

# 1 Opis notacji

## 1.1 Semafor

```
//Semafor binarny
semaforB identyfikator;
semaforB identyfikator = {wartość_początkowa,NULL};
//Semafor ogólny
semaforG identyfikator;
semaforG identyfikator = {wartość_początkowa, NULL};
```

## 1.2 Monitory

```
//deklaracja
monitor identyfikator {
    private:
        deklaracja_zmiennych
    public:
        deklaracja_procedur
}
//wywołanie procedury monitora
identyfikator_zmiennej_monitorowej.identyfikator_procedury (parametry);
```

## 1.3 Uruchamianie procesów współbieżnych

### 1.3.1 fork, join, quit

```
//uruchomienie procesu bez jawnego nadawania mu nazwy
fork identyfikator_podprogramu(parametry);
//uruchomienie procesu z jednoczesnym nadaniem mu nazwy
fork identyfikator_podprogramu(parametry) alias nazwa_procesu;
//pozostałe instrukcje
join nazwa_procesu;
quit;
```

### 1.3.2 parfor

```
//uruchomienie procesów bez nadawania im nazw
parfor zmienna_sterująca = wartość_początkowa to wartość_końcowa do{
    instrukcja1;
    instrukcja2;
    ...
    instrukcjaN;
}
//uruchomienie procesów z jednoczesnym nadaniem im nazw
parfor zmienna_sterująca = wartość_początkowa to wartość_końcowa do{
    as nazwa_procesu<zmienna_sterująca> {
        instrukcja1;
        instrukcja2;
        ...
        instrukcjaN;
    }
}
```

### 1.3.3 cobegin

```
//uruchomienie procesów bez jawnego nadawania im nazw
cobegin{
    instrukcja1;
    instrukcja2;
    ...
    instrukcjaN;
}
//uruchomienie procesów z jednoczesnym nadaniem im nazw
cobegin{
    as nazwa_procesu {
        instrukcja1,1;
        instrukcja1,2;
        ...
        instrukcja1,N1;
    };
    as nazwa_procesu {
        instrukcja2,1;
        instrukcja2,2;
        ...
        instrukcja2,N2;
    };
    ...
    as nazwa_procesu {
        instrukcjaK,1;
        instrukcjaK,2;
        ...
        instrukcjaK,NK;
    };
}
```

## 1.4 Przesyłanie komunikatów

```
//deklaracje
//kanał
channel {struktura komunikatu} identyfikator;
//port
port {struktura komunikatu} identyfikator;
//skrzynka pocztowa
mailbox {struktura komunikatu} identyfikator;

//wysyłanie
send komunikat to określenie_odbiornicy;
//odbieranie
receive identyfikator from określenie_nadawcy;
receive identyfikator;
//potwierdzanie odebrania komunikatu
reply;
```

## 1.5 Rozkazy dozorowane

### 1.5.1 Rozkaz dozorowany

dozór → lista\_rozkazów

### 1.5.2 Instrukcja alternatywy

```
guarded if {  
    dozór1 → lista1  
    □ dozór2 → lista2  
    ⋮  
    □ dozórN → listaN  
}  
guarded if zmienna_sterująca = wartość_początkowa to wartość_końcowa {  
    dozór → lista  
}
```

### 1.5.3 Instrukcja powtarzania

```
guarded do {  
    dozór1 → lista1  
    □ dozór2 → lista2  
    ⋮  
    □ dozórN → listaN  
}  
guarded do zmienna_sterująca = wartość_początkowa to wartość_końcowa {  
    dozór → lista  
}
```

## 1.6 Zdalne wywołanie procedury

```
//Deklarowanie serwera RPC  
server identyfikator {  
    local:  
        deklaracja_zmiennych  
        deklaracja_procedur_lokalnych  
    remote:  
        deklaracja_procedur_udostępnianych_zdalnie  
}  
//Uruchamianie serwera RPC  
startServer identyfikator_serwera alias nazwa_serwera;  
//Instrukcja accept  
accept identyfikator_procedury_zdalnej(parametry)  
    → lista_rozkazów  
//Wywołanie procedury zdalnej  
nazwa_serwera.identyfikator_procedury_zdalnej (lista_parametrów);
```