Global task

Mari Paz Guerrero Lebrero

Grado en Ingeniería Informática

Curso 2018/2019

Statement

- Working in pairs
- The ***1st delivery*** date: Thursday, May 23th (compiler + explanation + examples of entries ...). The teacher will detect the errors before the final delivery where the students will include their corrections.
- The ***2nd and final delivery*** date: Thursday, May 30th (compiler + explanation + examples of entries ...)
  - This day, you have to expose your global task.
- Write an upward translator
- It supports programs such as the following.
- Generate an intermediate AST.
- And generate assembly code from that AST.

Input example

```
int fact(int n)
{
  if (n <= 1)
    return 1;
  else
    return n * fact(n-1);
}

int numero;

int main()
{
  printf("Dame un entero : ");
  scanf("%d", &numero);
  printf("El factorial = %d\n", fact(numero));
  return 0;
}
```

Other input example

```
int fact(int n) {
  int acum;
  acum = 1;
  while (n > 1) {
    acum = acum * n;
  }
  n = n - 1;
  return acum;
}

int numero;

int main() {
  printf("Dame un entero : ");
  scanf("%d", &numero);
  printf("El factorial = %d\n", fact(numero));
  return 0;
}
```

Must support

- The *integer* type
- Typical assignment, arithmetic, relational and logical expressions
- The '&' for *scanf*
- String literals for *scanf* and *printf* type
  - To simplify, the *scanf* and *printf* may be considered special statements with syntax rules themselves (and not ordinary functions).
  - The chains only appear as the 1st parameter of *printf* and *scanf*
- Sentences:
  - *If*
  - *While*
  - Compound statement
- Calls to functions
- Static, dynamic parameters and local variables (all whole)