Subject: : AmigaOS4
Topic: : MineCraft (MineTest) work in progress help need it
Re: MineCraft (MineTest) work in progress help need it
Author: : kas1e
Date: : 2021/1/4 19:03:55
URL:

@Salas00, Capehill

Yeah, got it for c++ as well. All compiles fine now! Thanks a bunch!

Through, I still have that strange issue #3, with 4 error windows from newlib.library saying that exit() of process xx called from wrong process xx, using IExec->RemTask(NULL).

Maybe that side back of the issue #2, as in the log i have exactly 4 strings of such kind:

Quote:

> 2021-01-02 21:59:37: INFO[Main]: Couldn't convert UTF-8 string 0x4d61696e204d656e75 into wstring
> 2021-01-02 21:59:37: INFO[Main]: Couldn't convert UTF-8 string 0x4d696e6574657374 into wstring
> 2021-01-02 21:59:37: INFO[Main]: Couldn't convert UTF-8 string 0x352e302e31 into wstring
> 2021-01-02 21:59:38: INFO[Main]: Couldn't convert UTF-8 string 0x into wstring

Imho firstly need to deal with those errors, and if errors from newlib didn't go, then investigate after.

And that happens because of that string.cpp's code:
https://github.com/minetest/minetest/b ... aster/src/util/string.cpp

See there at top bool convert(...) function which use iconv() for conversion like iconv(cd, &inbuf_ptr, inbuf_left_ptr, &outbuf_ptr, outbuf_left_ptr);

And that function is failed when it called from std::wstring utf8_to_wide(const std::string &input){...}
in the same file when doing:

Quote:

```
if (!convert(DEFAULT_ENCODING, "UTF-8", outbuf, outbuf_size, inbuf, inbuf_size)) {
infostream << "Couldn't convert UTF-8 string 0x" << hex_encode(input)
<< " into wstring" << std::endl;
```

```
        delete[] inbuf;
    delete[] outbuf;
    return L"<invalid UTF-8 string>";
    }
```

And I have exactly that "invalid UTF-8 string".

I put some printfs right before call to that "convert()" function and "inbuf" there surely correct, i can just prinfs("%s\n") it. The same it correct if I put printf in the convert() function itself. But something going wrong exactly when we call "convert()" and it didn't translate the things as expected.

Maybe iconv() call is different on our side? Strangely that i have no needs to do -liconv on linking, it takes from somewhere else in SDK, so maybe just different implementations?

Btw, i also had to add in the string.h that:

```
 #ifdef __amigaos4__

#include <string>
#include <sstream>

namespace std {
   template <typename T> string to_string(const T &n) {
      ostringstream strm;
      strm << n;
      return strm.str();
   }

   typedef basic_string<wchar_t> wstring;
   typedef basic_ostream<wchar_t> wostream;
   typedef basic_ostringstream<wchar_t> wostringstream;
   typedef basic_stringstream<wchar_t> wstringstream;

}
#endif
```

But not sure it may be related there...

Any suggestions are very welcome of course :) Just want this damn Minecraft come to us.