Restricted Pointer Library Changes
WG14/N599 (X3J11/96-063)

Bill Homer
Tom MacDonald

Cray Research
An SGI Company
655F Lone Oak Drive
Eagan, MN 55121
homer@cray.com
tam@cray.com

6 September 1996

Introduction

The following library changes incorporate the **`restrict`** keyword into some library functions prototypes. This is needed when references to overlapping objects might introduce undefined behavior. At the moment, several library functions do not say anything about copying between overlapping objects even though they introduce undefined behavior.

Library Changes

Several library functions have parameters that should be adorned with the **`restrict`** keyword to avoid undefined behavior. The following library routines only have the ‘‘top’’ pointer type of the pointer that indirectly tracks the terminating point in the string: **`wcstok`**, **`mbstrowcs`**, and **`wcsrtombs`**. This keeps all the declarations below compatible with the current declaration. A few additional library routines were added since the last revision of this paper: **`freopen`**, **`setbuf`**, **`setvbuf`**, **`strtod`**, **`strtol`**, and **`strtoul`**. Some corrections were also made where library routines did not match the current specification.

```
#include <stdio.h>
FILE *freopen(const char * restrict filename, const char * restrict mode,
        FILE * restrict stream);
void setbuf(FILE * restrict stream, char * restrict buf);
int setvbuf(FILE * restrict stream, char * restrict buf, int mode,
        size_t size);
int fprintf(FILE * restrict stream, const char * restrict format, ...);
int fscanf(FILE * restrict stream, const char * restrict format, ...);
int printf(const char * restrict format, ...);
int scanf(const char * restrict format, ...);
int sprintf(char * restrict s, const char * restrict format, ...);
int sscanf(const char * restrict s, const char * restrict format, ...);
#include <stdarg.h>
int vfprintf(FILE * restrict stream, const char * restrict format, va_list arg);
int vprintf(const char * restrict format, va_list arg);
```

```
int vsprintf(char * restrict s, const char * restrict format, va_list arg);
char *fgets(char * restrict s, int n, FILE * restrict stream);
char *fputs(const char * restrict s, FILE * restrict stream);
size_t fread(void * restrict ptr, size_t size, size_t nmemb,
        FILE * restrict stream);
size_t fwrite(const void * restrict ptr, size_t size, size_t nmemb,
        FILE * restrict stream);
int fgetpos(FILE * restrict stream, fpos_t * restrict pos);
#include <stdlib>
double strtod(const char * restrict nptr, char ** restrict endptr);
long int strtol(const char * restrict nptr, char ** restrict endptr, int base);
unsigned long int strtoul(const char * restrict nptr, char ** restrict endptr,
        int base);
int mbtowc(wchar_t * restrict pwc, const char * restrict s, size_t n);
size_t mbstowcs(wchar_t * restrict pwcs, const char * restrict s, size_t n);
size_t wcstombs(char * restrict s, const wchar_t * restrict pwcs, size_t n);
#include <string.h>
void *memcpy(void * restrict s1, const void * restrict s2, size_t n);
char *strcpy(char * restrict s1, const char * restrict s2);
char *strncpy(char * restrict s1, const char * restrict s2, size_t n);
char *strcat(char * restrict s1, const char * restrict s2);
char *strncat(char * restrict s1, const char * restrict s2, size_t n);
size_t strxfrm(char * restrict s1, const char * restrict s2, size_t n);
char *strtok(char * restrict s1, const char * restrict s2);
#include <time.h>
size_t strftime(char * restrict s, size_t maxsize,
        const char * restrict format, const struct tm * restrict timeptr);
#include <wchar.h>
int fwprintf(FILE * restrict stream, const wchar_t * restrict format, ...);
int fwscanf(FILE * restrict stream, const wchar_t * restrict format, ...);
int wprintf(const wchar_t * restrict format, ...);
int wscanf(const wchar_t * restrict format, ...);
int swprintf(wchar_t * restrict s, size_t n,
        const wchar_t * restrict format, ...);
int swscanf(const wchar_t * restrict s, const wchar_t * restrict format, ...);
int vfwprintf(FILE * restrict stream, const wchar_t * restrict format,
        va_list arg);
int vwprintf(const wchar_t * restrict format, va_list arg);
int vswprintf(wchar_t * restrict s, size_t n, const wchar_t * restrict format,
        va_list arg);
wchar_t *fgetws(wchar_t * restrict s, int n, FILE * restrict stream);
int fputws(const wchar_t * restrict s, FILE * restrict stream);
wchar_t *wcscpy(wchar_t * restrict s1, const wchar_t * restrict s2);
wchar_t *wcsncpy(wchar_t * restrict s1, const wchar_t * restrict s2, size_t n);
wchar_t *wcscat(wchar_t * restrict s1, const wchar_t * restrict s2);
wchar_t *wcsncat(wchar_t * restrict s1, const wchar_t * restrict s2, size_t n);
size_t wcsxfrm(wchar_t * restrict s1, const wchar_t * restrict s2, size_t n);
wchar_t *wcstok(wchar_t * restrict s1, const wchar_t * restrict s2,
        wchar_t ** restrict ptr););
wchar_t *wmemcpy(wchar_t * restrict s1, const wchar_t * restrict s2, size_t n);
size_t wcsftime(wchar_t * restrict s, size_t maxsize,
        const wchar_t * restrict format, const struct tm * restrict timeptr);
size_t mbrlen(const char * restrict s, size_t n, mbstate_t * restrict ps);
size_t mbrtowc (wchar_t * restrict pwc, const char * restrict s, size_t n,
```

```
        mbstate_t * restrict ps);
size_t wcrtomb(char * restrict s, wchar_t wc, mbstate_t * restrict ps);
size_t mbstrowcs(wchar_t * restrict dst, const char ** restrict src,
        size_t len, mbstate_t *restrict ps);
size_t wcsrtombs(char * restrict dst, const wchar_t ** restrict src,
        size_t len, mbstate_t * restrict ps);
```

We may also want to say that if the type **va_list** in the header **<stdarg.h>** is a pointer type, then it shall be a restrict-qualified pointer type. This eliminates possible aliasing problems with **vfprintf** (for example).