Goals for Today

See how `sizeof` works with arrays

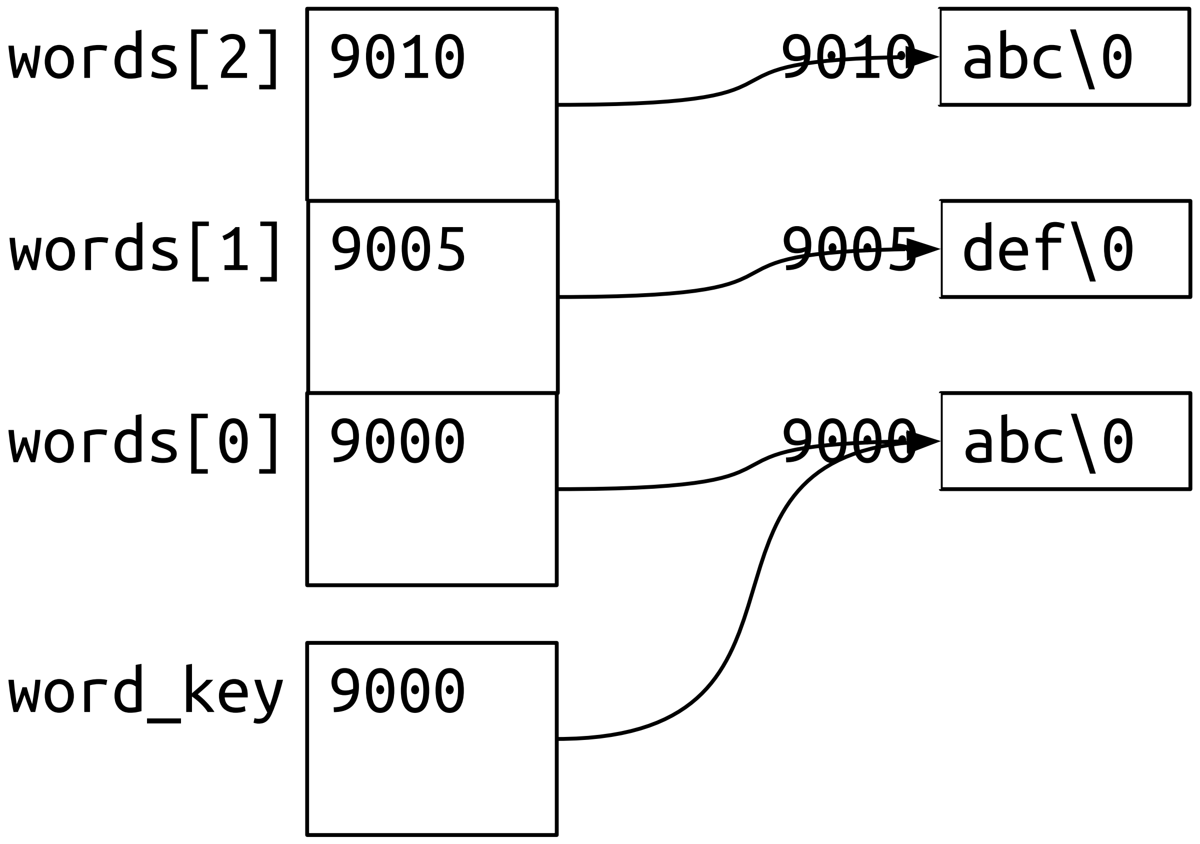
Understand the effect of type casts

Motivate the need for generic functions

Understand how to be a client of a generic function

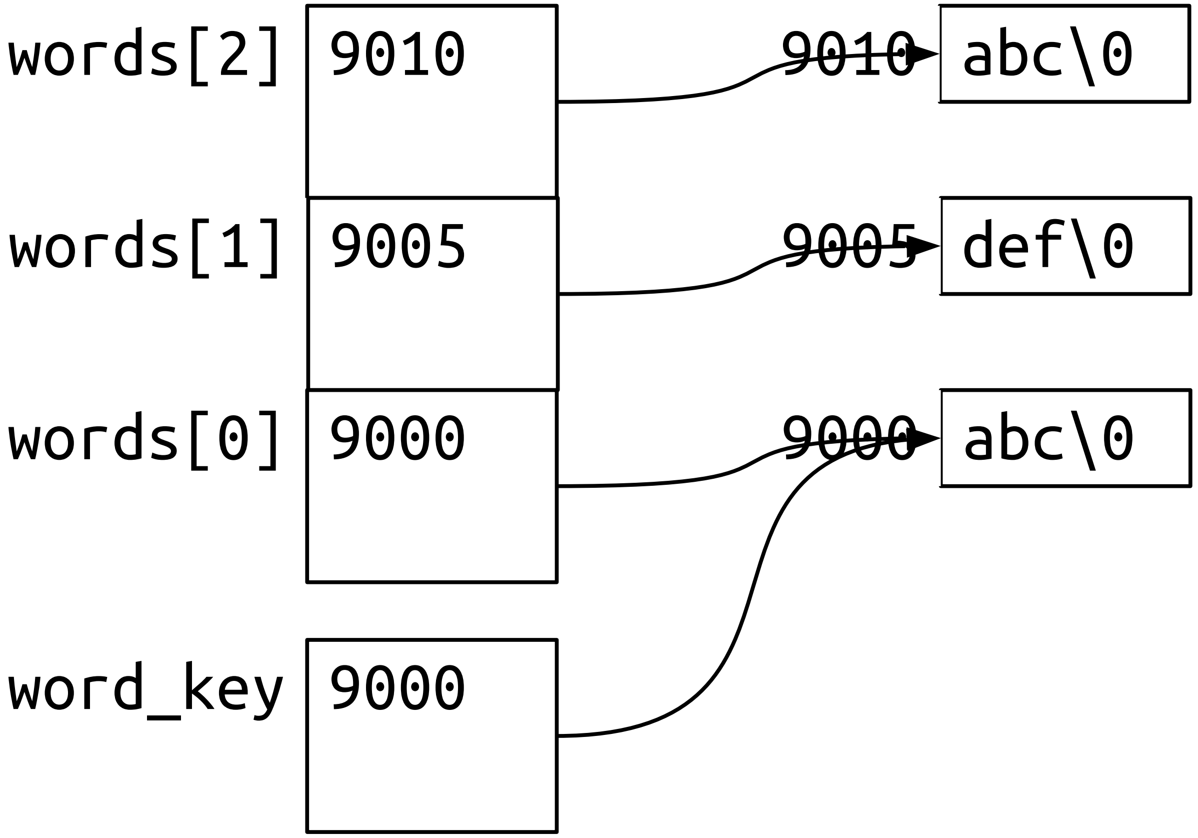
Code

count_str



words[2]
9010
words[1]
9005
words[0]
9000
word_key
9000
9010
abc\0
9005
def\0
9000
abc\0

count_str



Compares pointers for equality

Useful for finding duplicate pointers in array

Generics: What and Why

Generic function: function that can operate on different types of data

Examples

- ADTs: vector, map
- Algorithms: sorting, searching

