

## „Zapis na partycji UFS2 na Linuksie”

Za względu na to, iż obsługa systemu plików UFS2 (czyli systemu plików jaki znajduje się na dev\_hdd0) jest mocno problematyczna na Linuksie, postanowiłem wydzielić tę część materiału do osobnego poradnika.

Tekst podzieliłem na kilka części. Pierwsza poświęcona jest wersji HDD Reader dla Linuksa, który daje najmniejsze pole manewru, ale jest bezpieczny w użyciu. Druga część opisuje specjalną wersję fuse-ufs, która ze względu na szereg pułapek czekających na nieświadomą ofiarę jest bardzo niebezpieczna dla tablicy. I trzecia część, z której dowiesz się jak dodać natywną, choć eksperymentalną obsługę zapisu dla modułu jądra.

Znajomość bliźniaczego poradnika traktującego o konwersji w locie Big Endian na Little Endian za pomocą bswap16, generowaniu kluczy ATA i VFLASH z EID Root Key, deszyfracji i montowaniu systemów plików poszczególnych partycji, dla dwóch ostatnich przypadków jest **obligatoryjna**.

# HDD Reader

HDD Reader to port wersji na Windows, programu który sam zajmuje się operacjami na dysku PS3. W stosunku do swojego protoplasty, został dodatkowo wzbogacony o funkcję zastępowania plików i obsługę dysków twardych z modeli Arcade.

## Kompilacja ze źródeł

Nie musisz tego robić, możesz pobrać gotową „binarkę”. Gdybyś jednak mi nie ufał lub z jakiegoś powodu nie uruchamiała się na twojej ulubionej dystrybucji Linuksa, zawsze możesz samemu skompilować ze źródeł.

1. Po rozpakowaniu archiwum, upewnij się że masz w systemie zainstalowane [build-essential](#). Jeśli nie, to oczywiście doinstaluj.
2. Otwórz terminal, przejdź do katalogu ze źródłem i wpisz "make".

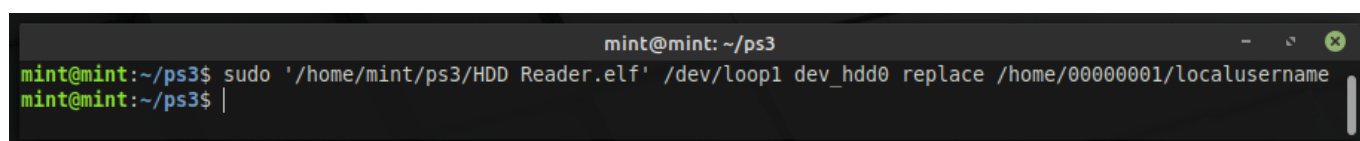
## Podmiany

Nie można dodawać własnych danych, można tylko i wyłącznie zastępować już istniejące. Co więcej, **rozmiar musi być równy** plikowi, który zastępujesz. Dzieje się tak dlatego, że HDD Reader nie zapisuje nic w tablicy UFS2, a jedynie podmienia okupowane przez wybrany plik sektory.

1. Szczytaj klucz EID Root Key z konsoli i wrzuc go w to samo miejsce gdzie znajduje się program pod nazwą "eid\_root\_key" (czyli bez rozszerzenia).
2. Jeśli jeszcze tego nie zrobiłeś, dodaj atrybut wykonywalny plikowi "HDD Reader.elf". Możesz to zrobić w managerze plików jeśli posiada taką funkcjonalność (np. w Nemo znajdziesz we właściwościach pliku jako checkbox) lub z poziomu terminalu za pomocą "chmod +x".
3. Aplikacja pracuje na urządzeniu, więc w składni wybierasz np. "/dev/sdx" lub "/dev/loopx", którym ma być podłączony do komputera dysk twardy z PS3 lub jego obraz. Odpada więc cała gehenna z tworzeniem specjalnych mapperów i ręcznego montowania systemu plików. Drugim parametrem jest partycja, koniecznie zgodna z nomenklaturą Cellos. Następnie polecenie (cp, ls, replace) i ścieżka na teże partycji do pliku lub folderu.

Kopiuwane pliki trafiają do katalogu z programem – i stamtąd też są wczytywane. Dlatego jeśli chcesz np. zastąpić wirtualną kartę pamięci gry o identyfikatorze SLES12345, to musisz ją najpierw wrzucić w to samo miejsce co "HDD Reader.elf", o identycznej nazwie czyli w tym przykładzie "SCEVMC0.VME", zaś w terminalu wpisać:

```
"sudo '/home/mint/ps3/HDD Reader.elf' /dev/sdx dev_hdd0 replace /home/00000001/ps2emu2_savedata/SLES12345/SCEVMC0.VME"
```



```
mint@mint: ~/ps3
mint@mint:~/ps3$ sudo '/home/mint/ps3/HDD Reader.elf' /dev/loop1 dev_hdd0 replace /home/00000001/localusername
mint@mint:~/ps3$ |
```

Powyżej inny przykład, tym razem z plikiem przechowującym nazwę użytkownika.

