci`.

2.6 Zrób małą zmianę w konfiguracji (np. włącz obsługę któregoś systemu plików).

Dodano obsługę systemu plików F2FS.

2.7 Co robi polecenie `make all`?

Polecenie `make all` buduje wszystkie targety oznaczone gwiazdką, czyli `vmlinux` i `modules` i skompresowane jądro (`bzImage`).

2.8 Zmierz czas kompilacji jądra po modyfikacji. Jakie informacje się wyświetlają podczas kompilacji?

Podczas kompilacji w terminalu wypisują się kolejno kompilowane pliki i moduły.

Zmierzony czas kompilacji: real 62m 38.067s user 194m 32.753s sys 30m 9.810s

2.9 Co robią polecenia `make modules_install` i `make install`?

Polecenie `make modules_install` instaluje wszystkie moduły jądra, natomiast komenda `make install` instaluje jądro w aktualnym systemie.

2.10 Zainstaluj jądro w systemie.

Wykonano komendy `make modules_install` oraz `make install`.

2.11 Zrestartuj system i uruchom nowe jądro.

Po restarcie i wpisaniu komendy `uname -a` wypisuje się następująca linia:

```
Linux ps2017 4.10.0JanTrzeci #1 SMP Fri Nov 13 09:04:03 CET 2020 x86_64
x86_64 x86_64 GNU/Linux
```

Jak widać, została zmieniona data ostatniej kompilacji oraz wypisuje się nazwa wpisana w `Local version`.

3. Moduł `trivial_module`.

Wynik po kompilacji modułu:

```

make -C /lib/modules/4.10.0JanTrzeci/build
M=/home/student/Downloads/trivial_module/intro/trivial_module modules
make[1]: Entering directory '/home/student/linux'
  CC [M]
/home/student/Downloads/trivial_module/intro/trivial_module/trivial.o
  Building modules, stage 2.
  MODPOST 1 modules
  CC
/home/student/Downloads/trivial_module/intro/trivial_module/trivial.mod.o
  LD [M]
/home/student/Downloads/trivial_module/intro/trivial_module/trivial.ko
make[1]: Leaving directory '/home/student/linux'

```

3.1 Załaduj `insmod`.

Załadowano stosując instrukcję `insmod trivial.ko`

3.2 Wylistuj moduły.

Wypisane moduły:

Module	Size	Used by
trivial	16384	0
xt_CHECKSUM	16384	1
ipt_MASQUERADE	16384	3
nf_nat_masquerade_ipv4	16384	1 ipt_MASQUERADE
nf_conntrack_netbios_ns	16384	1
nf_conntrack_broadcast	16384	1 nf_conntrack_netbios_ns
xt_CT	16384	1
ip6t_rpfilter	16384	1
ip6t_REJECT	16384	2
nf_reject_ipv6	16384	1 ip6t_REJECT
xt_conntrack	16384	22
ip_set	36864	0
nfnetlink	16384	1 ip_set
ebtable_nat	16384	1
ebtable_broute	16384	1
bridge	135168	1 ebtable_broute
stp	16384	1 bridge
llc	16384	2 bridge, stp
ip6table_nat	16384	1
nf_conntrack_ipv6	20480	12
nf_defrag_ipv6	36864	1 nf_conntrack_ipv6
nf_nat_ipv6	16384	1 ip6table_nat
ip6table_raw	16384	1
ip6table_mangle	16384	1
ip6table_security	16384	1
iptable_nat	16384	1
nf_conntrack_ipv4	16384	16
nf_defrag_ipv4	16384	1 nf_conntrack_ipv4
nf_nat_ipv4	16384	1 iptable_nat

```

nf_nat                28672   3
nf_nat_ipv6,nf_nat_masquerade_ipv4,nf_nat_ipv4
nf_conntrack          131072  11
nf_conntrack_ipv6,nf_conntrack_ipv4,ipt_MASQUERADE,nf_conntrack_broadcast,n
f_conntrack_netbios_ns,xt_CT,nf_nat_ipv6,nf_nat_masquerade_ipv4,xt_conntrac
k,nf_nat_ipv4,nf_nat
libcrc32c             16384   1 nf_nat
iptables_raw          16384   1
iptables_mangle       16384   1
iptables_security     16384   1
ebtable_filter        16384   1
ebtables              36864   3 ebtable_filter,ebtable_nat,ebtable_broute
ip6table_filter       16384   1
ip6_tables            28672   5
ip6table_mangle,ip6table_filter,ip6table_security,ip6table_raw,ip6table_nat
vmw_vsock_vmci_transport 28672   2
vsock                36864   3 vmw_vsock_vmci_transport
snd_seq_midi          16384   0
snd_seq_midi_event    16384   1 snd_seq_midi
kvm_intel             196608  0
kvm                   593920  1 kvm_intel
snd_ens1371           28672   5
snd_rawmidi           32768   2 snd_seq_midi,snd_ens1371
snd_ac97_codec        131072   1 snd_ens1371
ac97_bus              16384   1 snd_ac97_codec
snd_seq               65536   2 snd_seq_midi_event,snd_seq_midi
irqbypass            16384   1 kvm
crct10dif_pclmul      16384   0
snd_seq_device        16384   3 snd_seq,snd_rawmidi,snd_seq_midi
snd_pcm              118784   2 snd_ac97_codec,snd_ens1371
ppdev                 20480   0
crc32_pclmul          16384   0
snd_timer             32768   2 snd_seq,snd_pcm
vmw_balloon           20480   0
ghash_clmulni_intel   16384   0
joydev                20480   0
snd                   81920   17
snd_seq,snd_ac97_codec,snd_timer,snd_rawmidi,snd_ens1371,snd_seq_device,snd
_pcm
pcspkr                16384   0
soundcore             16384   1 snd
gameport              16384   1 snd_ens1371
nfit                  49152   0
parport_pc            28672   0
acpi_cpufreq          20480   0
tpm_tis               16384   0
tpm_tis_core          20480   1 tpm_tis
tpm                   40960   2 tpm_tis,tpm_tis_core
shpchp                36864   0
nfsd                  335872  1
vmw_vmci              69632   2 vmw_balloon,vmw_vsock_vmci_transport
i2c_piix4             24576   0
auth_rpcgss           61440   1 nfsd
nfs_acl               16384   1 nfsd

