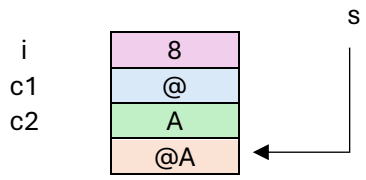
Julien RICHOZ & Paul RICCI
Group 1

Thursday, 11.04.2024

System-oriented Programming
Picasso

```
int main() {
    int i = 8 ;          // an int is represented by 32 bits.        8 in hex = 0x08
    char c1 = '@' ;      // a char is represented by 8 bits (1 parity bit + 7) = 1 byte
    char c2 = 'A' ;      // '@' in ASCII = 64 = 0x40          'A' in ASCII = 65 = 0x41
    char s[5] = ''@A" ;  // array of 5 chars = 5 bytes = 5 * 8 bits = 40 bits.
}
```

Symbolic names and values



Symbolic names and possible hexadecimal values of the adresses

Variable		Address		
name	value	name	possible value (hex-base)	possible value (10-base)
i	8	&i	0x00	0
c1	@	&c1	0x04	4
c2	A	&c2	0x08	8
	@A	s	0x0C	12

Big and little endian : symbolic memory cell and address values

Address	Big endian				Little endian				Address
0x00	0	0	0	8	0	0	0	8	0x00
0x04	@	***	***	***	***	***	***	@	0x04
0x08	A	***	***	***	***	***	***	A	0x08
0x0C	@	A	\0	***	***	\0	A	@	0x0C
0x10	***	***	***	***	***	***	***	***	0x10

Big and Little endian : hexadecimal memory cell and address values

Address	Big endian				Little endian				Address
0x00	0x00	0x00	0x00	0x08	0x00	0x00	0x00	0x08	0x00
0x04	0x40	***	***	***	***	***	***	0x40	0x04
0x08	0x41	***	***	***	***	***	***	0x41	0x08
0x0C	0x40	0x41	0x00	***	***	0x00	0x41	0x40	0x0C
0x10	***	***	***	***	***	***	***	***	0x10

Notes : We colored the variables, for more understanding, i.e. char s[5] uses 5 bytes...