

# BayOS

はりぼてOSとの違いについて

はりぼて友の会 bayside

<http://haribote.org/>

BayOS

とは？

はりぼてOSに  
インスパイヤ  
されたOS (笑)

一番の特徴

C++で

再実装

# 起動部分比較

はりぼてOS

```

struct BOOTINFO *binfo = (struct BOOTINFO *) ADR_BOOTINFO;
struct SHCTCL *shtctl;
char s[40];
struct FIFO32 fifo, keycmd;
int fifobuf[128], keycmd_buf[32];
int mx, my, i, new_mx = -1, new_my = 0, new_wx = 0x7fffffff, new_wy = 0;
unsigned int memtotal;
struct MOUSE_DEC mdec;
struct MEMMAN *memman = (struct MEMMAN *) MEMMAN_ADDR;
unsigned char *buf_back, buf_mouse[256];
struct SHEET *sht_back, *sht_mouse;
struct TASK *task_a, *task;
static char keytable0[0x80] = {
    0, 0, '1','2','3','4','5','6','7','8','9','0','-', '^', 0x08, 0,
    'Q', 'W', 'E', 'R', 'T', 'Y', 'U', 'I', 'O', 'P', '@', '[', 0x0a, 0, 'A', 'S',
    'D', 'F', 'G', 'H', 'J', 'K', 'L', ';', ':', 0, 0, ']', 'Z', 'X', 'C', 'V',
    'B', 'N', 'M', ',', '.', '/', 0, '*', 0, ' ', 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, '7','8','9','-','4','5','6','+','1',
    '2','3','0','-', 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0x5c, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0x5c, 0, 0
};
static char keytable1[0x80] = {
    0, 0, '!', 0x22, '#', '$', '%', '&', 0x27, '(', ')', '~', '=', '~', 0x08, 0,
    'Q', 'W', 'E', 'R', 'T', 'Y', 'U', 'I', 'O', 'P', ' ', '{', 0x0a, 0, 'A', 'S',
    'D', 'F', 'G', 'H', 'J', 'K', 'L', '+', '*', 0, 0, '}', 'Z', 'X', 'C', 'V',
    'B', 'N', 'M', '<', '>', '?', 0, '*', 0, ' ', 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, '7','8','9','-','4','5','6','+','1',
    '2','3','0','-', 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, ' ', 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, '!', 0, 0
};
int key_shift = 0, key_leds = (binfo->leds >> 4) & 7, keycmd_wait = -1;
int j, x, y, mmx = -1, mmy = -1, mmx2 = 0;
struct SHEET *sht = 0, *key_win, *sht2;
int *fat;
unsigned char *nihongo;
struct FILEINFO *finfo;
extern char hankaku[4096];

init_gdtidt();
init_pic();
io_sti(); /* IDT/PICの初期化が終わったのでCPUの割り込み禁止を解除 */
fifo32_init(&fifo, 128, fifobuf, 0);
*((int *) 0x0fec) = (int) &fifo;
init_pit();
init_keyboard(&fifo, 256);
enable_mouse(&fifo, 512, &mdec);
io_out8(PIC0_IMR, 0xf8); /* PITとPIC1とキーボードを許可(11111000) */
io_out8(PIC1_IMR, 0xef); /* マウスを許可(11101111) */
fifo32_init(&keycmd, 32, keycmd_buf, 0);

memtotal = memtest(0x00400000, 0xbfffffff);
memman_init(memman);
memman_free(memman, 0x00001000, 0x0009e000); /* 0x00001000 - 0x0009efff */
memman_free(memman, 0x00400000, memtotal - 0x00400000);

```

```

init_palette();
shtctl = shtctl_init(memman, binfo->vram, binfo->scrmx, binfo->scry);
task_a = task_init(memman);
fifo.task = task_a;
task_run(task_a, 1, 2);
*((int *) 0x0fe4) = (int) shtctl;
task_a->langmode = 0;

/* sht_back */
sht_back = sheet_alloc(shtctl);
buf_back = (unsigned char *) memman_alloc_4k(memman, binfo->scrmx * binfo->scry);
sheet_setbuf(sht_back, buf_back, binfo->scrmx, binfo->scry, -1); /* 透明色なし */
init_screen8(buf_back, binfo->scrmx, binfo->scry);

/* sht_cons */
key_win = open_console(shtctl, memtotal);

/* sht_mouse */
sht_mouse = sheet_alloc(shtctl);
sheet_setbuf(sht_mouse, buf_mouse, 16, 16, 99);
init_mouse_cursor8(buf_mouse, 99);
mx = (binfo->scrmx - 16) / 2; /* 画面中央はるよに座標計算 */
my = (binfo->scry - 28 - 16) / 2;

sheet_slide(sht_back, 0, 0);
sheet_slide(key_win, 32, 4);
sheet_slide(sht_mouse, mx, my);
sheet_updown(sht_back, 0);
sheet_updown(key_win, 1);
sheet_updown(sht_mouse, 2);
keywin_on(key_win);

/* 最初キーボード状態との食い違いがないように、設定しておくことにする */
fifo32_put(&keycmd, KEYCMD_LED);
fifo32_put(&keycmd, key_leds);

/* nihongo.fntの読み込み */
fat = (int *) memman_alloc_4k(memman, 4 * 2880);
file_readfat(fat, (unsigned char *) (ADR_DISKIMG + 0x000200));

finfo = file_search("nihongo.fnt", (struct FILEINFO *) (ADR_DISKIMG + 0x002600), 224);
if (finfo != 0) {
    i = finfo->size;
    nihongo = file_loadfile2(finfo->dustno, &i, fat);
} else {
    nihongo = (unsigned char *) memman_alloc_4k(memman, 16 * 256 + 32 * 94 * 47);
    for (i = 0; i < 16 * 256; i++) {
        nihongo[i] = hankaku[i]; /* フォントがなかったら半角部分をゴースト */
    }
    for (i = 16 * 256; i < 16 * 256 + 32 * 94 * 47; i++) {
        nihongo[i] = 0xff; /* フォントがなかったら全角部分を0xffで埋め尽くす */
    }
}
*((int *) 0x0fe8) = (int) nihongo;
memman_free_4k(memman, (int) fat, 4 * 2880);

