Pointers

12 September 2023 11:01

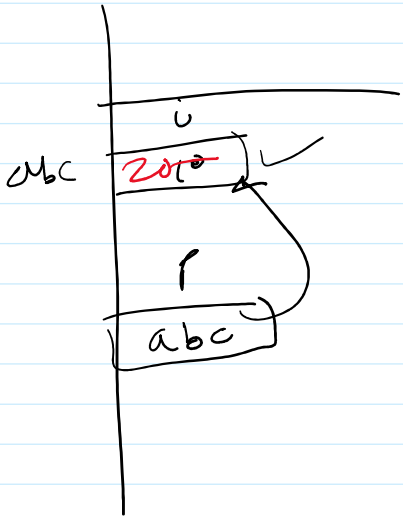
(1) Pointer is variable
↳ name, address, data type

int * p {nullptr} ← whose value is an address

int i {10} p = &i;

→ (*p) = (20);

(1) Read the value of p
(2) Is this a valid address
(3) if yes → go to that address
(4) R/W at that address



cout << p

const int ci {20};

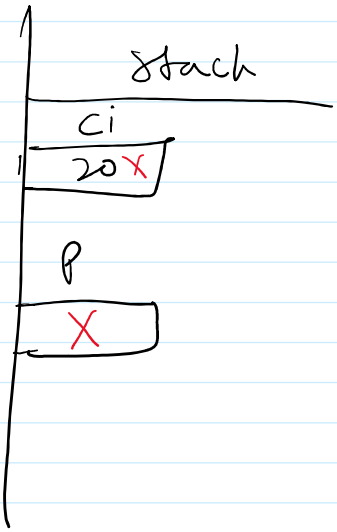
const int *p

*p = 30; ✗

int j = *p ✓

int k {30};

p = &k; ?



const int ci {30}
int *p {nullptr}

p = &ci ?

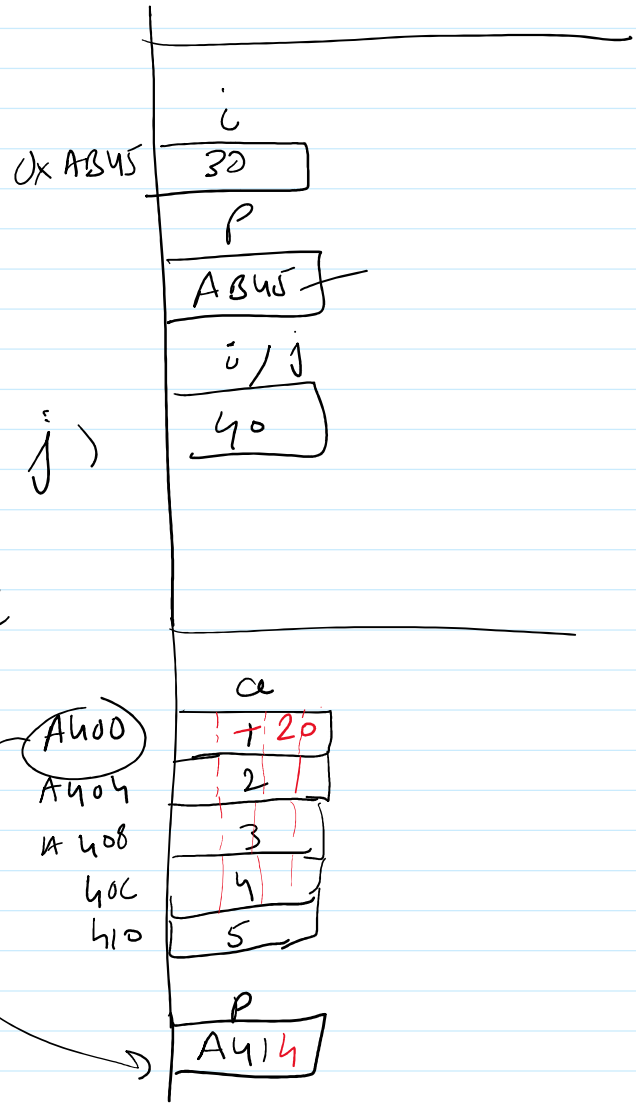
p = &c ?

int * const p {&k};

const int * const p {&k}

*p = 50; ✓

p = &i; X → f(&i)



f (int *p)
{
*p = 30;
}

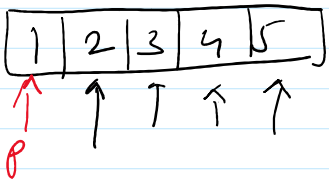
fc(i);

fc(int j)
{
j = 40;
}

int a[] = {1, 2, 3, 4, 5}

int *p {a};

a



*p = 20;

p++;

int j {20}

j++;

p++;

p++

char c {"Hello"}

char * cp {c}; cp++;

