

Lecture 10: Compile, link, load

- Discussion topics
- New ideas welcome!

Slide 5: Macros

- Why call-by-name parameters? Why not all-by-value or call-by-reference parameters?
- What type of code would be better to implement as macro than as subroutine?
 - Vice versa?
- What advantages do macros have as compared to subroutines? What disadvantages?

```
swap(i,j)
```

```
tmp = i;  
i = j;  
j = tmp;
```

```
swap(k, T[k]);
```

```
tmp = k;  
k = T[k];  
T[k] = tmp;
```

```
swap(k, sum += T[k]);
```

```
tmp = k  
k = sum += T[k];  
sum += T[k] = tmp;
```

Slide 8: HLL Compilation

- How would you optimize code:

```
for (i=1 to k)
  Z = Z + T[i]
  if (3*U+V mod 2 == 1)
    X = X + Z + 432102
  else
    Y = Y + T[i]
```

- Keep in register, memory or instruction?
i? k? T? T[i]? X? Y? Z?, U?, V?, 1? 2?, 432102?
- How does the value of k affect optimization?
- Could you get rid of branching? Why not or how?

Slide 10: Linking

- Give examples on situation where static linking would be better than dynamic linking?
 - Vice versa?