Pointing to Buffers - A Dangerous Practice

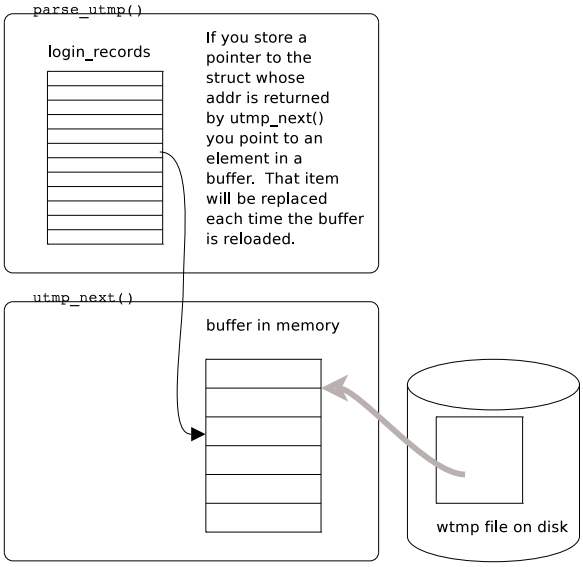
1. Introduction

The buffering library keeps an array of structs. The method that returns the 'next' item in the file actually returns a pointer to an element in this array. Perhaps you want to store the info in a particular record. Is it ok to save the pointer in a local variable?

2. No

If you copy the pointer from utmp_next() into a local variable, you have copy of the address of the struct, but not a copy of the data in the struct.

Later in the program, the buffering code will read in another set of structs from the disk. The data in the struct to which your pointer refers will be something differnt from the record you wanted to record for later reference. See the diagram below:



3. What Is the Alternative?

Store the information in a local struct or in a struct you allocate dynamically. To do the first, you need an array of structs:

```
struct utmp records[BIG_ENOUGH];   /* BIG_ENOUGH? ha! */
...
recp = utmp_next();
...
records[pos] = *recp;   /* struct copy */
```

And to do the second, you need malloc():

```
struct utmp *new_rec = malloc(sizeof(struct utmp));
list_of_recs[pos] = new_rec;
*new_rec = *recp;     /* struct copy */
```