(1) /usr/src/kernel 内のファイル tty.c
DEV_WRITE を探す。
（コメントではなく）プログラム中で

参照されているのは[　　] 行目。
この行で、関数 do_write()をコール。

(2) 同じ名前の関数が、/usr/src/kernel
の２ヶ所で定義されている。
もう一つは、
ファイル[　　]にある。

いずれも、PRIVATE 宣言されている。
PRIVATE とは何か調べてみると、
/usr/include/minix/const.h
内で、PRIVATE は[　　]と
定義されている。

ｃ言語で、このスコープ（引用範囲）は

[　　]
であるため、該当するのは、tty.c の内部で定義されている関数 do_write()であることがわかる。
定義されているのは[　　] 行目。

(3) handle_events()をコールしている。
この関数は 634 行目で定義されている。
この 660 行目では、
tty_t 構造体のメンバである

[　　] 変数
が格納している関数をコールしている。

この変数には、/usr/src/kernel 内では
３ヶ所で関数名が代入されている。

(4) rs232.c の rs_write と、
pty.c の pty_write。
そして、console.c の[　　]行目
である。
ここでは、コンソール端末を仮定すると、
handle_event()では、最終的に

[　　]
がコールされることになる。

（５）関数 cons_write()は、console.c の

[　　]行目で定義されている。

この関数の中では、
phys_copy()がコールされている。
phys_copy はアセンブラで定義されていて、データを端末に送信している。
ここで、文字列が画面表示される。

Kernel が起動する際、タスクテーブルから登録されたタスクが一つずつ起動される。
その中の一つに、tty_task()や、clock_task()がある。

このタスクテーブル tasktab は、ファイル table.c で定義されている。

Clock_task()は、ファイル clock.c の

[　　]行目で定義されている。

学生証番号：

氏名：