## Fuse-UFS2+BE

Jest to fork fuse-ufs, który usprawnia nieco oryginalny kod. Niestety projekt jest wciąż w powijakach (testowana wersja to build-004), który może łatwo doprowadzić do uszkodzeń tablicy systemu plików. Nie należy przeprowadzać operacji na zbyt wielu plikach (np. kopiowanie kilkudziesięciu zestawów trofeów; może to całkowicie zawiesić środowisko graficzne), a nawet nie jest bezpiecznie pracować na zbyt dużym zagnieżdżeniu. Program nadaje się do kosmetycznych poprawek i na wszelki wypadek nigdy bez kopii zapasowej. Pomimo tego złowieszczonego wstępu, to nadal użyteczny kawałek kodu, ponieważ daje znacznie szersze pole manewru niż HDD Reader i nie wymaga pałowania się z kompilacją modułu kernela, która nie jest taka banalna na LiveCD/USB.

### Kompilacja ze źródeł

Nie musisz tego robić, możesz pobrać gotową „binarkę”. Gdybyś jednak mi nie ufał lub z jakiegoś powodu nie uruchamiała się na twojej ulubionej dystrybucji Linuksa, zawsze możesz samemu skompilować ze źródeł.

1. Po rozpakowaniu archiwum, upewnij się że masz w systemie [g++](#), [libfuse-dev](#) i [e2fslibs-dev](#). Jeśli nie, to oczywiście doinstaluj.
2. Otwórz terminal, przejdź do katalogu ze źródłem i wpisz `./configure`.
3. Przeglądnij log i jeśli niczego nie brakuje to wpisz `make`. Po udanej kompilacji pojawią się pliki `./fuse-ufs/fuse-ufs` (który przenieśliśmy do katalogu `ps3` i któremu zmieniłem nazwę na `fuse-ufs2+be.elf` bo tak nazywa się ten projekt i dlatego, że mam cyfrowy fetysz do nazw z rozszerzeniami, mimo że na Linuksie są zupełnie zbędne) plus dodatkowo `./fuse-ufs/fuse-ufs.probe` (jest niepotrzebny).

### Montaż

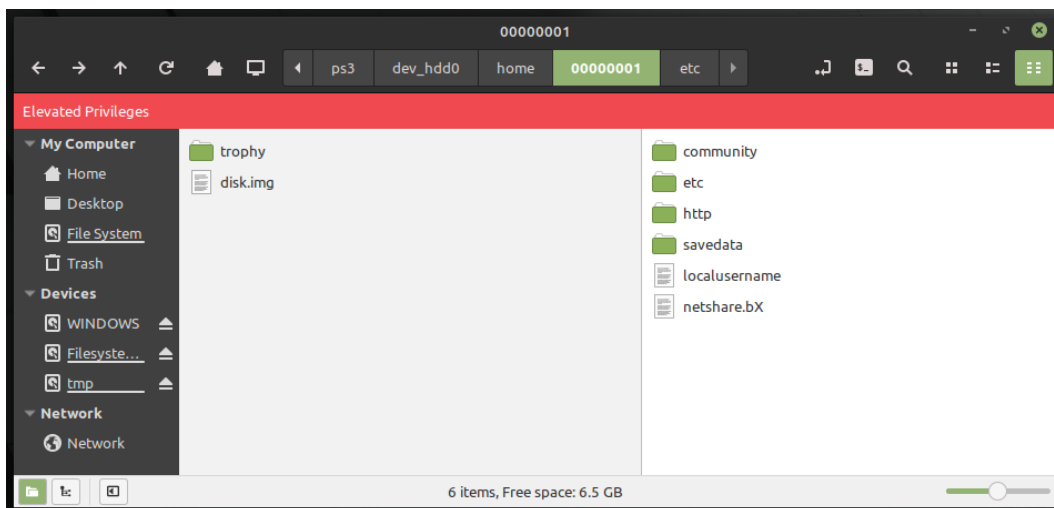
1. Jeśli jeszcze tego nie zrobiłeś, dodaj atrybut wykonywalny plikowi `fuse-ufs2+be.elf`. Możesz to zrobić w managerze plików jeśli posiada taką funkcjonalność (np. w Nemo znajdziesz we właściwościach pliku jako checkbox) lub z poziomu terminalu za pomocą `chmod +x`.
2. Wystaw partycję użytkownika (czyli wg nomenklatury CellOS `dev_hdd0`) jako mapper. Pamiętaj jednak aby go nie montować bo tym zajmie się fuse-ufs2+be.
3. Na koniec wklep: `"fuse-ufs2+be.elf /dev/mapper/ps3hdd2 /home/mint/ps3/dev_hdd0 -o rw+"`. Oczywiście mapper musi odpowiadać zmapowanej partycji z UFS2. Jeśli postępowałeś zgodnie z poradnikiem o montowaniu i posiadasz konsolę z pamięcią NOR to zawsze będzie to `ps3hdd2`. Miejsce gdzie zamontujesz fs nie ma znaczenia, u mnie to po prostu folder o nazwie `dev_hdd0` w katalogu `ps3`.

```
root@mint: /home/mint/ps3
mint@mint:~/ps3$ sudo su
root@mint:/home/mint/ps3# losetup loop1 /media/mint/WINDOWS/ps3res/disk.img
root@mint:/home/mint/ps3# insmod '/home/mint/ps3/bswap16-ecb.ko'
root@mint:/home/mint/ps3# cryptsetup create -c bswap16-ecb -d /dev/zero ps3hdd-bs /dev/loop1
root@mint:/home/mint/ps3# cryptsetup create -c aes-cbc-null -d /home/mint/ps3/ata_key.bin -s 192 ps3hdd /dev/mapper/ps3hdd-bs
root@mint:/home/mint/ps3# kpartx -a /dev/mapper/ps3hdd
root@mint:/home/mint/ps3# "/home/mint/ps3/fuse-ufs2+be.elf" /dev/mapper/ps3hdd2 /home/mint/ps3/dev_hdd0 -o rw+
version:'0.0.1', fuse version:'29'
libufs in ufs_disk_fillout_blank
libufs in sbread
libufs in bread
libufs in bread
libufs in ufs_disk_close
Mounting /dev/mapper/ps3hdd2 in Read-Write mode
root@mint:/home/mint/ps3# lsblk -b /dev/loop1
NAME        MAJ:MIN RM   SIZE RO TYPE  MOUNTPOINT
loop1         7:1      0 3921674240 0 loop
└─ps3hdd-bs 253:0      0 3921674240 0 crypt
   └─ps3hdd 253:1      0 3921674240 0 crypt
      └─ps3hdd1 253:2      0 268435456 0 part
         └─ps3hdd2 253:3      0 1505746944 0 part  /home/mint/ps3/dev_hdd0
            └─ps3hdd3 253:4      0 2147479552 0 part
root@mint:/home/mint/ps3#
```

