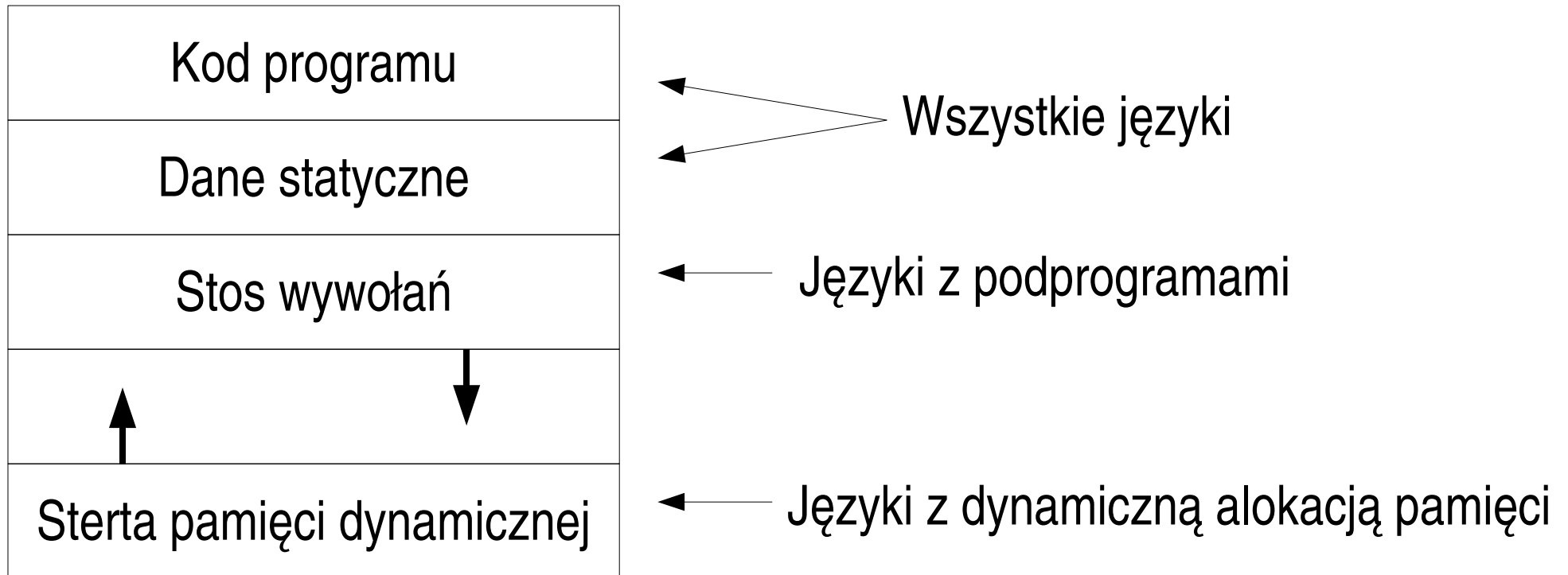


Środowisko wykonawcze

- Organizacja pamięci oraz system jej przydzielania/zwalniania w trakcie działania programu
- Realizacja wywołania podprogramu (klasy) oraz przekazywanie parametrów
- Dostęp do zmiennych, które nie są lokalne (o ile takie zmienne występują)



Rekord aktywacji procedury

1. Wartość zwracana
2. Parametry aktualne
3. Powiązanie dynamiczne
4. Dostęp do zmiennych zewn.
5. Status systemu
6. Dane lokalne
7. Zmienne pomocnicze

Przykład:

```
int func(int x, int y)
{
    char a='a'; int b=10;
    return x+y+b-(int)a;
}

void func2()
{ int z=func(1,2); }

void main()
{ func2(); }
```

Rekordy aktywacji – przydział statyczny

1. Kompilator przegląda wszystkie zmienne i na podstawie typu określa wielkość zajmowanej przez nie pamięci
2. Położenie rekordów aktywacji jest ustalane podczas kompilacji (a w związku z tym położenie ich zmiennych lokalnych również)

Zalety:

1. Rozmiar programu jest stały i znany
2. Efektywne wykonanie programu

Ograniczenia:

1. Musimy znać rozmiary wszystkich zmiennych występujących w programie
2. Nie można zrealizować procedur rekurencyjnych
3. Nie można zrealizować dynamicznego przydziału pamięci

Rekordy aktywacji – alokacja na stosie

1. Wywołanie procedury powoduje umieszczenie nowego rekordu aktywacji na stosie
2. Zakończenie wykonania powoduje usunięcie rekordu ze stosu
3. Przy pewnych założeniach, można zdefiniować pojedynczą *sekwencję startową* oraz *sekwencję końcową* dla każdego wywołania procedury

Sekwencja startowa:

- a) Wywołujący oblicza parametry aktualne i zapisuje je
- b) Wywołujący zapisuje stan stosu oraz adres powrotu
- c) Wywoływany zapisuje resztę informacji o stanie maszyny
- d) Wywoływany inicjalizuje swoje dane i rozpoczyna wykonywanie kodu

Sekwencja końcowa:

- a) Wywoływany zapisuje zwracaną wartość
- b) Wywoływany przywraca stan maszyny sprzed wywołania i przekazuje sterowanie do wywołującego (rekord aktywacji nie jest fizycznie usuwany z pamięci - zmieniamy jedynie wskaźnik wierzchołka stosu!)
- c) Wywołujący może pobrać wartość zwracaną przez wywoływanego

Zakres widzialności identyfikatorów

- *Zakres leksykalny (statyczny) bez zagnieżdżania bloków*
 - Wszystkie nazwy, które nie są definiowane wewnątrz procedur, są globalne i alokowane w danych statycznych
 - Wszystkie zmienne lokalne dla procedur, alokowane są w obszarze danych lokalnych.
- *Zakres statyczny z zagnieżdżaniem bloków*
 - Odpowiednie pole w rekordzie aktywacji wskazujące na rekord procedury nadrzędnej
 - Tablice powiązań dla odpowiednich stopni zagnieżdżenia (display)
- *Zakres dynamiczny*
 - Deep access - wyszukiwanie po powiązaniach z rekordem wywołującego
 - Shallow access - lokalizacja identyfikatorów jest przechowywana globalnie

Przekazywanie parametrów

- *Przez wartość*
 - Pamięć dla parametrów formalnych jest alokowana w rekordzie aktywacji procedury
 - Wywołujący oblicza wartości aktualne i zapisuje je w alokowanej pamięci
- *Przez referencję*
 - Jeżeli przekazywany parametr jest inną zmienną (posiadającą lokalizację), wywołujący przekazuje tę lokalizację do procedury
 - Jeżeli parametr aktualny jest wyrażeniem, dokonujemy obliczenia wyrażenia do tymczasowej lokalizacji i przekazujemy tę lokalizację.
- *Przekazywanie hybrydowe (copy-restore)*
 - Parametry przekazujemy tak jak w przypadku przekazania przez wartość, ale dodatkowo zapamiętujemy lokalizacje parametrów aktualnych.
 - Po zakończeniu procedury kopiujemy odpowiednie wartości do zapamiętanych lokalizacji parametrów aktualnych.

Różne

- Alokacja rekordu aktywacji na stercie
- Odśmiecanie (garbage collection)
- Problem wiszących referencji
- Niejawna dealokacja pamięci
 - Zliczanie referencji
 - Markowanie nieużywanych bloków
- Struktura maszyny wykonującej kod