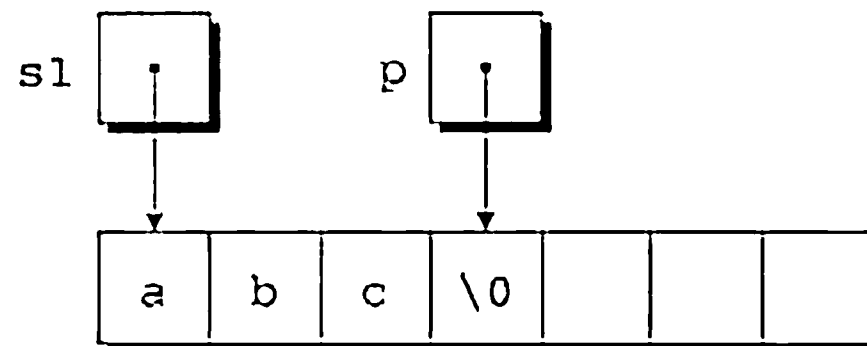
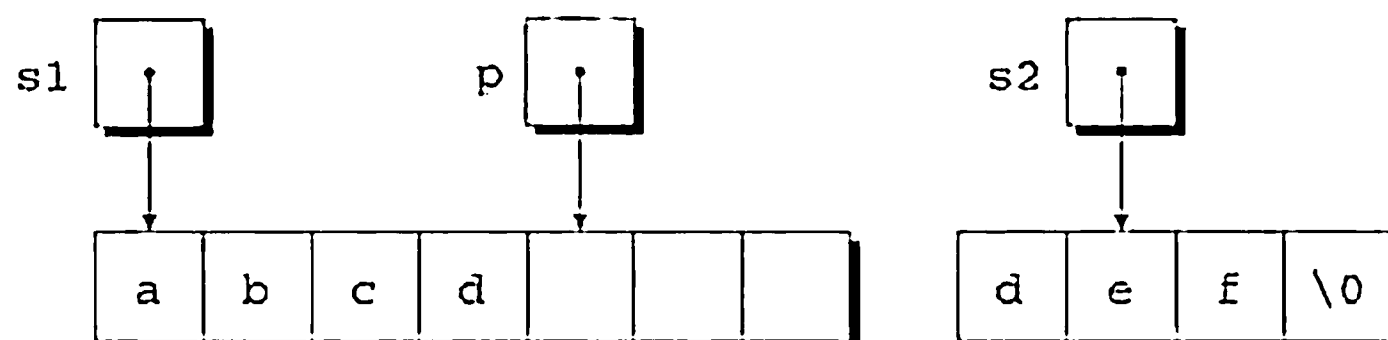


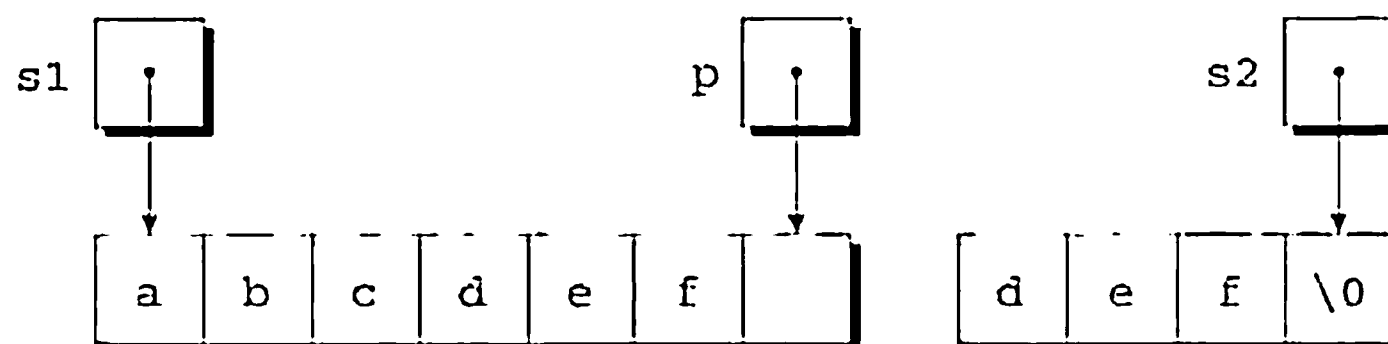
`p` is then incremented as long as it doesn't point to a null character. When the loop terminates, `p` must be pointing to the null character:



The second `while` statement implements step (2). The loop body copies one character from where `s2` points to where `p` points, then increments both `p` and `s2`. If `s2` originally points to the string "def", here's what the strings will look like after the first loop iteration:



The loop terminates when `s2` points to the null character:



After putting a null character where `p` is pointing, `strcat` returns.

By a process similar to the one we used for `strlen`, we can condense the definition of `strcat`, arriving at the following version:

```
char *strcat(char *s1, const char *s2)
{
    char *p = s1;

    while (*p)
        p++;
    while (*p++ = *s2++)
        ;
    return s1;
}
```