```

lockd	94208	1	nfsd
grace	16384	2	nfsd, lockd
sunrpc	331776	7	auth_rpcgss, nfsd, nfs_acl, lockd
vmwgfx	241664	9	
drm_kms_helper	155648	1	vmwgfx
ttm	98304	1	vmwgfx
drm	352256	12	vmwgfx, ttm, drm_kms_helper
e1000	143360	0	
mptspi	24576	2	
crc32c_intel	24576	1	
scsi_transport_spi	32768	1	mptspi
mptscsih	40960	1	mptspi
ata_generic	16384	0	
serio_raw	16384	0	
pata_acpi	16384	0	
mptbase	102400	2	mptscsih, mptspi
fjes	73728	0	

3.3 Obejrzyj komunikaty jądra: `dmesg`.

Ostatnie kilka linii komunikatów jądra:

```
[ 16.589857] audit: type=1131 audit(1605264201.757:68): pid=1 uid=0
auid=4294967295 ses=4294967295 subj=system_u:system_r:init_t:s0
msg='unit=plymouth-read-write comm="systemd" exe="/usr/lib/systemd/systemd"
hostname=? addr=? terminal=? res=success'
[ 17.953577] NET: Registered protocol family 40
[ 22.487106] ip6_tables: (C) 2000-2006 Netfilter Core Team
[ 22.816639] Ebtables v2.0 registered
[ 23.855172] nf_conntrack version 0.5.0 (16384 buckets, 65536 max)
[ 24.081306] IPv6: ADDRCONF(NETDEV_UP): ens33: link is not ready
[ 24.092062] e1000: ens33 NIC Link is Up 1000 Mbps Full Duplex, Flow
Control: None
[ 25.477558] bridge: filtering via arp/ip/ip6tables is no longer
available by default. Update your scripts to load br_netfilter if you need
this.
[ 25.766839] Netfilter messages via NETLINK v0.30.
[ 25.871808] ip_set: protocol 6
[ 27.892680] virbr0: port 1(virbr0-nic) entered blocking state
[ 27.892683] virbr0: port 1(virbr0-nic) entered disabled state
[ 27.892818] device virbr0-nic entered promiscuous mode
[ 28.392613] virbr0: port 1(virbr0-nic) entered blocking state
[ 28.392618] virbr0: port 1(virbr0-nic) entered listening state
[ 28.501450] virbr0: port 1(virbr0-nic) entered disabled state
[ 64.426793] perf: interrupt took too long (2505 > 2500), lowering
kernel.perf_event_max_sample_rate to 79000
[ 120.126454] perf: interrupt took too long (3138 > 3131), lowering
kernel.perf_event_max_sample_rate to 63000
[ 160.773030] perf: interrupt took too long (3951 > 3922), lowering
kernel.perf_event_max_sample_rate to 50000
[ 216.153396] perf: interrupt took too long (5051 > 4938), lowering
kernel.perf_event_max_sample_rate to 39000
```



```
[ 274.723108] perf: interrupt took too long (6483 > 6313), lowering
kernel.perf_event_max_sample_rate to 30000
[ 630.758485] perf: interrupt took too long (8143 > 8103), lowering
kernel.perf_event_max_sample_rate to 24000
[ 936.077694] perf: interrupt took too long (10195 > 10178), lowering
kernel.perf_event_max_sample_rate to 19000
[ 1106.225203] perf: interrupt took too long (12821 > 12743), lowering
kernel.perf_event_max_sample_rate to 15000
[ 1224.936587] trivial: loading out-of-tree module taints kernel.
[ 1224.936956] trivial: module verification failed: signature and/or
required key missing - tainting kernel
[ 1224.944378] Hello world! I'm a trivial module!
```

3.4 Usunąć: `rmmod`.

Wykonano komendę `rmmod trivial`.

