

C

1972

(Kernighan & Ritchie)

Unix en C

1973

C++

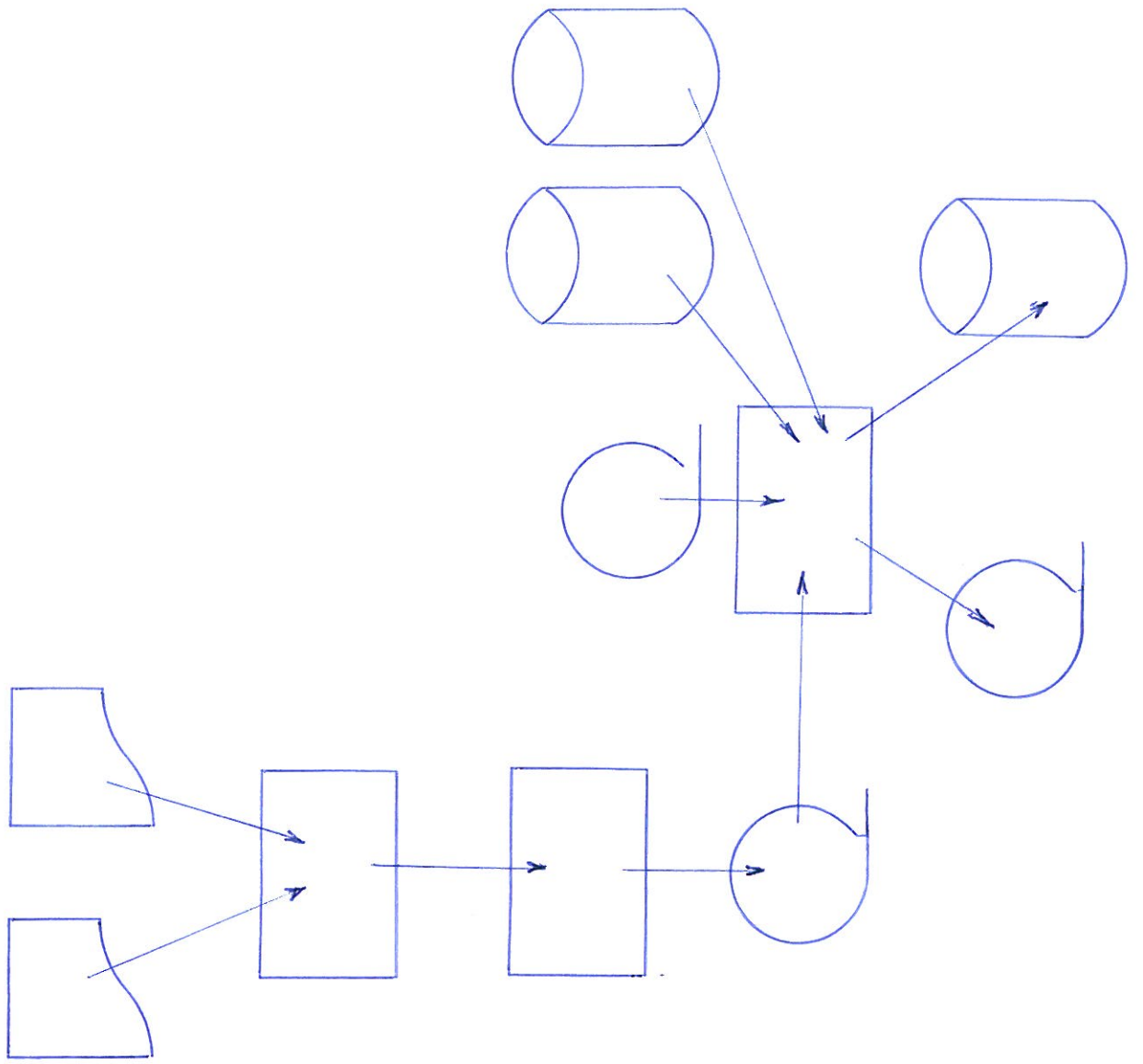
1983

(Stroustrup)

JAVA

1995

(Gosling)



asm	default	for	short	union
auto	do	goto	signed	unsigned
break	double	if	sizeof	void
case	else	int	static	volatile
char	enum	long	struct	while
const	extern	register	switch	
continue	float	return	typedef	

signed	1	char	
unsigned	2	short	
	4	int	
	4	long	const
	4	float	volatile
	8	double	register
	12	long double	auto
		void	static
target_type	4	*	extern

() [] → · * & (type) sizeof
 ! ~ ++ -- + - * &
 * / %
 + - << >>
 < <= > >=
 == !=
 & ^ _ &&
 || ?:
 = += -= *= /= %= &= ^= <<= >>=
 ,