```



長すぎ . . . (汗)

BayOS

```
/* 物理メモリサイズ */
```

```
DWORD size = get_physical_size(0x00400000, 0xbfffffff);  
g_km.init(0x00400000, size); /* メモリマネージャ初期化 */
```

```
Descriptor::init(); /* GDT/IDTの初期化 */  
PIC::init(); /* PICの初期化 */  
PIT::init(); /* PITの初期化 */  
File::init(); /* ファイルマネージャの初期化 */  
SheetCtl::init(); /* 下敷きの初期化 */  
Console::init(); /* コンソールを初期化 */  
Keyboard::init(); /* キーボードの初期化 */  
Mouse::init(); /* マウスの初期化 */  
Scheduler::init(); /* スケジューラの初期化 */  
io_out8(PIC0_IMR, 0xf8); /* PITとPIC1とキーボードを許可  
(11111000) */  
io_out8(PIC1_IMR, 0xed); /* ネットワークとマウスを許可  
(11101101) */
```

```
io_sti(); /* IDT/PICの初期化が終わったのでCPUの  
割り込み禁止を解除 */
```

```
for (;;) {  
    io_stihlt(); /* hltしないとCPU 100%になる */  
}
```

すっきり！！

この  
新スツキリ味は  
明らかに  
実感できる。



# 開発ツール比較

はりぼてOS

nask+

GO (C言語)



BayOS

nask+

MinGW (C++)

C++ 对应版

toiset

表示色数比較

はりぼてOS

256色

BayOS

65536色



# ファイル システム

はりぼてOS

FAT12

BayOS

独自

(advance/filesystem)

# フォント比較

はりぼてOS

16ドット  
(固定幅)



BayOS

12ドット

(フ°ロホ°ーシヨナル)

AAも安心

(アスキーアート)

# 文字コード比較

(内部)

はりぼてOS

ソフトJIS

BayOS

UTF-8



# メモリ管理

セグメント

(共通)

ただし

BayOSは当然

new, delete

OK!!

# タスク管理

プリエンプティ  
ブマルチタスク  
(共通)

# 実行バイナリ 形式

共通



ただし

多色化に

ともない、API

互換性低下 (汗

# GUI比較

はりぼてOS

Windows風？

QEMU



```
console
>ncest gview fujisan.jpg
>ncest invader
>nmlplay daigo.mml
```

```
mmlplay
04D daigo.mml
交響曲第5番八短調 "運命" op.67
```

```
invader
SCORE:00000000 HIGH:00000000
GAME OVER
[5 columns of green alien invaders]
[blue player ship]
```



BayOS

クラシック

Mac風





```
console
> mem
メモリサイズ : 29360128 バイト
ディスクサイズ : 27804720 バイト
ROMサイズ : 1555408 バイト
> help
CUI コントロール コマンド一覧
help/? - ヘルプ
ls/dir - ファイル 一覧 表示
cat/type - ファイル 内容 表示
date/time - 日時 表示
uname/ver - バージョン 表示
clear/cls - 画面 クリア
ps - プロセス 一覧 表示
mem - メモリ 使用状況 表示
exit/quit - シュットアウト
> noiz2bg
```



壁紙も

変えられます

ほかにも

BayGUI

移植中

# BayGUI

とは？

Java AWT風

GUI

ツールキット

```

GBBall ()
{
    setBounds((800 - 212) / 2, (600 - 228) / 2, 212, 228);
    setTitle("bball");
}

void paint(Graphics* g)
{
    g->setColor(0, 0, 0);
    g->fillRect(0, 0, 200, 200);
    for (int i = 0; i <= 14; i++) {
        int x0, y0, dis;
        x0 = point[i][0]; y0 = point[i][1];
        for (int j = i + 1; j <= 15; j++) {
            dis = j - i; /* 2つの点の距離 */
            if (dis >= 8) dis = 15 - dis; /* 逆回りに数える */
            if (dis != 0) {
                g->setColor(color4bit[16 - dis][0], color4bit[16 -
dis][1], color4bit[16 - dis][2]);
                g->drawLine(x0, y0, point[j][0], point[j][1]);
            }
        }
    }
}
}

```

ライセンス



MIT/X

取得先

[http://www.ha  
ribote.org/](http://www.ha<br/>ribote.org/)

CVSで管理

近々

ソースコード

以下移転予定

[http://haribote  
.sourceforge.jp](http://haribote.sourceforge.jp)

BayOSで

一緒に

遊びませんか？