### Software Tools

C, Unix (Linux), and tools

# Address Arithmetic

- We can add an integer to a pointer:
  - int \*p;
  - p+3;
- This represents a pointer to an integer 12 bytes further down (assuming an integer is 4 bytes)
- The difference of two pointers is an integer
- Pointers to void do not have a defined arithmetic.

# Memory allocation

- Build simple memory allocation functions
- To allocate: alloc.
- To free: afree.

```
#define ALLOCSZ 10000
static char allocbuf[ALLOCSZ]
static char *allocp=allocbuf;
char *alloc(int n)
{
    if (allocbuf+ALLOCSZ-allocp>=n) {
        allocp += n;
        return allocp-n;
    } else
        return NULL;
}
```

### A few observations

- We declare allocbuf and allocp as static to keep them private to this file.
- Comparisons between pointer work as expected if the pointers are from the same array. It is architecture dependent if they are not.
- Assignments between pointers work as long as they are of the same type or one is void \*.

### **Pointers and Arrays**

- They are almost the same (you knew that)
- char mssg[ ] = "whatever";
  - -mssg[1] = 'H'; LEGAL

- mssg++; ILLEGAL

- char \*ptr = "something";
  - -ptr[1] = 'O'; ILLEGAL

- ptr++; LEGAL

- Defined arrays are not variables
- Pointers are variables

### strcpy

#### • Four versions:

```
void strcpy(char *s, char *t)
{
    while ( (*s=*t)!='\0' ){
        s++;
        t++;
    }
}
```

```
void strcpy{char *s, char *t)
{
    while ( (*s++=*t++)!='\0')
    ;
}
```

```
void strcpy(char *s, char *t)
{
    int i;
    i=0;
    while ( (s[i]=t[i])!='\0')
        i++;
}
```

```
void strcpy{char *s, char *t)
{
    while (*s++=*t++)
    ;
}
```

# Arrays of Pointers

- We have seen argc and argv
- char s[] means s is an array of characters
- char \*s[] means the contents of address s is an array of characters

# Pointers to pointers

- Int \*ip; means the contents of ip is an integer
  - Or ip is a pointer to integer
- Int \*\*ipp; means the contents of ipp is a pointer to integer
- This is one way to create multidimensional arrays

# Multidimensional arrays

- int vec[10]; means vec is an array of 10 integers
- int mat[12][10]; means mat is an array of 12 arrays with 10 integers each
- We **do not** write
  - mat[2,3];
- We write
  - mat[2][3];

# Problems

- Add the assignment operator to the prefix program. The program should be able to handle assignments to single letter variables (all lower case). To do this create an array with 26 elements, one for each letter. Function prefix should have two arguments now: one a pointer to a double as before and one which is a pointer to an integer index to the array of the 26 variables.
- Write the function strncmp (see the manual page). Try to be as brief and compact as possible.

### Problems

• Write the function strncpy (see the manual page). Try to be as brief and compact as possible.