Subject: Re: Linux .brc handling bug
Posted by Balage on Tue, 19 Dec 2006 23:06:14 GMT
View Forum Message <> Reply to Message

BINARY_MASK is also dangerous.

```
#define BINARY_MASK(i, m) \
extern "C" byte *i[]; \
extern "C" int COMBINE(i, _length)[]; \
extern "C" int COMBINE(i, _count); \
extern "C" char *COMBINE(i, _files)[];
```

The xxx_files part is generated like this:

```
char *xxx_files[] = {
 "File1.cpp",
 "File2.cpp",
};
```

That's an array of pointers, pointing to string literals. And string literals are const. So trying to modify xxx_files[i][n] also segfaults.

The solution is to either:
- Modify type to this: const char* xxx_files[]
(This also means to modify BINARY_MASK macro)

Or:
- Generate an array for each filename, then add those to xxx_files, like this:

```
static char xxx_file_1[] = "File1.cpp";

static char xxx_file_2[] = "File2.cpp";

char *xxx_files[] = {
 xxx_file_1,
 xxx_file_2,
};
```

Anyway, I also think that byte should be replaced with char, as that's what is used for the definition.

The current fix:

```
#define BINARY(i, f) \
extern "C" char i[]; \
extern "C" int COMBINE(i, _length);

#define BINARY_ARRAY(i, x, f) \
extern "C" char *i[]; \
extern "C" int COMBINE(i, _length)[]; \
extern "C" int COMBINE(i, _count);

#define BINARY_MASK(i, m) \
extern "C" char *i[]; \
extern "C" int COMBINE(i, _length)[]; \
extern "C" int COMBINE(i, _count); \
extern "C" const char *COMBINE(i, _files)[];
```

This makes types correct, so you don't accidentally segfault.