February 9 2010

Design of embedded and real-time systems: C test

1) Reimplement yourself the following functions:

size_t strlcpy(char *dst, const char *src, size_t dstsize);

The strlcpy() function copies at most dstsize-1 characters (dstsize being the size of the string buffer dst) from src to dst, truncating src if necessary. The result is always null-terminated. The function returns strlen(src).

char *strtok_r(char *s1, const char *s2, char **lasts);

The strtok_r() function considers the null-terminated string s1 as a sequence of zero or more text tokens separated by spans of one or more characters from the separator string s2. The argument lasts points to a user-provided pointer which points to stored information necessary for strtok_r() to continue scanning the same string.

In the first call to strtok_r(), s1 points to a null-terminated string, s2 to a null-terminated string of separator characters, and the value pointed to by lasts is ignored. The strtok_r() function returns a pointer to the first character of the first token, writes a null character into s1 immediately following the returned token, and updates the pointer to which lasts points.

In subsequent calls, s1 is a null pointer and lasts is unchanged from the previous call so that subsequent calls move through the string s1, returning successive tokens until no tokens remain. The separator string s2 can be different from call to call. When no token remains in s1, a null pointer is returned.

char * i2a (int value, int base);

Convert integer to string (non-standard function) Converts an integer value to a null-terminated string using the specified base and returns a pointer to that string.

If base is 10 and value is negative, the resulting string is preceded with a minus sign (-). With any other base, value is always considered unsigned.

Duration 2 hours, with open books and notes.