3.5 Jeszcze raz obejrzyj komunikaty jądra.

Ostatnich kilka linii komunikatów jądra:

```
[ 16.589857] audit: type=1131 audit(1605264201.757:68): pid=1 uid=0
auid=4294967295 ses=4294967295 subj=system_u:system_r:init_t:s0
msg='unit=plymouth-read-write comm="systemd" exe="/usr/lib/systemd/systemd"
hostname=? addr=? terminal=? res=success'
[ 17.953577] NET: Registered protocol family 40
[ 22.487106] ip6_tables: (C) 2000-2006 Netfilter Core Team
[ 22.816639] Ebtables v2.0 registered
[ 23.855172] nf_conntrack version 0.5.0 (16384 buckets, 65536 max)
[ 24.081306] IPv6: ADDRCONF(NETDEV_UP): ens33: link is not ready
[ 24.092062] e1000: ens33 NIC Link is Up 1000 Mbps Full Duplex, Flow
Control: None
[ 25.477558] bridge: filtering via arp/ip/ip6tables is no longer
available by default. Update your scripts to load br_netfilter if you need
this.
[ 25.766839] Netfilter messages via NETLINK v0.30.
[ 25.871808] ip_set: protocol 6
[ 27.892680] virbr0: port 1(virbr0-nic) entered blocking state
[ 27.892683] virbr0: port 1(virbr0-nic) entered disabled state
[ 27.892818] device virbr0-nic entered promiscuous mode
[ 28.392613] virbr0: port 1(virbr0-nic) entered blocking state
[ 28.392618] virbr0: port 1(virbr0-nic) entered listening state
[ 28.501450] virbr0: port 1(virbr0-nic) entered disabled state
[ 64.426793] perf: interrupt took too long (2505 > 2500), lowering
kernel.perf_event_max_sample_rate to 79000
[ 120.126454] perf: interrupt took too long (3138 > 3131), lowering
kernel.perf_event_max_sample_rate to 63000
[ 160.773030] perf: interrupt took too long (3951 > 3922), lowering
kernel.perf_event_max_sample_rate to 50000
[ 216.153396] perf: interrupt took too long (5051 > 4938), lowering
kernel.perf_event_max_sample_rate to 39000
[ 274.723108] perf: interrupt took too long (6483 > 6313), lowering
```

```
kernel.perf_event_max_sample_rate to 30000
[ 630.758485] perf: interrupt took too long (8143 > 8103), lowering
kernel.perf_event_max_sample_rate to 24000
[ 936.077694] perf: interrupt took too long (10195 > 10178), lowering
kernel.perf_event_max_sample_rate to 19000
[ 1106.225203] perf: interrupt took too long (12821 > 12743), lowering
kernel.perf_event_max_sample_rate to 15000
[ 1224.936587] trivial: loading out-of-tree module taints kernel.
[ 1224.936956] trivial: module verification failed: signature and/or
required key missing - tainting kernel
[ 1224.944378] Hello world! I'm a trivial module!
[ 1432.358171] perf: interrupt took too long (16117 > 16026), lowering
kernel.perf_event_max_sample_rate to 12000
[ 1593.403846] Oh no, why are you doing this to me? Argh!
```

Komenda `dmseg` to komenda służąca do wyświetlenia zawartości lub zmiany bufora kołowego jądra.

4. Uruchomianie jądra w QEMU.

4.1 Skompiluj jądro, podobnie jak we wcześniejszym zadaniu.

Przed kompilacją zmieniono nazwę wersji lokalnej na `QEMU`.

4.2 Uruchom:

```
qemu-system-x86_64 \
  -kernel arch/x86/boot/bzImage \
  -hda ~/fs/CentOS6.x-AMD64-root_fs \
  -append 'root=/dev/sda'
```

4.3 Zaloguj się do systemu. Login: root, hasło: puste.

Po zalogowaniu się i wykonaniu komendy `uname -a` ukazuje się następująca informacja:

```
Linux localhost.localdomian 4.10.0QEMU #2 SMP Fri Nov 13 12:21:26 CET 2020
x86_64 x86_64 GNU/Linux
```

Zatem widać, że w QEMU została uruchomiona skompilowana wersja jądra.

4.4 Zakończ pracę z gościem.

Aby zakończyć pracę wykonano komendę `poweroff`. Komenda `exit` powoduje tylko wylogowanie się.

5. Kompilacja jądra UML.

5.1 Wykonaj komendę `make ARCH=um defconfig`. Co ona robi?

Informacje wypisujące się po wykonaniu komendy:

```
*** Default configuration is based on 'x86_64_defconfig'
kernel/time/Kconfig:155:warning: range is invalid
#
# configuration written to .config
#
```

Komenda ta zmienia konfigurację jądra na domyślną dla architektury wyszczególnionej w argumencie ARCH. W tym przypadku architekturą tą jest tryb użytkownika.

5.2 Skompiluj jądro komendą `make ARCH=um`. Ile czasu zajęła kompilacja?

Czas kompilacji: real 1m28.880s user 4m25.051s sys 0m42.474s

5.3 Uruchom `./vmlinux ubd0=~/.fs/CentOS6.x-AMD64-root_fs`.

Po uruchomieniu pojawia się informacja:

```
CentOS release 6.6 (Final)
Kernel 4.10.0 on an x86_64
```

5.4 Zaloguj się do systemu. Login i hasło jak poprzednio.

Po zalogowaniu się i wpisaniu komendy `uname -a` pokazuje się następująca informacja:

```
Linux localhost.localdomain 4.10.0 #3 Fri Nov 13 12:58:10 CET 2020 x86_64
x86_64 x86_64 GNU/Linux
```

Ponownie, widać, że środowisko zostało uruchomione ze skompilowanego jądra.

5.5 Zamontuj hostfs za pomocą komendy `mount none /host -t hostfs` Co pojawiło się po zamontowaniu systemu plików?

Po wpisaniu komendy nic się nie wypisuje w terminalu, natomiast w folderze `/host` pojawia się następująca zawartość:

```
1  boot  etc  lib  lost+found  mnt  proc  run  srv  tmp  var
bin  dev  home  lib64  media  opt  root  sbin  sys  usr
```

Po każdej komendzie dodatkowo wypisuje się informacja o błędzie:

```
modprobe: FATAL: Could not load /lib/modules/4.10.0/modules.dep: No such
file or directory
```

5.6 Jakie procesy są widoczne w gościu i po stronie hosta?

Procesy widoczne w gościu:

