Disque /dev/sdb : 1,9 GiB, 2062548992 octets, 4028416 secteurs

Unités : sectors of 1 \* 512 = 512 octets

Sector size (logical/physical): 512 bytes / 512 bytes

I/O size (minimum/optimal): 512 bytes / 512 bytes

Disklabel type: gpt

Disk identifier: 6DA651CD-ED6E-4D59-99A2-32009A773E8F

Périphérique Start Fin Secteurs Size Type

/dev/sdb1 2048 411647 409600 200M Linux filesystem

/dev/sdb2 411648 821247 409600 200M Microsoft basic data

/dev/sdb3 821248 1230847 409600 200M Apple HFS/HFS+

/dev/sdb4 1230848 1755135 524288 256M Linux filesystem

/dev/sdb5 1755136 2164735 409600 200M Microsoft basic data

/dev/sdb6 2164736 2574335 409600 200M Partition d'échange Linux

/dev/sdb7 2576384 4028382 1451999 709M Linux filesystem

#include <stdio.h>

#include <stdlib.h>

#include <sys/types.h>

#include <sys/stat.h>

#include <fcntl.h>

#include <unistd.h>

#include <errno.h>

#include <string.h>

int main()

{

char buf\_name[260];

char buf[512];

for (int i = 1; i <= 7; i++)

{

snprintf(buf\_name, 260, "/dev/sdb%d", i);

int fdPart = open(buf\_name, O\_RDONLY);

if(fdPart != -1)

{

ssize\_t len = read(fdPart, buf, 512);

if (len == 512)

{

snprintf(buf\_name, 260, "sdb%d\_0\_512.bin", i);

int fd\_file\_out = open(buf\_name, O\_WRONLY);

if(-1 != fd\_file\_out)

{

write(fd\_file\_out, buf, 512);

close(fd\_file\_out);

}

close(fdPart);

}

}

else

{

printf("open error %s\n", strerror(errno));

}

}

return 0;

}