

Osoba 1: Obsługa programu od uruchomienia do przetworzenia danych wejściowych

1. Inicjalizacja programu:

- **Implementacja funkcji głównej (main):**
 - Obsługa argumentów wywołania programu (jeśli są przewidziane w projekcie).
 - Inicjalizacja niezbędnych struktur danych (np. zmienne środowiskowe, historia).
 - Obsługa sygnałów systemowych (np. SIGINT, SIGQUIT).
- **Konfiguracja środowiska:**
 - Zainicjalizowanie zmiennych środowiskowych (getenv, setenv).
 - Przygotowanie historii poleceń.

2. Obsługa interfejsu użytkownika:

- **Wyświetlanie prompta:**
 - Stworzenie funkcji do generowania dynamicznego prompta (np. zawierającego nazwę użytkownika, ścieżkę bieżącą).
- **Zbieranie danych wejściowych:**
 - Obsługa wejścia użytkownika za pomocą readline i przechowywanie wpisanych poleceń w historii (add_history).
 - Obsługa skrótów klawiszowych (ctrl-C, ctrl-D, ctrl-\\).

3. Analiza i wstępne przetwarzanie danych:

- **Rozbicie wprowadzonej komendy:**
 - Parsowanie ciągu znaków na listę argumentów (np. z użyciem strtok lub podobnych).
 - Obsługa cudzysłowów (pojedynczych i podwójnych), które modyfikują sposób interpretacji polecenia.
- **Rozpoznanie polecenia:**
 - Sprawdzenie, czy wpisana komenda to wbudowana funkcja (np. cd, exit), czy zewnętrzny program (np. ls, grep).
- **Zarządzanie zmiennymi środowiskowymi:**
 - Wykrywanie zmiennych środowiskowych (np. \$HOME) i rozwijanie ich wartości.

Osoba 2: Obsługa przetworzonych danych do momentu uzyskania efektu końcowego

1. Interpretacja przetworzonych danych:

- Odbiór struktury danych zawierającej:
 - Nazwę polecenia.
 - Argumenty.
 - Ewentualne informacje o redirekcjach, potokach itp.
- Rozpoznanie, czy polecenie wymaga dalszej obsługi (np. redirekcji, potoku) czy można je wykonać bezpośrednio.

2. Obsługa wbudowanych komend:

- Implementacja funkcji dla poleceń:
 - `cd` – zmiana katalogu.
 - `pwd` – wyświetlenie bieżącej ścieżki.
 - `echo` – wyświetlenie tekstu z opcją `-n`.
 - `export` i `unset` – zarządzanie zmiennymi środowiskowymi.
 - `env` – wyświetlenie zmiennych środowiskowych.
 - `exit` – zakończenie działania shella.

3. Obsługa zewnętrznych programów:

- **Wyszukiwanie ścieżki do programu:**
 - Użycie zmiennej `PATH` do znalezienia wykonywalnego programu.
- **Uruchamianie programu:**
 - Stworzenie procesu potomnego za pomocą `fork`.
 - Wykonanie programu za pomocą `execve` w procesie potomnym.
 - Oczekiwanie na zakończenie procesu (`wait` lub `waitpid`).

4. Obsługa potoków i redirection:

- **Redirekcje:**
 - Implementacja obsługi `<`, `>`, `<< i >>`.
 - Przekierowanie wejścia/wyjścia za pomocą `dup2`.
- **Potoki (`|`):**
 - Łączenie wyjścia jednego procesu z wejściem kolejnego za pomocą `pipe`.