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
root	1	0.0	8.2	19284	2256	?	Ss	07:03	0:00	/sbin/init
root	2	0.0	0.0	0	0	?	S	07:03	0:00	[kthreadd]
root	3	0.0	0.0	0	0	?	S	07:03	0:00	
[kworker/0:0]										
root	4	0.0	0.0	0	0	?	S<	07:03	0:00	
[kworker/0:0H]										
root	6	0.0	0.0	0	0	?	S	07:03	0:00	
[ksoftirqd/0]										
root	7	0.0	0.0	0	0	?	S<	07:03	0:00	[lru-add-
drain]										
root	8	0.0	0.0	0	0	?	S	07:03	0:00	
[kdevtmpfs]										
root	9	0.0	0.0	0	0	?	S<	07:03	0:00	[netns]
root	10	0.0	0.0	0	0	?	S	07:03	0:00	
[oom_reaper]										
root	11	0.0	0.0	0	0	?	S	07:03	0:00	
[kworker/u2:1]										
root	45	0.0	0.0	0	0	?	S<	07:03	0:00	
[writeback]										
root	46	0.0	0.0	0	0	?	S<	07:03	0:00	[crypto]
root	47	0.0	0.0	0	0	?	S	07:03	0:00	
[kworker/0:1]										
root	49	0.0	0.0	0	0	?	S<	07:03	0:00	[bioset]
root	51	0.0	0.0	0	0	?	S<	07:03	0:00	[kblockd]
root	73	0.0	0.0	0	0	?	S	07:03	0:00	[kswapd0]
root	74	0.0	0.0	0	0	?	S<	07:03	0:00	[bioset]
root	227	0.0	0.0	0	0	?	S<	07:03	0:00	[bioset]
root	236	0.0	0.0	0	0	?	S	07:03	0:00	
[jbd2/ubda-8]										
root	237	0.0	0.0	0	0	?	S<	07:03	0:00	[ext4-rsv-
conver]										
root	305	0.0	5.4	10700	1488	?	S<s	07:03	0:00	
/sbin/udev -d										
root	400	0.0	5.2	10696	1432	?	S<	07:03	0:00	
/sbin/udev -d										
root	406	0.0	0.0	0	0	?	S<	07:03	0:00	
[kworker/0:1H]										
root	576	0.0	9.1	66268	2500	?	Ss	07:03	0:00	
/usr/sbin/sshd										
root	589	0.0	9.1	52352	2492	?	Ss	07:03	0:00	login --
root										
root	595	0.0	9.6	11488	2632	tty0	Ss	07:03	0:00	-bash
root	606	0.0	0.0	0	0	?	S	07:06	0:00	
[kworker/u2:2]										
root	637	0.0	6.8	13372	1860	tty0	R+	07:16	0:00	ps aux

Procesy widoczne w hoście:

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
root	1	0.1	0.2	149308	6908	?	Ss	11:43	0:07	/usr/lib/system
root	2	0.0	0.0	0	0	?	S	11:43	0:00	[kthreadd]
root	4	0.0	0.0	0	0	?	S<	11:43	0:00	[kworker/0:0H]
root	6	0.0	0.0	0	0	?	S	11:43	0:02	[ksoftirqd/0]
root	7	0.0	0.0	0	0	?	S	11:43	0:05	[rcu_sched]
root	8	0.0	0.0	0	0	?	S	11:43	0:00	[rcu_bh]
root	9	0.1	0.0	0	0	?	S	11:43	0:06	[rcuos/0]
root	10	0.0	0.0	0	0	?	S	11:43	0:00	[rcuob/0]
root	11	0.0	0.0	0	0	?	S	11:43	0:00	[migration/0]
root	12	0.0	0.0	0	0	?	S<	11:43	0:00	[lru-add-drain]
root	13	0.0	0.0	0	0	?	S	11:43	0:00	[watchdog/0]
root	14	0.0	0.0	0	0	?	S	11:43	0:00	[cpuhp/0]
root	15	0.0	0.0	0	0	?	S	11:43	0:00	[cpuhp/1]
root	16	0.0	0.0	0	0	?	S	11:43	0:00	[watchdog/1]
root	17	0.0	0.0	0	0	?	S	11:43	0:00	[migration/1]
root	18	0.0	0.0	0	0	?	S	11:43	0:00	[ksoftirqd/1]
root	20	0.0	0.0	0	0	?	S<	11:43	0:00	[kworker/1:0H]
root	21	0.0	0.0	0	0	?	S	11:43	0:01	[rcuos/1]
root	22	0.0	0.0	0	0	?	S	11:43	0:00	[rcuob/1]
root	23	0.0	0.0	0	0	?	S	11:43	0:00	[cpuhp/2]
root	24	0.0	0.0	0	0	?	S	11:43	0:00	[watchdog/2]
root	25	0.0	0.0	0	0	?	S	11:43	0:00	[migration/2]
root	26	0.0	0.0	0	0	?	S	11:43	0:01	[ksoftirqd/2]
root	28	0.0	0.0	0	0	?	S<	11:43	0:00	[kworker/2:0H]
root	29	0.0	0.0	0	0	?	S	11:43	0:01	[rcuos/2]
root	30	0.0	0.0	0	0	?	S	11:43	0:00	[rcuob/2]
root	31	0.0	0.0	0	0	?	S	11:43	0:00	[cpuhp/3]
root	32	0.0	0.0	0	0	?	S	11:43	0:00	[watchdog/3]
root	33	0.0	0.0	0	0	?	S	11:43	0:00	[migration/3]
root	34	0.0	0.0	0	0	?	S	11:43	0:01	[ksoftirqd/3]
root	36	0.0	0.0	0	0	?	S<	11:43	0:00	[kworker/3:0H]

```

root      37  0.0  0.0      0  0 ?      S   11:43  0:01 [rcuos/3]
root      38  0.0  0.0      0  0 ?      S   11:43  0:00 [rcuob/3]
root      39  0.0  0.0      0  0 ?      S   11:43  0:00
[kdevtmpfs]
root      40  0.0  0.0      0  0 ?      S<  11:43  0:00 [netns]
root      41  0.0  0.0      0  0 ?      S   11:43  0:00
[oom_reaper]
root      42  0.0  0.0      0  0 ?      S<  11:43  0:00
[writeback]
root      43  0.0  0.0      0  0 ?      S   11:43  0:00
[kcompactd0]
root      44  0.0  0.0      0  0 ?      SN  11:43  0:00 [ksmd]
root      45  0.0  0.0      0  0 ?      SN  11:43  0:00
[khugepaged]
root      46  0.0  0.0      0  0 ?      S<  11:43  0:00 [crypto]
root      47  0.0  0.0      0  0 ?      S<  11:43  0:00
[kintegrityd]
root      48  0.0  0.0      0  0 ?      S<  11:43  0:00 [bioset]
root      49  0.0  0.0      0  0 ?      S<  11:43  0:00 [kblockd]
root      52  0.0  0.0      0  0 ?      S<  11:43  0:00 [ata_sff]
root      53  0.0  0.0      0  0 ?      S<  11:43  0:00 [md]
root      54  0.0  0.0      0  0 ?      S<  11:43  0:00
[devfreq_wq]
root      55  0.0  0.0      0  0 ?      S<  11:43  0:00
[watchdogd]
root      57  0.0  0.0      0  0 ?      S   11:43  0:00 [kauditd]
root      58  0.0  0.0      0  0 ?      S   11:43  0:02 [kswapd0]
root      59  0.0  0.0      0  0 ?      S<  11:43  0:00 [vmstat]
root      60  0.0  0.0      0  0 ?      S<  11:43  0:00 [bioset]
root      93  0.0  0.0      0  0 ?      S   11:43  0:00
[kworker/2:1]
root     109  0.0  0.0      0  0 ?      S<  11:43  0:00 [kthrotld]
root     110  0.0  0.0      0  0 ?      S<  11:43  0:00
[acpi_thermal_p
root     111  0.0  0.0      0  0 ?      S   11:43  0:00
[scsi_eh_0]
root     112  0.0  0.0      0  0 ?      S<  11:43  0:00
[scsi_tmf_0]
root     113  0.0  0.0      0  0 ?      S   11:43  0:00
[scsi_eh_1]
root     114  0.0  0.0      0  0 ?      S<  11:43  0:00
[scsi_tmf_1]
root     115  0.0  0.0      0  0 ?      R   11:43  0:01
[kworker/u128:2
root     117  0.0  0.0      0  0 ?      S<  11:43  0:00
[dm_bufio_cache
root     118  0.0  0.0      0  0 ?      S<  11:43  0:00
[ipv6_addrconf]
root     162  0.0  0.0      0  0 ?      S<  11:43  0:00 [bioset]
root     384  0.0  0.0      0  0 ?      S<  11:43  0:00
[mpt_poll_0]
root     385  0.0  0.0      0  0 ?      S<  11:43  0:00 [mpt/0]
root     389  0.0  0.0      0  0 ?      S<  11:43  0:00
[kworker/0:1H]