Jeśli masz zamiar dograć jakieś dane, problemem mogą być nie tylko wycieki pamięci, ;) ale i problem z określeniem ilości wolnego miejsca. Jeśli manager plików tego nie pokazuje, można posłużyć się narzędziem `df` (dla urządzenia) i `du` (dla katalogów).

```
root@mint: /home/mint/ps3
root@mint:/home/mint/ps3# df /dev/mapper/ps3hdd2
Filesystem            1K-blocks    Used Available Use% Mounted on
/dev/mapper/ps3hdd2    1423014  514180    794994   40% /home/mint/ps3/dev_hdd0
root@mint:/home/mint/ps3# |
```

Wszelkie operacje plikowe można przeprowadzać zarówno z poziomu terminalu jak i menedżera plików. Jak ci wygodniej.



Aby przeglądać zawartość, wymagane są uprawnienia roota.

4. Na koniec możesz już odmontować system plików wpisując "`umount -l /home/mint/ps3/dev_hdd0`". Naturalnie, potem pozostałe mappery również pozdejmuj.

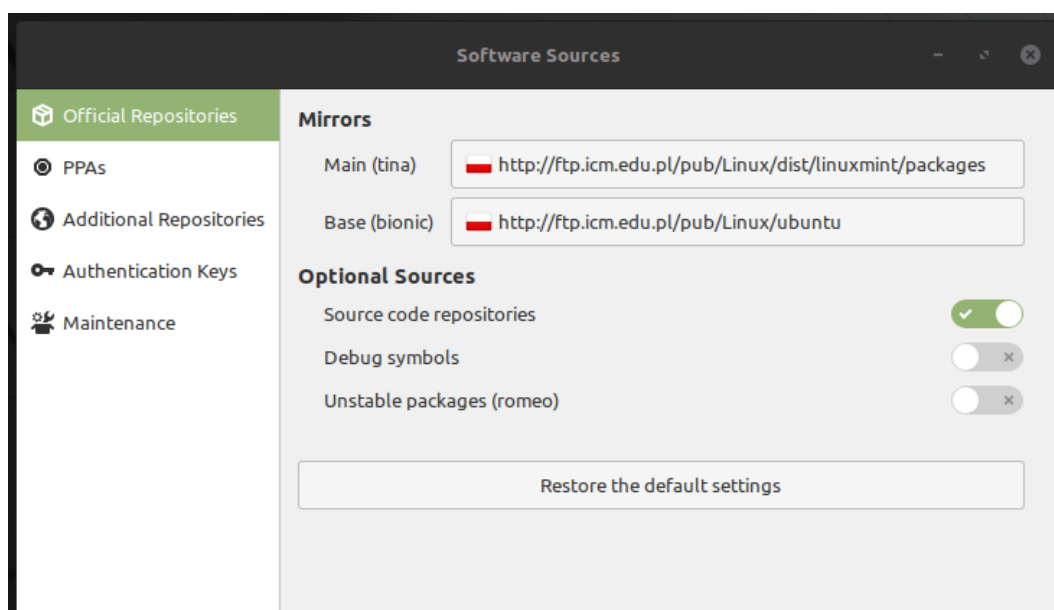
# UFS Kernel Module

## Kompilacja modułu jądra z flagą rw dla UFS2

Być może brzmi to dla ciebie jak język kosmitów, ale chodzi o to iż domyślnie kernel (jądro) Linux nie pozwala nic zapisywać na partycjach UFS2 (a taki właśnie system plików używa „główna partycja” HDD w PS3). Zapis jest w dalszym ciągu eksperymentalny, toteż domyślnie go wyłączono. Oznacza to oczywiście ryzyko uszkodzenia systemu plików, a więc utraty danych gdyby coś poszło nie tak. Czuć się ostrzeżony.

Haczyk polega na tym, że ciężko jest samemu skompilować (de facto co najmniej połowę) i to w konkretnej wersji ze źródeł w konkretnej wersji, a do tego trwa to bardzo długo (na moim trupie około 4 godziny...). Być może brakuje mi wiedzy, a być może coś jest nie tak z apt-get i/lub dystrybucją Linux Mint bo na wersji live kompilacja jest niemożliwa, a na zainstalowanej nie pobiera się ta sama wersja co używanego jądra – mało tego – moduł po kompilacji jest w jeszcze innej wersji, więc nie pasuje do żadnej z nich (ani używanego kernela, ani do tego ze źródeł). Nie rozumiem dlaczego tak się dzieje i szczerze powiedziawszy poddałem się w trosce o swój czas i zdrowie psychiczne... Mimo wszystko niżej znajdziesz przepis – mnie niestety wychodzi zakalec. ;)