Motivation

- Unification: no copy/pasting
- Performance: optimize once
- Convenience: put in standard library

Generics: How

New kind of pointer: `void *`

- Pointer to anything
- **Not** “pointer to void”--void is not a type

No compiler type checking

- Can assign any pointer to/from `void *`

Generic count

```
int count_int(int key, int arr[], int n);
int count_float(float key, float arr[], int n);
int count_str(char *key, char *arr[], int n);
int gcount(____ key, void *arr, int n);
```

Generic count

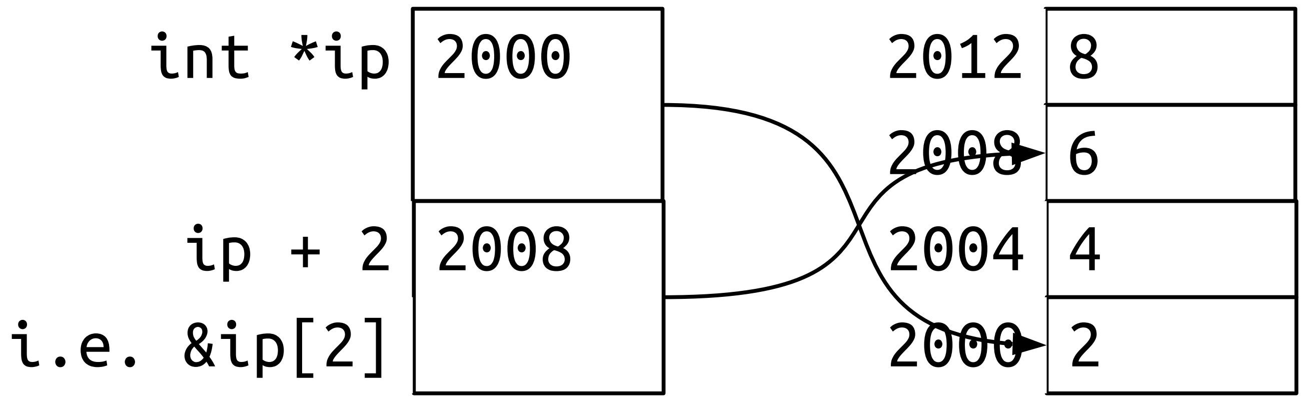
```
int gcount(void *key, void *arr, int n);
```