```

root	392	0.0	0.0	0	0 ?	S	11:43	0:00	
[scsi_ah_2]									
root	393	0.0	0.0	0	0 ?	S<	11:43	0:00	
[scsi_tm_2]									
root	394	0.0	0.0	0	0 ?	S<	11:43	0:00	[bioset]
root	409	0.0	0.0	0	0 ?	S<	11:43	0:00	[ttm_swap]
root	416	0.0	0.0	0	0 ?	S<	11:43	0:00	
[kworker/3:1H]									
root	417	0.0	0.0	0	0 ?	S<	11:43	0:00	
[kworker/2:1H]									
root	472	0.0	0.0	0	0 ?	S<	11:43	0:00	[kdmflush]
root	475	0.0	0.0	0	0 ?	S<	11:43	0:00	[bioset]
root	484	0.0	0.0	0	0 ?	S<	11:43	0:00	[kdmflush]
root	487	0.0	0.0	0	0 ?	S<	11:43	0:00	[bioset]
root	506	0.0	0.0	0	0 ?	S	11:43	0:00	[jbd2/dm-0-8]
root	507	0.0	0.0	0	0 ?	S<	11:43	0:00	[ext4-rsv-conve]
root	591	0.0	0.0	0	0 ?	S<	11:43	0:00	
[kworker/1:1H]									
root	596	0.0	0.6	101700	19420 ?	Ss	11:43	0:02	
/usr/lib/system									
root	616	0.0	0.0	0	0 ?	S<	11:43	0:00	[rpciod]
root	617	0.0	0.0	0	0 ?	S<	11:43	0:00	[xpriod]
root	628	0.0	0.1	49412	5988 ?	Ss	11:43	0:01	
/usr/lib/system									
root	657	0.0	0.0	0	0 ?	S<	11:43	0:00	[nfit]
root	740	0.0	0.0	0	0 ?	S	11:43	0:00	
[jbd2/sda1-8]									
root	741	0.0	0.0	0	0 ?	S<	11:43	0:00	[ext4-rsv-conve]
root	769	0.0	0.0	55572	1648 ?	S<sl	11:43	0:00	
/sbin/auditd									
root	771	0.0	0.0	84544	1680 ?	S<sl	11:43	0:00	
/sbin/audispd									
root	774	0.0	0.0	43656	764 ?	S<	11:43	0:00	
/usr/sbin/sedis									
root	792	0.0	0.0	16880	820 ?	SNs	11:43	0:00	
/usr/sbin/alsac									
root	793	0.0	0.1	395452	3580 ?	Ssl	11:43	0:00	
/usr/libexec/ac									
root	795	0.0	0.0	4416	632 ?	Ss	11:43	0:00	/sbin/rngd-f
avahi	796	0.0	0.0	50320	2200 ?	Ss	11:43	0:00	avahi-daemon: r
root	797	0.1	0.1	317988	4356 ?	Ssl	11:43	0:06	
/usr/bin/vmtool									
root	799	0.0	0.1	50240	3696 ?	Ss	11:43	0:00	
/usr/lib/system									
rtkit	802	0.0	0.0	186848	2232 ?	SNsl	11:43	0:00	
/usr/libexec/rt									
dbus	803	0.0	0.1	57928	3644 ?	Ssl	11:43	0:01	
/usr/bin/dbus-d									
avahi	804	0.0	0.0	50196	348 ?	S	11:43	0:00	avahi-

```

daemon: c
root      809  0.0  0.0 204616  2016 ?          Ssl  11:43  0:00
/usr/sbin/gsspr
root      821  0.0  0.8 350856 25948 ?          Ssl  11:43  0:03
/usr/bin/python
root      822  0.0  0.1 417744  4192 ?          Ssl  11:43  0:00
/usr/sbin/Modem
root      823  0.0  0.0   6568  1700 ?          Ss   11:43  0:00
/usr/sbin/mcelo
root      836  0.0  0.1 447208  4280 ?          Ssl  11:43  0:00
/usr/sbin/abrt-d
chrony     844  0.0  0.0 112064  2388 ?          S    11:43  0:00
/usr/sbin/chron
polkitd    850  0.0  0.3 533672 11208 ?          Ssl  11:43  0:01
/usr/lib/polkit
root      875  0.0  0.1 541512  4348 ?          Ss   11:43  0:00
/usr/bin/abrt-d
root      877  0.0  0.1 573468  4640 ?          Ss   11:43  0:00
/usr/bin/abrt-d
root      889  0.0  0.3 551704 11952 ?          Ssl  11:43  0:00
/usr/sbin/Netwo
root      904  0.0  0.0 100452  2244 ?          Ss   11:43  0:00
/usr/sbin/sshd
root      912  0.0  0.3 912368  9552 ?          Ssl  11:43  0:01
/usr/sbin/libvi
root      923  0.0  0.1 409848  3796 ?          Ssl  11:43  0:00
/usr/sbin/gdm
root      924  0.0  0.0   25960  1628 ?          Ss   11:43  0:00
/usr/sbin/atd -
root      926  0.0  0.0 132936  2324 ?          Ss   11:43  0:00
/usr/sbin/crond
root      968  0.0  0.1 372048  4172 ?          Sl   11:43  0:00 gdm-
session-wor
gdm       1008  0.0  0.1   66088  3988 ?          Ss   11:43  0:00
/usr/lib/system
gdm       1016  0.0  0.0   99044  3020 ?          S    11:43  0:00 (sd-pam)
gdm       1023  0.0  0.1 427644  4616 tty1       Ssl+  11:43  0:00
/usr/libexec/gd
gdm       1039  0.0  0.0   56532  2476 ?          Ssl  11:43  0:00
/usr/bin/dbus-d
gdm       1048  0.0  0.1 669136  5436 tty1       Sl+   11:43  0:00
/usr/libexec/gn
gdm       1057  0.2  2.2 1947572 70176 tty1       Sl+   11:43  0:12
/usr/bin/gnome-
root      1083  0.0  0.1 431844  3784 ?          Ssl  11:43  0:00
/usr/libexec/up
nobody    1149  0.0  0.0   49104  1616 ?          S    11:43  0:00
/sbin/dnsmasq -
root      1151  0.0  0.0   49076   356 ?          S    11:43  0:00
/sbin/dnsmasq -
gdm       1171  0.6  0.5 255872 16904 tty1       Sl+   11:43  0:37
/usr/bin/Xwayla
gdm       1241  0.0  0.0 344700   2976 ?          Ssl  11:43  0:00
/usr/libexec/at

