

[Q]: setlocale() в OS/2

[A]: Alex Samorukov (2:463/598)

Итак, в стандарте ANSI определена ф-ия setlocale, которая позволяет устанавливать локаль процесса. Мне это потребовалось заюзать в одной из своих софтинок. Оказалось это несколько не так просто сделать как мне думалось 😊

Итак, варианты LIBC:

EMXLIBC

"C" Locale only, сразу отпадает.

Innotek LIBC:

setlocale() есть и работает. При этом используется системная OS2 локаль, локаль C существует и работает. Особых проблем при использовании не выявлено.

Watcom LIBC

аналогично

VAC 3.06 RT: В принципе работает. Правда, какой-то косяк с наследованием в DLL, а также системная локаль НЕ ЮЗАЕТСЯ. Для функционирования надо прописать LOCPATH к папке с lcl файлами (внутри это dll). Причём было замечено, что lcl файлы от других версий VAC`а не подходят. Короче, не самая удобная вещь, но жить можно. Синтаксис вызова такой: setlocale(LC\_ALL, "ru\_ru.ibm-866"). Это подразумевает что в %locpath% у вас есть директория ru\_ru и в ней лежит ibm-866.loc. В случае неуспеха остаётся на "c" локали. Лучше юзать static linking или инициализировать локаль как в DLL так и в основном коде. OS/2 LIBC (ACP2): Как известно, в OS/2 входит свой LIBC который большая часть OS/2 програм и юзает. В нём, в частности есть

setlocale(). И она даже работает 😊 Более того, она не требует LCL файлов юзая внутреннюю OS/2 подсистему. И не имеет проблем с dll (локаль наследуется). Но имеет другую, крайне

неприятную особенность - "c" locale такой на самом деле не является 😊 т.е. setlocale(LC\_ALL, "c") printf("out: A=%c locale in exe=%s\n\n", toupper(0xa0), setlocale(LC\_TYPE, NULL));

даст A=A вместо положенных A=a в C locale. Что является ну совсем нехорошо и для моей задачи не подошло. Хотя, если не считать этой баги всё остальное работает хорошо.

From:

<http://www.osfree.org/doku/> - osFree wiki

Permanent link:

<http://www.osfree.org/doku/doku.php?id=ru:os2faq:os2prog:os2prog.065>

Last update: 2014/06/20 05:08



