NAME

adbgen – generate adb script

SYNOPSIS

/usr/lib/adb/adbgen *filename***.adb** . . .

AVAILABILITY

SUNWesu

DESCRIPTION

adbgen makes it possible to write **adb**(1) scripts that do not contain hard-coded dependencies on structure member offsets. The input to **adbgen** is a file named *filename***.adb** which contains header information, then a null line, then the name of a structure, and finally an **adb** script. **adbgen** only deals with one structure per file; all member names are assumed to be in this structure. The output of **adbgen** is an **adb** script in *filename*. **adbgen** operates by generating a C program which determines structure member offsets and sizes, which in turn generate the **adb** script.

The header lines, up to the null line, are copied verbatim into the generated C program. Typically these are **#include** statements, which include the headers containing the relevant structure declarations.

The **adb** script part may contain any valid **adb** commands (see **adb**(1)), and may also contain **adbgen** requests, each enclosed in **{ }**s. Request types are:

- Print a structure member. The request form is {*member***,***format*}. *member* is a member name of the *structure* given earlier, and *format* is any valid **adb** format request. For example, to print the **p_pid** field of the *proc* structure as a decimal number, you would write **{p_pid,d}**.
- Reference a structure member. The request form is {∗*member***,***base*}. *member* is the member name whose value is desired, and *base* is an **adb** register name which contains the base address of the structure. For example, to get the **p_pid** field of the *proc* structure, you would get the *proc* structure address in an **adb** register, for example **<f**, and write {∗**p_pid,<f**}.
- Tell **adbgen** that the offset is ok. The request form is **{OFFSETOK}**. This is useful after invoking another **adb** script which moves the **adb** *dot*.
- Get the size of the *structure*. The request form is **{SIZEOF}**. **adbgen** replaces this request with the size of the structure. This is useful in incrementing a pointer to step through an array of structures.
- Calculate an arbitrary C expression. The request form is **{EXPR,***expression***}**. **adbgen** replaces this request with the value of the expression. This is useful when more than one structure is involved in the script.
- Get the offset to the end of the structure. The request form is **{END}**. This is useful at the end of the structure to get **adb** to align the *dot* for printing the next structure member.

adbgen keeps track of the movement of the **adb** *dot* and generates **adb** code to move forward or backward as necessary before printing any structure member in a script. **adbgen**'s model of the behavior of **adb**'s *dot* is simple: it is assumed that the first line of the script is of the form *struct_address* / *adb text* and that subsequent lines are of the form

+/*adb text*. The **adb** *dot* then moves in a sane fashion. **adbgen** does not check the script to ensure that these limitations are met. **adbgen** also checks the size of the structure member against the size of the **adb** format code and warns you if they are not equal.

EXAMPLES

If there were an include file **x.h** which contained

```
struct x {
        char    *x_cp;
        char    x_c;
        int     x_i;
};
```

then an **adbgen** file (call it **script.adb**) to print the file **x.h** would be:

```
#include "x.h"

x
./"x_cp"16t"x_c"8t"x_i"n{x_cp,X}{x_c,C}{x_i,D}
```

After running **adbgen** the output file **script** would contain:

```
16t"x_c"8t"x_i"nXC3+D"" ./"x_cp"16t"x_c"8t"x_i"nXC3+D
```

To invoke the script you would type:

```
example% adb program
x$<script
```

FILES

/usr/platform/*platform-name***/lib/adb/***
platform-specific **adb** scripts for debugging the kernel

/usr/lib/adb/* **adb** scripts for debugging the kernel

SEE ALSO

adb(1), **kadb**(1M)

DIAGNOSTICS

Warnings are given about structure member sizes not equal to **adb** format items and about badly formatted requests. The C compiler complains if you reference a structure member that does not exist. It also complains about an ampersand before array names; these complaints may be ignored.

NOTES

platform-name can be found using the **–i** option of **uname**(1).

BUGS

adb syntax is ugly; there should be a higher level interface for generating scripts.

Structure members which are bit fields cannot be handled because C will not give the address of a bit field. The address is needed to determine the offset.