```


gdm	1246	0.0	0.0	56304	1716	?	Sl	11:43	0:00	/bin/dbus-daemo
gdm	1249	0.0	0.0	223372	2468	?	Sl	11:43	0:00	
/usr/libexec/at										
gdm	1261	0.0	0.1	1247336	4768	?	Sl	11:43	0:01	
/usr/bin/pulsea										
gdm	1288	0.0	0.1	460164	5040	tty1	Sl	11:43	0:00	ibus-
daemon --x										
gdm	1291	0.0	0.0	382568	2516	tty1	Sl	11:43	0:00	
/usr/libexec/ib										
gdm	1293	0.0	0.4	576408	13520	tty1	Sl	11:43	0:00	
/usr/libexec/ib										
gdm	1300	0.0	0.1	441548	3244	?	Ssl	11:43	0:00	
/usr/libexec/xd										
root	1311	0.0	2.2	636028	69912	?	Ssl	11:43	0:05	
/usr/libexec/pa										
root	1312	0.0	0.0	66264	2428	?	Ss	11:43	0:00	
/usr/sbin/wpa_s										
gdm	1317	0.0	0.5	1256168	16208	tty1	Sl+	11:43	0:01	
/usr/libexec/gn										
gdm	1356	0.0	0.0	308768	2548	tty1	Sl	11:43	0:00	
/usr/libexec/ib										
colord	1386	0.0	0.1	415760	5128	?	Ssl	11:43	0:00	
/usr/libexec/co										
root	1408	0.0	0.1	394844	4640	?	Sl	11:43	0:00	gdm-
session-wor										
student	1415	0.0	0.1	66048	4636	?	Ss	11:43	0:00	
/usr/lib/system										
student	1421	0.0	0.1	99044	3120	?	S	11:43	0:00	(sd-pam)
student	1431	0.0	0.1	471164	3396	?	Sl	11:43	0:00	
/usr/bin/gnome-										
student	1434	0.0	0.1	427656	4032	tty2	Ssl+	11:43	0:00	
/usr/libexec/gd										
student	1439	2.2	1.7	399204	52740	tty2	Sl+	11:43	2:05	
/usr/libexec/Xo										
student	1455	0.0	0.1	57272	4188	?	Ssl	11:44	0:02	
/usr/bin/dbus-d										
student	1458	0.0	0.1	669268	5696	tty2	Sl+	11:44	0:00	
/usr/libexec/gn										
student	1501	0.0	0.0	344704	3028	?	Ssl	11:44	0:00	
/usr/libexec/at										
student	1506	0.0	0.0	56432	2344	?	Sl	11:44	0:00	/bin/dbus-daemo
student	1509	0.0	0.1	223264	3468	?	Sl	11:44	0:01	
/usr/libexec/at										
student	1521	0.0	0.1	395036	3188	?	Ssl	11:44	0:00	
/usr/libexec/gv										
student	1526	0.0	0.0	417804	2496	?	Sl	11:44	0:00	
/usr/libexec/gv										
student	1542	3.3	4.4	2142056	135304	tty2	Sl+	11:44	3:07	
/usr/bin/gnome-										
student	1550	0.0	0.1	1542676	5064	?	Sl	11:44	0:01	
/usr/bin/pulsea										
student	1563	0.0	0.3	868544	9460	?	Sl	11:44	0:00	

```

/usr/libexec/gn
student 1564 0.0 0.1 460196 5108 tty2 Sl 11:44 0:00 ibus-
daemon --x
student 1569 0.0 0.0 382564 2808 tty2 Sl 11:44 0:00
/usr/libexec/ib
student 1571 0.0 0.2 463104 8872 tty2 Sl 11:44 0:00
/usr/libexec/ib
student 1582 0.0 0.5 1510144 15380 ? Ssl 11:44 0:00
/usr/libexec/ev
student 1583 0.0 0.1 441548 3076 ? Ssl 11:44 0:00
/usr/libexec/xd
student 1593 0.0 0.1 422356 3320 ? Ssl 11:44 0:00
/usr/libexec/gv
root 1596 0.0 0.1 388016 4188 ? Ssl 11:44 0:00
/usr/libexec/ud
student 1607 0.0 0.0 385420 2080 ? Ssl 11:44 0:00
/usr/libexec/gv
student 1610 0.0 0.2 874072 8188 ? Sl 11:44 0:00
/usr/libexec/go
student 1614 0.0 0.1 394520 3396 ? Ssl 11:44 0:00
/usr/libexec/gv
student 1620 0.0 0.1 492792 3560 ? Ssl 11:44 0:00
/usr/libexec/gv