%richt

%richt
%richt

nom designe [éventuellement un tableau de] pointeurs [s]

* nom [10];

type

au bout du/de chaque pointeur on trouve une donnée de ce type

& (variable)

fournit l'ordre de premier octet de la variable

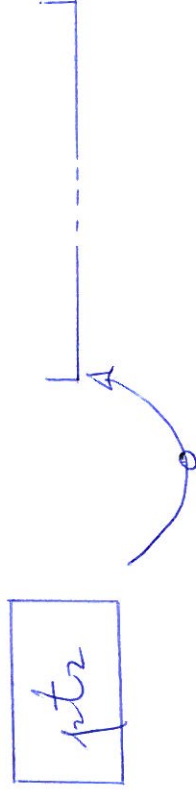
* pointeur

fournit la variable pointée par le pointeur

(à la définition du pointeur on a dit le type de cette variable. !)

```
True * ptr;
```

```
ptr = (True *) malloc (sizeof (True));
```



```
free (ptr);
```


prototype
= type de fonction

pointeur (pour un type)
de fonction

initialisation du ptr
nom de fonction =
ptr initialisé!

appel via un pointeur

définition de fonction

```
int fonction (float p) ;
```

```
int (*pf) (float p) ;
```

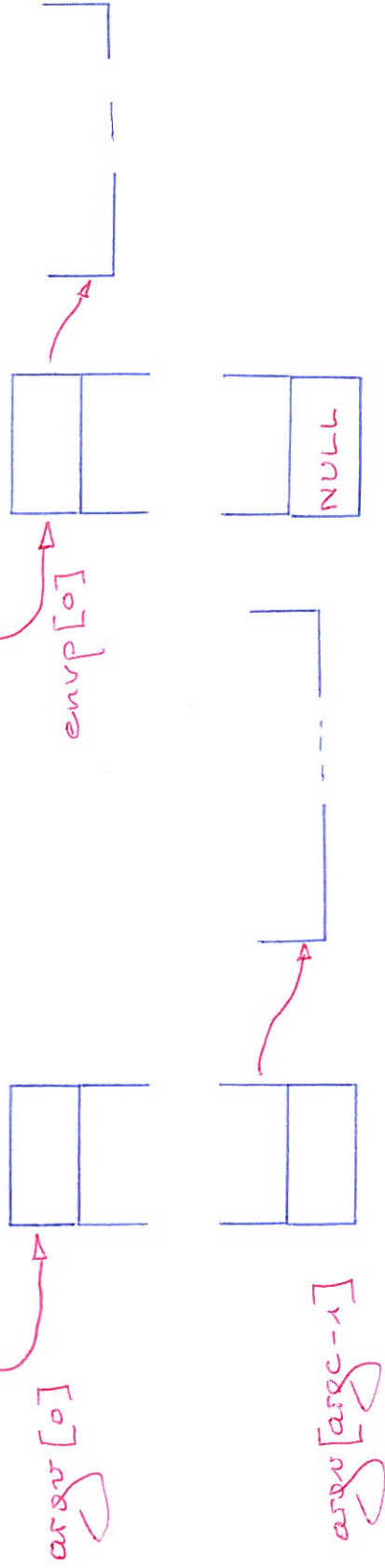
```
pf = fonction ;
```

```
int z = (*pf) (3.14 f) ;
```

```
int {  
    fonction (float p)
```

```
return (...);  
}
```

```
int main (int argc, char * arg[], char * envp[])
```



strcpy (char * cible, char * source)

memcpy (char * cible, char * source, int len)

strcat (char * deju, char * enplus)

strcmp (char * a, char * b)

memcmp (char * a, char * b, int len)

scanf ("...", ...)

sprintf ("...", ...)

```
typedef struct -TheType
```

```
{
```

```
---
```

```
struct -TheType * ptr;
```

```
---
```

```
}
```

```
TheType;
```

```
typedef union
```

```
{
```

```
---
```

```
}
```

```
TheUnion;
```

```
typedef enum { Lundi, Mardi, --- } Jour;
```

dup
pipe
socket
accept

open fdopen close	fopen fclose
read write seek	fread fwrite fseek
	fscanf fprintf fgets fputs
	fflush feof

ioctl
fcntl

size_t
fread (void * p, size_t s, size_t n, FILE * f)
fwrite (count

int
fseek (FILE * f, long offset, SEEK_SET
SEEK_CUR
SEEK_END)

long
ftell (FILE * f)

size_t
read (int fd, void * buf, size_t count)
write (int fd, const