- Pass pointer to key, not key itself

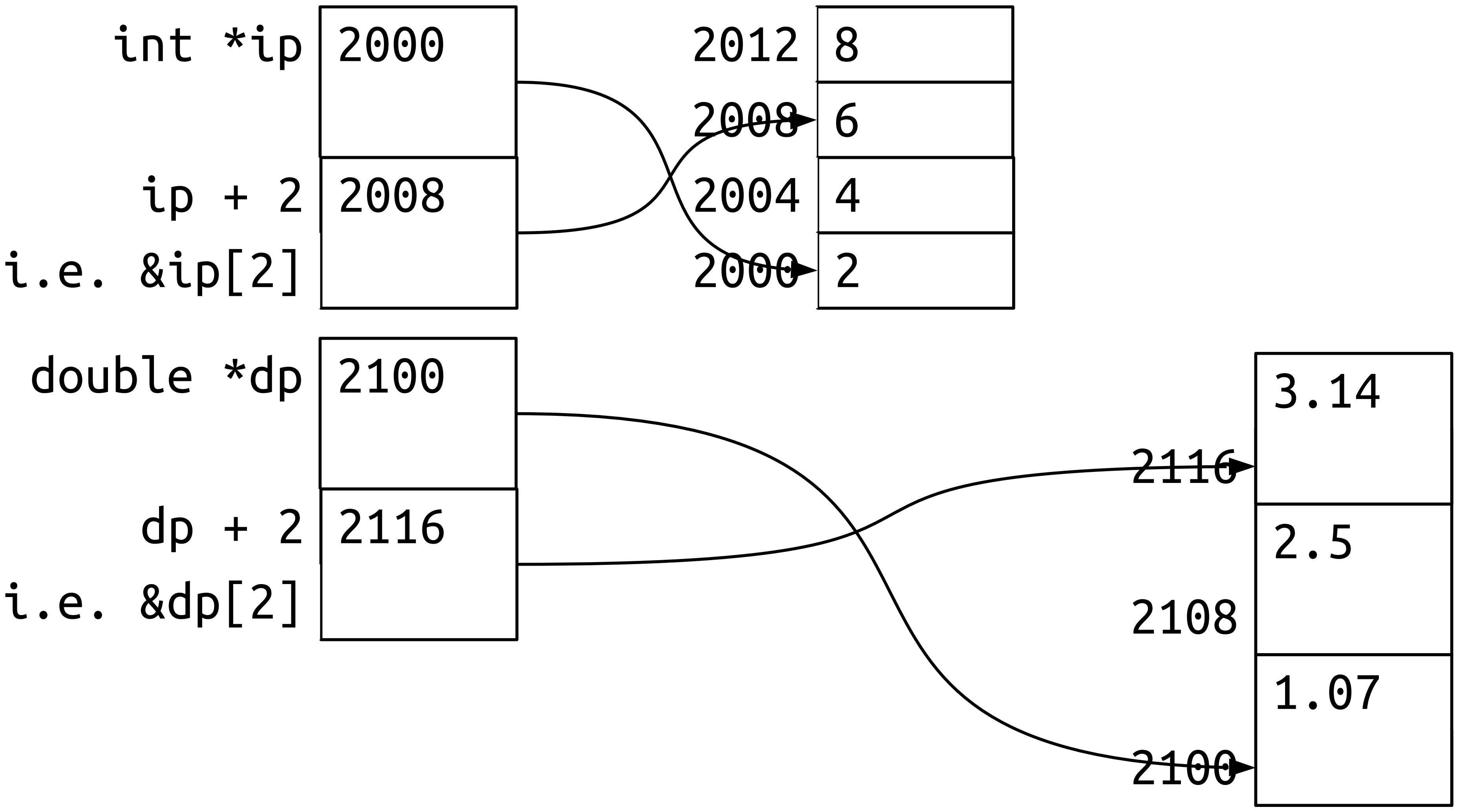
Generic functions cannot operate on elements directly. They always operate on pointers to elements.