student 1625 0.0 0.0 378212 3036 ? Ssl 11:44 0:00
/usr/libexec/gv
student 1635 0.0 0.1 598992 5100 ? Sl 11:44 0:00
/usr/libexec/go
student 1646 0.0 0.5 1235196 15792 tty2 Sl+ 11:44 0:01
/usr/libexec/gn
student 1664 0.0 0.4 1054420 12952 ? Ssl 11:44 0:00
/usr/libexec/ev
student 1676 0.1 0.3 381960 9336 tty2 S+ 11:44 0:06
/usr/bin/vmtool
student 1678 0.0 0.1 565136 5644 tty2 SNl+ 11:44 0:00
/usr/libexec/tr
student 1679 0.0 0.2 655032 7232 tty2 SNl+ 11:44 0:00
/usr/libexec/tr
student 1680 0.0 0.2 635936 6356 tty2 SNl+ 11:44 0:00
/usr/libexec/tr
student 1683 0.0 0.2 456304 6488 tty2 Sl+ 11:44 0:00
/usr/bin/seappl
root 1685 0.0 0.1 212688 3264 ? Ss 11:44 0:00
/usr/sbin/cupsd
student 1686 0.0 0.2 744864 6656 tty2 SNl+ 11:44 0:00
/usr/libexec/tr
student 1696 0.0 0.1 622952 4392 tty2 Sl+ 11:44 0:00
/usr/libexec/gs
student 1708 0.0 0.3 604460 10296 ? Ssl 11:44 0:00
/usr/libexec/tr
student 1718 0.0 0.4 1146008 14036 ? Sl 11:44 0:00
/usr/libexec/ev
student 1749 0.0 0.0 308896 2600 tty2 Sl 11:44 0:00
/usr/libexec/ib
student 1767 0.0 0.0 187400 2952 ? Sl 11:44 0:00

```

```

/usr/libexec/dc
student 1771 0.0 0.3 1040928 10836 ? Ssl 11:44 0:00
/usr/libexec/ev
student 1774 0.0 0.4 1073652 14344 ? Sl 11:44 0:00
/usr/libexec/ev
root 1787 0.0 0.1 343668 4260 ? Sl 11:44 0:00
/usr/sbin/abrt-
student 1811 0.0 0.4 1289500 13968 ? Sl 11:44 0:00
/usr/libexec/ev
student 1940 0.0 0.0 471056 2972 ? Sl 11:44 0:00
/usr/libexec/gv
student 1987 0.4 1.2 827360 39184 tty2 Sl+ 11:44 0:23
/usr/bin/python
student 2041 0.0 0.1 123436 4384 pts/0 Ss 11:44 0:00 /bin/bash
student 2117 1.6 7.4 1410396 227604 tty2 Sl+ 11:44 1:34
/usr/lib64/fire
student 2176 0.0 0.0 180584 3024 ? S 11:45 0:00
/usr/libexec/gc
student 2242 0.7 3.4 5027224 107072 ? SLsl 11:45 0:39
/usr/share/code
student 2250 0.0 0.3 453540 10616 ? S 11:45 0:00
/usr/share/code
student 2251 0.0 0.3 453540 10748 ? S 11:45 0:00
/usr/share/code
student 2275 2.1 2.3 654736 71620 ? Sl 11:45 2:00
/usr/share/code
student 2289 0.0 1.1 541812 34776 ? Sl 11:45 0:02
/usr/share/code
student 2299 5.4 6.7 19683388 206604 ? Sl 11:45 5:00
/usr/share/code
student 2319 0.1 2.1 4858188 67540 ? Sl 11:45 0:10
/usr/share/code
student 2339 0.5 3.2 4703364 100556 ? Sl 11:45 0:30
/usr/share/code
student 2355 0.0 1.2 4630060 38020 ? Sl 11:45 0:02
/usr/share/code
student 2663 0.1 0.8 1090468 25216 ? Sl 12:00 0:04
/usr/bin/nautil
root 2796 0.0 0.0 0 0 ? S 12:00 0:00
[kworker/1:0]
student 2852 0.0 0.4 598792 15332 tty2 Sl+ 12:00 0:00 abrt-
applet
student 3309 0.0 0.1 309608 3104 ? Ssl 12:02 0:00
/usr/libexec/gv
root 49755 0.0 0.6 87132 18548 ? S 12:34 0:00
/sbin/dhclient
root 49935 0.0 0.0 0 0 ? S 12:44 0:00
[kworker/2:2]
root 49936 0.0 0.0 0 0 ? S 12:44 0:00
[kworker/1:1]
root 49988 0.0 0.0 0 0 ? S 12:46 0:00
[kworker/0:2]
root 50776 0.0 0.0 0 0 ? S 12:53 0:00
[kworker/u128:0