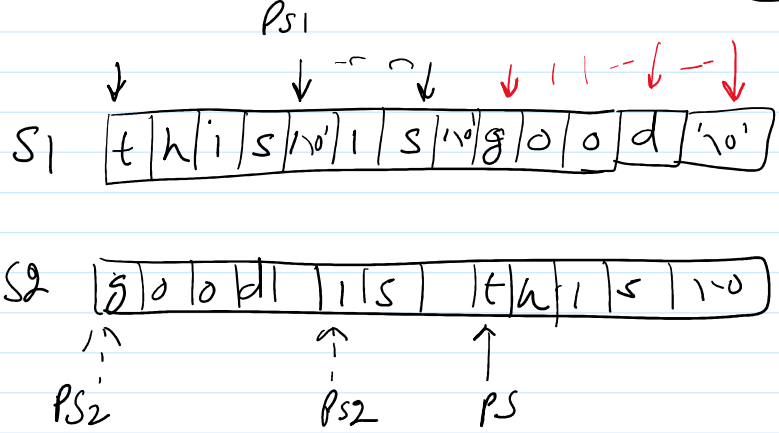
cp = (char*) a;

cp++;

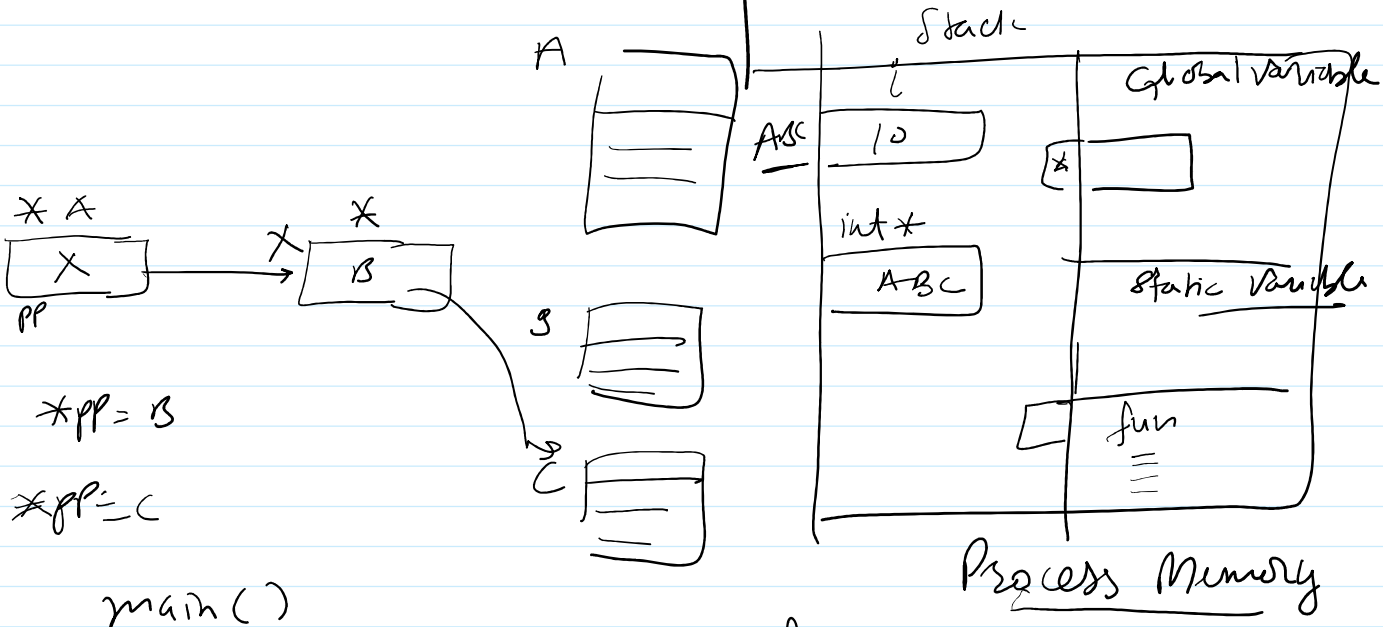
char S[] {"This is good"};

char S[] { "This is good" };

Pb: reverse the string word wise → good is This




```
strcpy(ps2, ps1)
for( *ps2 != '\0')
    ps2++
ps2 += strlen(ps2)
```




```
main()
{
   fun();
}
```

```
fun()
{ ==
}
```

① Application — Marks / CPI

- Sorting required

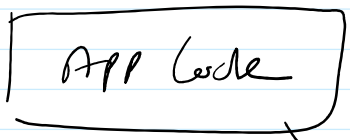
App code → sort(↓ Array)

= Std Sorting Algo

② App

App Code → Compiler

② App



→ sorting → Items by Price

→ sort (Array)

Sort (Collection a)

for ( )

Actual sorting Algo

if (cpi of a[i] > cpi of a[i-1]) → CPI Appl

Swap

Comp (i1 , i2)

i1 > i2

return

Upper layer

main ()

sortFun = Comp;

sort()

Lower layer

if (sortFun(i1, i2))

Swap

public

bool (*sortFun) (float a, float b)