1. Przejdź do źródeł oprogramowania w menu i włącz "Source code repositories", zgadzając się na proponowaną aktualizację pamięci podręcznej (ekwiwalent "[apt update](#)").



2. Teraz po kolei wyklep w terminalu:

```
"sudo apt build-dep linux linux-image-$(uname -r)"
```

```
"sudo apt install libncurses5-dev flex bison libssl-dev dkms libelf-dev libudev-dev libpci-dev libiberty-dev"
```

```
"sudo apt source linux-image-$(uname -r)"
```

```
"sudo apt install linux-headers-$(uname -r)"
```

3. Skopiuj plik `"/boot/config-<wersja>"` do `"/usr/src/<wersja kernela>/"` i wklej jako `".config"` (w ten sposób skłonujesz ustawienia kernela i ominie cię męczarnia związana z `"make menuconfig"`).

4. Otwórz ten plik edytorem tekstowym i zastąp całą linijkę z `"# CONFIG_UFS_FS_WRITE is not set"` frazą `"CONFIG_UFS_FS_WRITE=y"`.

5. Przejdź w terminalu do katalogu ze źródłem i wpisz `"sudo make modules"`, mając nadzieję że po wielu godzinach wypłuty zostanie upragniony plik `"/fs/ufs/ufs.ko"`.

6. Po pomyślnej kompilacji skopiuj go sobie w bezpieczne miejsce, np. do `"/home/mint/ps3/"` i sprawdź czy moduł pasuje do używanego kernela za pomocą `"modinfo"`.

7. Jeśli tak to go załaduj wpisując `"insmod '/home/mint/ps3/ufs.ko'"`.

8. Od teraz obsługa UFS2 powinna umożliwiać także zapis na partycjach z tym systemem plików, a więc należy to uwzględnić podczas montowania z mappera:

```
"mount -t ufs -o ufstype=ufs2,rw /dev/mapper/ps3hdd2 /home/mint/ps3/dev_hdd0"
```

Jak wspomniałem wcześniej, mnie się ta sztuka nie udała ponieważ wypłuty moduł był dla innego kernela – i to nie tylko tego, którego używałem, ale nawet tego którego źródła się pobrały (mimo że powinny się pobrać zgodne z tym co zwraca zmienna "uname -r"). W powyższej procedurze prawdopodobnie tkwi gdzieś błąd, ale nie potrafię tego wyśledzić i szczerze powiedziawszy z racji czasochłonnych testów, straciłem wolę jego znalezienia. Tym bardziej, że ta metoda również nie gwarantuje stabilnej obsługi UFS2...