```

```

root      60938  0.0  0.0      0      0 ?      S    13:02  0:00
[kworker/0:0]
student   60942  1.1  1.0  45552 33640 pts/0    S+   13:03  0:09 ./vmlinux
ubd0=
student   60947  0.0  1.0  45552 33640 pts/0    S+   13:03  0:00 ./vmlinux
ubd0=
student   60948  0.0  1.0  45552 33640 pts/0    S+   13:03  0:00 ./vmlinux
ubd0=
student   60949  0.0  1.0  45552 33640 pts/0    S+   13:03  0:00 ./vmlinux
ubd0=
student   60950  0.0  0.0  14908  2264 pts/0    t+   13:03  0:00 ./vmlinux
ubd0=
student   61080  0.0  0.0  14140   696 pts/0    t+   13:03  0:00 ./vmlinux
ubd0=
student   61224  0.0  0.0  14084   788 pts/0    t+   13:03  0:00 ./vmlinux
ubd0=
student   61523  0.0  0.0  13860   740 pts/0    t+   13:03  0:00 ./vmlinux
ubd0=
student   61549  0.0  0.0  15144  1320 pts/0    t+   13:03  0:00 ./vmlinux
ubd0=
student   61556  0.0  0.0  15284  1872 pts/0    t+   13:03  0:00 ./vmlinux
ubd0=
root      61575  0.0  0.0      0      0 ?      S    13:05  0:00
[kworker/3:0]
root      61672  0.0  0.0      0      0 ?      S    13:10  0:00
[kworker/3:1]
root      61751  0.0  0.0      0      0 ?      S    13:15  0:00
[kworker/1:2]
root      61784  0.0  0.0      0      0 ?      S    13:16  0:00
[kworker/3:2]
student   61807  1.5  0.7 465392 21672 ?      Sl   13:17  0:00
/usr/libexec/gn
student   61809  3.7  1.3 1108360 40628 ?      Sl   13:17  0:00
/usr/libexec/gn
student   61812  4.0  1.5 1518836 46864 ?      Sl   13:17  0:00
/usr/bin/gnome-
student   61814  5.2  1.1 582040 34292 ?      Sl   13:17  0:00
/usr/bin/gnome-
student   61815  0.0  0.1 297664  5640 ?      Sl   13:17  0:00
/usr/libexec/gn
student   61817  4.7  1.2 1116500 37668 ?      SLl  13:17  0:00
/usr/bin/seahor
student   61819 15.5  2.1 1048112 66856 ?      Sl   13:17  0:00
/usr/bin/gnome-
student   61820  4.5  1.0 725644 33684 ?      Ssl  13:17  0:00
/usr/libexec/gn
student   61912  0.6  0.1 123304  4828 pts/1    Ss   13:17  0:00 bash
student   61968  0.0  0.1 150020  3764 pts/1    R+   13:17  0:00 ps aux

```

Po odfiltrowaniu rekordów ze słowem **vmlinux**:

```
student 60942 1.0 1.0 45552 33640 pts/0 S+ 13:03 0:09 ./vmlinux
ubd0=/home/student/fs/CentOS6.x-AMD64-root_fs
student 60947 0.0 1.0 45552 33640 pts/0 S+ 13:03 0:00 ./vmlinux
ubd0=/home/student/fs/CentOS6.x-AMD64-root_fs
student 60948 0.0 1.0 45552 33640 pts/0 S+ 13:03 0:00 ./vmlinux
ubd0=/home/student/fs/CentOS6.x-AMD64-root_fs
student 60949 0.0 1.0 45552 33640 pts/0 S+ 13:03 0:00 ./vmlinux
ubd0=/home/student/fs/CentOS6.x-AMD64-root_fs
student 60950 0.0 0.0 14908 2264 pts/0 t+ 13:03 0:00 ./vmlinux
ubd0=/home/student/fs/CentOS6.x-AMD64-root_fs
student 61080 0.0 0.0 14140 696 pts/0 t+ 13:03 0:00 ./vmlinux
ubd0=/home/student/fs/CentOS6.x-AMD64-root_fs
student 61224 0.0 0.0 14084 788 pts/0 t+ 13:03 0:00 ./vmlinux
ubd0=/home/student/fs/CentOS6.x-AMD64-root_fs
student 61523 0.0 0.0 13860 740 pts/0 t+ 13:03 0:00 ./vmlinux
ubd0=/home/student/fs/CentOS6.x-AMD64-root_fs
student 61549 0.0 0.0 15144 1320 pts/0 t+ 13:03 0:00 ./vmlinux
ubd0=/home/student/fs/CentOS6.x-AMD64-root_fs
student 61556 0.0 0.0 15284 1892 pts/0 t+ 13:03 0:00 ./vmlinux
ubd0=/home/student/fs/CentOS6.x-AMD64-root_fs
student 62038 0.0 0.0 119372 912 pts/1 S+ 13:19 0:00 grep --
color=auto vmlinu
```

Zatem część procesów z gościa jest przenoszona zapewne na hosta.

5.7 Zakończ pracę z gościem.

Analogicznie, jak w przypadku QEMU, aby wyjść ze środowiska należało wywołać komendę `poweroff`, ponieważ `exit` powoduje tylko wylogowanie się.