- Parameters, return values

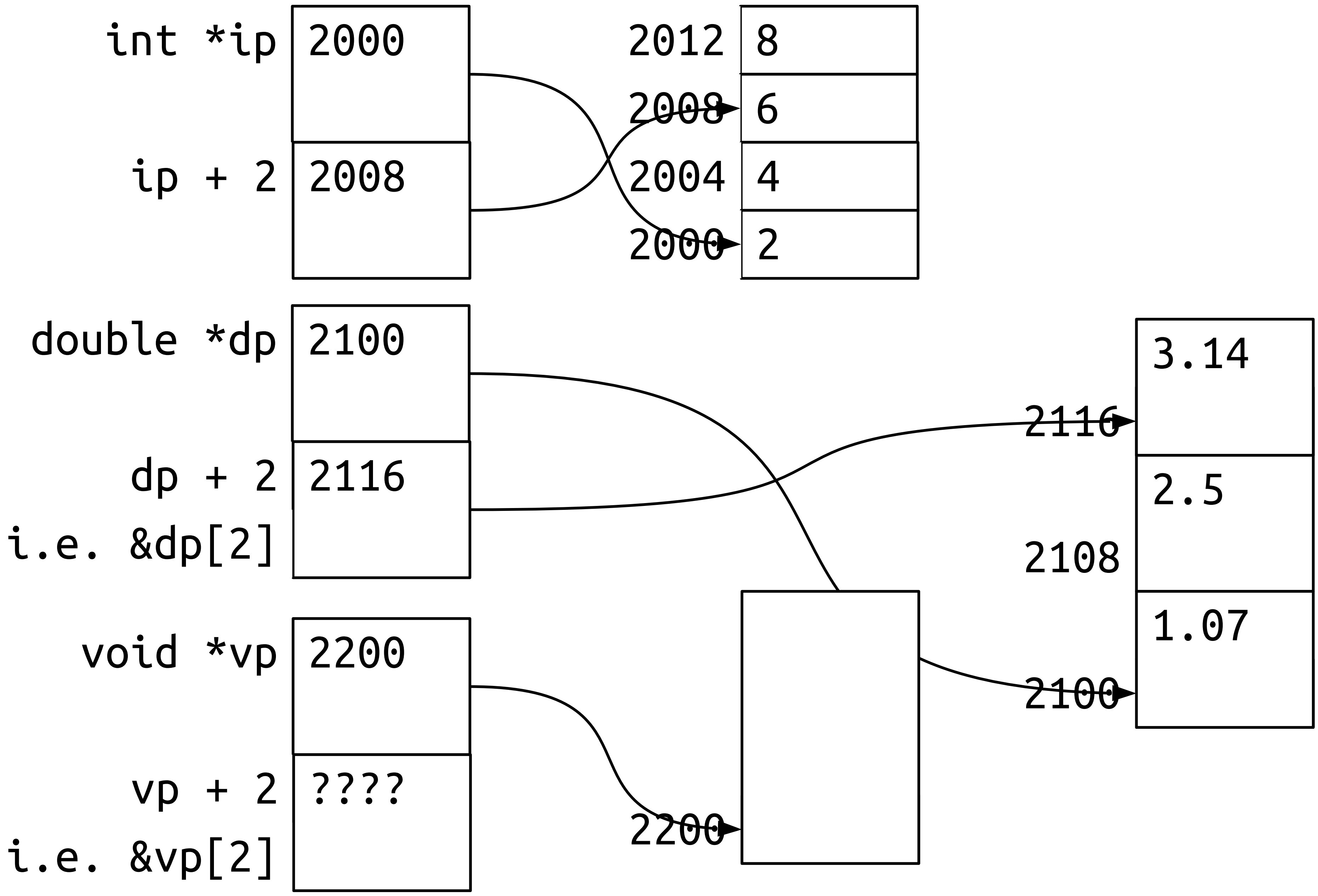
Pointer Arithmetic



Pointer Arithmetic



Pointer Arithmetic



Generic Algorithms

Comparison Function

```
int cmp(int a, int b);
```

- Negative if a < b
- Zero if a == b
- Positive if a > b

qsort

```
void qsort(void *arr, size_t nelems, size_t elemsz,
    int (*cmpfn)(const void *, const void *));
```

- Sort arr according to cmpfn

cmpfn: Callback function

- Takes pointers to elements
- Client (we) write a function for our array
- Use typecast to interpret a and b as correct type
- qsort calls cmpfn to determine order

Summary

See how `sizeof` works with arrays

Understand the effect of type casts

Motivate the need for generic functions

Understand how to be a client of a generic function